La crisis del software libre científico

David Santo Orcero *

irbis@df.ibilce.unesp.br

Julio de 2000

Esta ponencia forma parte del III Congreso Hispalinux[1] virtual, celebrado en Noviembre de 2000. Puedes participar en él en la sección de eventos de BarraPunto[2]

Resumen

Ciencia y software libre siempre han tenido una relación estrecha. Antes de que Richard Stallman desarrollara la licencia GPL[3], ya existia gran cantidad de software científico libre. En las épocas en que el peso del software propietario era más grande en el mundo "comercial", la mayor parte del software libre se concentraba en el área científica.

Por otro lado, el software libre siempre ha sido una piedra angular del desarrollo científico. Permitiendo al científico una más rápida verificación de sus teorias aprovechando código de los otros, y asegurando la posibilidad de acceder a el código a bajo costo -ya que hay una amplia tradición histórica de que los científicos compartan código apenas a cambio de reconocimiento-, el esfuerzo de desarrollo compartido ha permitido acelerar el adelanto científico en muchas areas, como por ejemplo en la quimica computacional con el proyecto QCPE[4] o el CCL[5]; y al mismo tiempo ha permitido que pueda ser realizada ciencia también en lugares con bajo presupuesto.

Sin embargo, la situación ha comenzado a mudar. En el área comercial, el software libre se hace popular hasta el punto de comenzar a plantar cara al software propietario[6]. Por otro lado, en el area científica cada vez son más frecuentes los paquetes propietarios, y vemos al software libre ahogarse y retroceder lentamente. El software científico bajo licencia GPL en muchas areas actualmente es, desafortunadamente, prácticamente inexistente.

En este artículo analizaremos la crisis del software libre científico; intentaremos analizar algunas causas de por qué su esplendor original está reduciéndose, que acciones deberían ser realizadas, así como qué necesita la GPL para poder impulsar otra vez el desarrollo de gran cantidad de software científico libre de calidad.

1 La crisis del software libre científico

Tradicionalmente, la ciencia ha sido uno de los nichos de mercado más importantes del software libre. El científico entiende naturalmente la necesidad de compartir información para ayudar en el avance científico. La misma filosofía *hacker* que ha hecho avanzar el software libre es común dentro de la ciencia.

Podemos encontrar la primera definición clara de qué es software libre con la licencia GPL[7], desarrollada por Richard Stallman[8]. Después de la infructuosa relación[9] con el software propietario que sucedió al abandono

[1] http://congreso.hispalinux.es
[2] http://barrapunto.com/eventos
[3] http://www.gnu.org/copyleft/gpl.html
[4] ftp://qcpe6.chem.indiana.edu
[5] http://www.ccl.net
[6] http://www.gnu.org/philosophy/categories.html\#ProprietarySoftware
[7] http://www.gnu.org/copyleft/gpl.html
[8] http://www.stallman.org/
[9] http://www.gnu.org/gnu/thegnuproject.html

del ITS -libre- por el laboratorio de inteligencia artificial del MIT, Richard Stallman dimitió de su posición y creó la FSF -Free Software Foundation- para difundir el software libre, tarea que comenzó con su célebre manifesto[10], y viene realizando desde entonces[11].

De cualquier modo, aunque Richard Stallman es el responsable de formalizar el concepto de software libre, de distinguirlo claramente del software gratuito y de haber sido uno de sus principales teóricos, no es el primer programador en desarrollar software libre. En el ámbito científico encontramos grandes proyectos colaborativos de desarrollo de software libre anteriores a RMS, como es el caso de GAMESS.

Sin embargo, últimamente el desarrollo de software libre científico se ha ralentizado. El estudio de este fenómeno es importante porque la existencia de software científico libre facilita que cualquier laboratorio del mundo pueda desarrollar ciencia libre usando este software como herramienta de trabajo.

2 La causa de la crisis del software científico

Hay varias razones que pueden ser propuestas como causa de la crisis. No son mutuamente excluyentes; de facto, es posible que la verdadera causa sea una mezcla de todas ellas. Sin embargo, todas tienen en común que son causadas por el método de valorar los avances científicos y el mecanismo actual de financiación de la actividad científica.

