

¿Cómo estudiamos el software libre?

Vicente Matellán Olivera

28 de octubre de 2003

Empiezan a surgir diferentes cursillos, asignaturas, cursos de especialización, “masters”, etc. relacionados con el software libre en general y con GNU/Linux en particular. Esto se puede interpretar como el software libre se sigue consolidando como alternativa, que los usuarios comienzan a identificarlo como algo necesario, útil, etc.

Sin embargo, en mi opinión esta proliferación puede llegar a ser contraproducente, porque no se están explicando los principios básicos del software libre, y además porque por lo que llevo visto, todas estas aproximaciones son bastante incompletas, aunque como siempre existen honrosas excepciones.

De cursos está el mundo lleno

Entre la gran oferta podemos distinguir diferentes categorías. Por una parte están los cursos clásicos que podríamos definir “de ofimática”, sólo que en el caso que nos ocupa se utilizan mayoritariamente programas libres, generalmente sobre GNU/Linux. Así, se cuenta el manejo básico de las X y de algún escritorio, KDE o Gnome; se usa la *suite* OpenOffice, o el `gnnumeric` y el `abiword`, el `gimp` para el tratamiento de imágenes, etc.

El principal problema con estos cursos es que los alumnos no aprenden que eso del “software libre”, primero porque no suele estar en el programa y segundo porque no entienden que sea algo importante. Además, en el caso de que se les explique, lo que es el software libre, la idea que les queda en mi experiencia es que es “gratis”, en el mejor de los casos. El motivo es que para ellos es una moda más, un sistema operativo que dicen que es mejor pero en el que se hacen básicamente las mismas cosas que en el conocido MS-Windows(TM). De hecho en muchos de estos cursos se mezclan herramientas libres con otras propietarias que simplemente se han portado.

Otro tipo de curso es el de administradores, donde se cuenta como instalar y administrar sistemas basados en software libre. El temario típico de estos cursos tiene que ver con la gestión de usuarios (altas, permisos, cuotas, etc.), con la instalación básica de los equipos (tarjetas gráficas, de sonido, impresoras, etc.) y con la gestión de los servicios de Internet, donde se cuenta el servidor web Apache, servidores de correo electrónico, de aplicaciones, etc.

De nuevo el problema aquí es la falta de docencia sobre los principios básicos del software libre, las famosas libertades de uso, de modificación y de redistribución que tantas veces hemos citado. En este caso el problema es convencer a los alumnos de que esas libertades son importantes. De hecho muchos piensan que por tener el código fuente, o parte del mismo de los problemas que usan ya tienen resueltos sus problemas. No se dan cuenta de que no van a poder con muchas de las licencias redistribuir por ejemplo sus modificaciones ni obtener los arreglos (parches) que hayan hecho otros administradores.

Otro tipo de cursos son los centrados en los aspectos de metodología de desarrollo. Así, algunos cursos empiezan a estudiar el aspecto más informático del modelo del desarrollo del software libre. Inciden en lo que podríamos llamar la “Ingeniería del Software Libre”. En esa línea estudian como se organiza un proyecto de software libre, como evoluciona, como se podría implementar en una empresa un mecanismo similar, que motivaciones existen para hacerlo, etc. Desgraciadamente prácticamente no existe este tipo de curso. Excepto quizás el “Master Internacional en Software Libre” de la Universitat Oberta de Catalunya (UOC) donde se cuentan parte de estos temas.

Esto nos lleva a la oferta “reglada”, entendiendo por ella, la ofrecida por los organismos tradicionales de educación, en particular las universidades. Lamentablemente es difícil encontrar algún sitio en el que se estudie el software libre en sí mismo. Si que es cierto que en gran cantidad de universidades, por no decir en casi todas, se utiliza en mayor o menor medida software libre. De nuevo el problema es que se usa, sin explicar a los alumnos las implicaciones que tiene. Por supuesto siempre hay honradas excepciones, así se pueden encontrar algunos cursos de doctorado (por ejemplo en mi universidad :-)) sobre el tema.

Lo que siempre quiso saber y nunca se ha estudiado

Yo creo que el “fenómeno” del software libre ya tiene la suficiente entidad, se mida como se mida: en millones de usuarios, en número de desarrolladores, en líneas de código, etc. como para merecer ser objeto de estudio en si mismo. En esa línea mi curso ideal iría más en la línea de la “Ingeniería del software libre”.

Además, yo hecho en falta otros aspectos, en particular creo que nos falta un estudio más sociológico de las implicaciones del software libre. Trataré de explicar que quiero decir con eso de aproximación sociológica con un ejemplo: parece que todo el mundo está de acuerdo en que uno de los factores del éxito de GNU/Linux ha sido la capacidad de Linus Torvalds de mantener coexionada a la comunidad alrededor de GNU/Linux ¹. Sin embargo, no somos capaces, o al menos yo no lo soy, de saber ni qué es eso de “cohesionar la comunidad” ni mucho menos cómo se hace.

Probablemente este asunto tenga poco de informático y caiga más en la “ingeniería social” es decir, en el estudio de la resolución de problemas sociales, pero es evidente que se trata de algo vital para que un proyecto de software libre, al menos, los que conocemos hasta la fecha prosperen (además de la utilidad y oportunidad del proyecto). Esto ha sido siempre objeto de discusión, pero para mi está meridianamente claro que hay tres factores que hay que combinar en un proyecto de software libre: la calidad del software que se desarrolla, la calidad del “material humano” que participa y la calidad de la estructura organizativa que se emplee.