La ciencia en los últimos años se viene haciendo más competitiva. Por razones que quedan fuera del ámbito de este trabajo -quizás desempleo, quizás más valorización del científico- cada vez son más las personas que al terminar su graduación se deciden por cursar un curso de posgraduación, y que prestan concurso para la carrera docente en alguna universidad. El hecho de aumentar la competencia ha hecho que la competición se haga más difícil. Y, de entre los distintos factores que intervienen en un concurso -prueba didáctica, experiencia docente...- o que intervienen en la obtención de una beca de investigación -experiencia investigadora- aquel en que es más fácil marcar diferencia entre personas de la misma promoción es el número de artículos publicados en revistas indexadas.

Por otro lado, entre los grupos ya consolidados la situación es similar. Cada vez existen más grupos de investigación, y la financiación no ha crecido proporcionalmente, por lo que la competencia es cada vez mayor. En muchos países, no llega a manos del investigador una perra gorda si no presenta un rendimiento en publicaciones, via artículos publicados en revistas indexadas.

Este sistema "publica o perece" perjudica a los que desarrollan software libre, dado que no existen revistas indexadas específicas de software libre, por lo que es difícil conseguir una publicación de una nueva herramienta desarrollada en una revista indexada. El único método es publicar en una revista de informática teórica, lo que no siempre es posible. Además, esto hace que los que trabajan en informática teórica lo tengan más fácil, porque mientras que desarrollar una solución libre completa con un algoritmo nuevo involucra también un interface gráfico, un montón de pruebas y gran robustez, una comparativa entre dos algoritmos necesita menos tiempo para ser desarrollada. Por otro lado, si no implementa un algoritmo nuevo, no puede ser publicado; de nada vale decir: "mi programa es el primer y único programa libre que hace tal cosa". Si alguien ya planteó el problema, aunque no haya desarrollado la aplicación, él publicó y la aplicación completamente desarrollada, funcional, no generará ninguna publicación. Queda publicar en revistas de otra área; por ejemplo, publicar en revistas de biofísica los programas libres de biofísica. sin embargo, contamos con los mismos problemas que en las revistas de informática teórica, junto con los problemas de que tenemos que hacer un trabajo bien por encima de la media de los de la revista por estar fuera de area. Por si esto fuera poco *handicap*, además, muchas universidades y agencias de fomento de investigación ponderan negatívamente, o incluso ni valoran aquellos artículos indexados realizados en revistas de fuera del área -en nuestro caso, informática-.

Además, los mecanismos de avaliación de rendimiento son ahora a muy corto plazo. Por ejemplo, la agencia de becas que financia mi beca de doctorado exige un informe anual de rendimiento científico, con número de publicaciones en revistas indexadas. Si el rendimiento no es bueno, pierdo la beca.

Por otro lado, los proyectos de software científico que deben ser encarados actualmente en muchas áreas -física, química, medicina, entre otras- son de gran porte, necesitando años para su desarrollo.

Antiguamente era rentable a un grupo consolidado tener uno o varios científicos trabajando a tiempo completo desarrollando software libre. El placer de crear, de descubrir, suponía ya suficiente aliciente como para perder un poco de rentabilidad científica durante unos años, ya que parte del grupo estaba "entretenida" en tareas que no iban a suponer publicaciones -desarrollo de software científico-. Sin embargo, actualmente, cada vez se da menos este caso. La supervivencia del grupo y el mantenimiento de la beca de cada uno de los investigadores depende de las publicaciones con vista a un año. Esto supone que en los grupos muchas veces se

[10] http://www.gnu.org/gnu/manifesto.html
[11] http://www.gnu.org/gnu/gnu-history.html

prioriza la investigación con resultados inmediatos. Rob Pike, en su trabajo[12] sobre por qué la investigación de informática de sistemas estaba muerta, ya daba la voz de alarma. En areas que no sean la informática, como es el caso de la ciencia base, la investigación en herramientas informáticas sufre más todavía de los problemas expuestos por Rob Pike.

La licencia GPL, aunque excelente, no ayuda tampoco a desarrollar software científico libre. Por un lado, es legal según la GPL cambiar el nombre del autor -aunque no sea ético-. La GPL no permite realizar restricciones adicionales a la licencia, cuando sería necesario establecer algún mecanismo para que se citen los trabajos del autor del programa. Por otro lado, a diferencia de otros áreas, no tiene sentido en muchas áreas de ciencia base abrir una empresa de servicios de valor añadido. Aunque esto funcione bien en algunos casos -como es el caso de HelixCode, o de cualquiera de las empresas de distribuciones- la masa crítica de posibles usuarios es lo suficientemente baja como para que no sea posible la supervivencia dando consultoría. Sí, existen universidades y algunas empresas con grandes paquetes de software cerrado. Pero esos paquetes llevan años desarrollándose -algunos más de 20 años-, con el principio que no tenían competencia; y cada dia que pasa, los paquetes son mejores y es más difícil que el software libre pueda presentar competencia real. Un ejemplo es el Whatif. Es un excelente paquete de ingeniería de proteínas, pero es propietario. Existen projectos intentando realizar sustitutos, como es el caso de Platon, que son lo suficientemente poco conocidos y lo suficientemente poco desarrollados como para no ser tenidos en cuenta. Y cada vez la distancia se hace más grande, y ahora que el Whatif se ejecuta maravillosamente en Linux, es probable que acabe con la competencia. Esto supone que en una cosa tan crítica como el diseño de medicinas a partir de proteínas sintéticas no contaremos con software libre plenamente desarrollado.

Por tanto, una primera causa para esta desaceleración del crecimiento del software científico es el sistema de valoración del rendimiento científico. Mas, aunque seamos lo suficientemente idealistas como para jugar a la ruleta rusa con nuestra beca, las cosas no suelen ser tan simples. Universidades y centros de investigación suelen estar rígidamente jerarquizados. A diferencia de estructuras como el ejercito, donde desde el principio está claro quien es el jefe, en los centros de investigación las relaciones de poder suelen ser delicadas, difusas y necesitan de una gran cantidad de diplomacia. Modificar la licencia de un proyecto científico para licencia libre supone, como mínimo, convencer al orientador y al jefe de grupo de investigación de la idoneidad de liberar un código a cambio de nada. También hay que convencer a los compañeros de grupo, que ayudan incorporando su conocimiento a nuestra aplicación. En caso de responsabilidades compartidas -varios jefes-, la cosa se complica. En el momento que uno ponga reticencias, todos se cerrarán en banda. Y si el viejo gurú del laboratorio dice que eso no sirve para nada, necesitaremos un milagro para continuar con el proyecto.

3 Posibles soluciones al problema del software científico

La primera solución puede ser la existencia de una licencia libre, mas que recoja las singularidades del software científico. Esta licencia debería compartir el espíritu de la GPL y gran parte de su letra; y debería asegurar la preservación de las libertades propuestas por RMS. Por otro lado, también debería incluir algunas restricciones, fáciles de cumplir y que no limitan la libertad final del código según los criterios expuestos por RMS[13], como son:

- Si se usa el programa para realizar una investigación, y esa investigación termina en uno o varios artículos publicados, los artículos de dicha investigación deben reconocer qué programas fueron usados, así como citar los artículos en los que fueron descritos. Esta restricción no impide la libertad de libre uso, y, de hecho, la totalidad de los programas libres científicos la incluyen; este sistema hasta ahora parece funcionar. La principal objeción a esta restricción es similar a la crítica que RMS hizo a la licencia BSD[14]; sin embargo, en este caso no se produce el exceso de verborrea en las citaciones; ya que, en caso de que alguien pueda publicar un artículo en una revista indexada, por la naturaleza de revisadas por asesores externos de estas revistas, ha de ser cierta importancia la aportación al código. Y para un cálculo en particular no es tan grande el número de citaciones derivadas. Por ejemplo, el CCP4 es un proyecto libre, con modelo de desarrollo de bazar y una cantidad muy alta de personas que contribuyeron. Para un uso particular del programa -por ejemplo, cálculo de la estructura de una proteína por sustitución molecular- es apenas preciso citar, a lo sumo, tres artículos. Observamos que cualquier uso no científico del software -como es el caso del uso educativo- no entra dentro de esta restricción, y puede ser usado en condiciones iguales a la GPL.

[12] `http://www.cs.bell-labs.com/who/rob/utah2000.ps`

[13] `http://www.gnu.org/philosophy/free-sw.html`

[14] `http://www.gnu.org/philosophy/bsd.html`

- Si se modifica el código, se ha de identificar adecuadamente la parte modificada como tal. Esto no impide la libertad de modificación, mas permite que, en caso de que la modificación suponga una mejora en el rendimiento o la calidad de los resultados, el autor de la modificación tenga el debido reconocimiento; y, en caso de que la modificación introduzca errores, evita manchar el nombre del programa y del grupo que lo desarrolló. Cómo se pueden distribuir las mejoras, es un tema que sería interesante debatir en la comunidad. Si es mejor un modelo centralizado, como el de la QTL, o distribuido, como el de la GPL. Quizás fuera mejor un modelo distribuido basado en parches, con el espíritu de las modificaciones, sección 4, de la licencia GPL de documentación[15], que permite al usuario ser consciente de qué mejoras de qué autores está incorporando al código que va a compilar, para después citar adecuadamente a los autores de las mejoras.