La primera es la que habitualmente más ha preocupado a la comunidad del software libre, principalmente porque ha estado formada por programadores. Así los líderes de los proyectos han tratado siempre de atraer a los mejores programadores. De hecho la capacidad como programadores ha sido la que se ha empleado para medir el segundo factor, la calidad del material humano: “en tal proyecto está mengaquito que es un hacker (habilitoso programador), luego debe seguro que prospera”. Por último, la organización también ha sido algo despreciado, los líderes han pensado que con una lista de correo y un CVS todo estaba resuelto.

Desgraciadamente yo creo que eso no es así, se necesita que los participantes en un proyecto tengan capacidad para trabajar en equipo, para aceptar decisiones aún en contra de su criterio técnico, para aceptar líderes, etc. que están por estudiar. De igual forma necesitamos entender como hay que organizar los proyectos. De hecho ese “curso ideal” habría que organizarlo en base a casos, analizando proyectos exitosos o proyectos fallidos que no deberían haberlo sido.

Otro tema que creo que se debe tratar en los cursos sobre software libre es el modelo económico que subyace en la producción de software libre. Es cierto que tenemos empresas que empiezan a vivir del software libre, en muchos caso desde que las administraciones públicas se dieron cuenta de que era una opción a tener en cuenta, tanto por el desarrollo de empresas locales de tecnología como por sus implicaciones en la privacidad, seguridad, etc. que tantas veces hemos enumerado.

Estos modelos económicos son los que nos permitirán saber si la intuición que todos tenemos de que el software libre es un modelo más eficiente que el software propietario es real o no. Este es un campo que a día de hoy está prácticamente virgen, pero que es muy necesario abordar. De nuevo en curso la única solución a día de hoy sería abordarlo en forma de casos: modelos de empresas que han sobrevivido a la burbuja tecnológica, las que fallecieron, evolución de las empresas tradicionales, estrategia de las administraciones públicas, etc.

Ambos campos son prácticamente vírgenes, hay muy poco trabajo. Por seguir con la autopropaganda, en mi universidad se están empezando a dar algunos pasos en esa línea, estudiando por ejemplo cuál ha sido la evolución de algunos de los proyectos ² de software libre más importantes como Debian. La Unión Europea parece que empieza a tomarse en serio la opción del software libre como una oportunidad tecnológica frente al cuasi monopolio americano de este campo y financiará proyectos de estudio sobre el software libre dentro del VI programa marco.

Los recursos disponibles

Otro aspecto importante para que la enseñanza del software libre pueda progresar en todos estos asuntos es disponer de los recursos, principalmente de información, que permitan organizar la enseñanza de este campo. Esos recursos tienen que salir en gran parte de la propia comunidad de desarrolladores, en forma de descripción

¹http://www.wired.com/wired/archive/11.11/linus_pr.html

²<http://barba.dat.escet.urjc.es>

y análisis de los proyectos en marcha, pero también de la comunidad docente e investigadora tanto informática, como económica o sociológica.

Dentro de estos recursos, lo que necesitamos en primer lugar es documentación, libros (en papel o en electrónico), artículos, etc. que permitan ir generando un sustrato de estudio. En segunda instancia, lo que habrá que hacer es criticar constructivamente esas primeras aproximaciones para ir construyendo un buen catálogo bibliográfico para los diferentes cursos, masters o asignaturas.

Tenemos en realidad muy pocos “fondos” en esta biblioteca del software libre comparando con cualquier otra rama de la informática. De hecho tenemos alguna recopilación como “La catedral y el bazar”, artículos sueltos, etc. pero que son claramente insuficientes para poder definir el corpus de esta disciplina. La buena noticia es que parece que la cosa empieza a cambiar y que en breve podremos tener una primera generación de recursos.

Otro tipo de recursos habitual en cualquier comunidad científica o técnica es la existencia de publicaciones periódicas, léase revistas, donde se puede apreciar la evolución de la propia comunidad. Un primer ejemplo es esta misma publicación, donde esta sección y otras, abordan los aspectos básicos de la filosofía del software libre. Existen en el mercado varias revistas de este tipo, sin embargo, la mayor parte de ellas están enfocadas al usuario final, es decir, se centran en comentar las últimas versiones de las aplicaciones, los “trucos”, los manuales de instalación, etc. sin abordar esos otros temas que he citado antes.

No es ningún demérito de las editoriales, es lo natural, el mercado de usuarios finales demanda esas publicaciones. Lo que me falta es el conjunto de “revistas científicas” en las que se publiquen los estudios relacionados con el software libre, y ahí los que estamos en las universidades tenemos que reconocer nuestros pecados: nos estamos limitando a ser usuarios del software libre, haciendo dejación de nuestra obligación de investigadores y docentes de estudiar este campo.

Para terminar, me gustaría aprovechar para precisamente incentivar a todos aquellos que tengan alguna inquietud por estos temas para que hagan sus aportaciones. Hace tiempo que oigo a los líderes de diversos proyectos decir que no necesitan sólo programadores, que hace falta gente que haga la documentación, que la traduzca, ... Yo me uno a esa petición y la amplío, necesitamos gente que estudie el fenómeno del software libre en sí mismo y que difunda esos estudios. ¡Anímate!

©2003 Vicente Matellán Olivera. vmo@barrapunto.com

Se otorga permiso para copiar y distribuir este documento completo en cualquier medio si se hace de forma literal y se mantiene esta nota.