- Por último, sería interesante abrir un debate en el seno de la comunidad del software libre sobre hasta qué punto es ético usar software libre para desarrollar ciencia no libre. Particularmente, trabajo desarrollando código de bioinformática; y no me ilusiona la idea de que programas desarrollados por mí sean usados por las multinacionales farmacéuticas para patentar genes y proteínas que ya existan en la naturaleza y cuya estructura se haya descubierto usando mis programas. Personalmente veo distinto el caso de las proteínas creadas mediante ingeniería molecular; son invención, y, por lo tanto, no veo nada malo en que sean patentables. Sin embargo, hay corrientes que no ven problema en que el software libre sea usado para calcular estructuras que ya existan en la naturaleza -como proteínas humanas y el genoma humano- y patentarlas -como permite, por ejemplo, la GPL-, y personas que se oponen a patentar cualquier tipo de información biológica. Este es, pues, un interesante debate que se debería abrir en la comunidad; porque no es posible mapear genes ni calcular estructuras de proteínas sin usar ordenadores, que muchas veces están ejecutando software libre. Me refiero aquí a ciencia libre en su sentido más amplio -que se pueda publicar para conocimiento de la comunidad científica-, aunque seria excelente que en el futuro emplear la definición estricta de ciencia libre[16].

Curiosamente, ya existe una licencia que contiene gran parte del espíritu de esta hipotética licencia de software científico. Es la licencia GPL de documentación[17], que la FSF ha estado desarrollando en los últimos años. El espíritu de esa licencia puede permitir desarrollar una licencia válida para potenciar la creación de gran cantidad de software científico libre de calidad.

La segunda solución que puede tener el problema del software científico libre es la creación de una revista indexada destinada específicamente a productos cuya licencia sea libre. Debería ser todo lo amplia que fuera posible -es decir, no aceptando apenas la GPL, sino también licencias como la BSD, la artística o la de domínio público-, con la restricción de que la licencia del software asegure las tres libertades propuestas por RMS[18]. Sería una buena idea que dicha revista se hiciese ya con todos los requisitos para ser aceptada por indexadores e indexada[19]. Estos requisitos suelen ser de tipo linguísticos, desgraciadamente -el más común es que el título, las palabras clave y el resumen estén disponibles en inglés-. La mejor licencia para esta revista indexada de software libre sería la licencia GPL de documentación[20].

La tercera solución es el apoyo de las instituciones gubernamentales. En una primera solución, financiando proyectos libres, como ya hace el gobierno alemán, solo que en el área de software científico libre. Otra posibilidad es abrir grandes proyectos temáticos de software libre científico, como ya realizó el Reino Unido con los famosos CCP. La última solución, draconiana, es que aquellos proyectos financiados con dinero del estado deben ser libres. Esta restricción ya la tiene la legislación de los EEUU -y es por eso que el GAMESS es libre-, y es de sentido común: si todos estamos pagando con nuestros impuestos el desarrollo de un programa, deberíamos beneficiarnos todos del desarrollo, y no apenas el autor.

La última corresponde a la universidad: liberando los resultados de proyectos fin de carrera, de resultados de investigación y de tesis de doctorado bajo GPL de documentación, y el código desarrollado bajo una hipotética GPL científica con las características anteriormente comentadas, ganamos todos. La comunidad porque cuenta con más código libre para compartir, y el investigador porque ve compensado su trabajo con el reconocimiento, los artículos y las citaciones que necesitará para conseguir becas y fondos para seguir investigando.

[15] http://www.gnu.org/copyleft/fdl.html
[16] http://ethology.zool.su.se/freescience/
[17] http://www.gnu.org/copyleft/fdl.html
[18] http://www.gnu.org/philosophy/free-sw.html
[19] http://www.isinet.com/isi/hot/essays/199701
[20] http://www.gnu.org/copyleft/fdl.html