

Vicente Cerveron Lleó,
Francisco Grimaldo Moreno
*Departament d'Informàtica, Escola Tècnica
Superior d'Enginyeria, Universitat de
València*

<vicente.cerveron@uv.es>,
<francisco.grimaldo@uv.es>

Entrevista a Richard Stallman

Introducción

Aprovechando la estancia de Richard Stallman en Valencia el pasado mes de febrero para pronunciar unas conferencias, tuvimos ocasión de entrevistarle en representación de la Escuela Técnica Superior de Ingeniería de la Universitat de València (ETSE-UV), junto a la periodista Núria Cadenes, de la revista El Temps.

Mientras que la periodista le hizo algunas preguntas más generales sobre libertad y derechos de autor, sobre el software libre y los objetivos de la *Free Software Foundation* (FSF), a las que Stallman respondió citando las cuatro libertades esenciales y refiriéndose a las fuentes de información al respecto que ya están publicadas en Internet, nosotros nos centramos en preguntarle sus opiniones sobre el software libre en la educación universitaria y sobre sus puntos de vista respecto a algunos temas candentes del mundo de la informática. Recomendamos al lector, si no lo ha hecho ya, recurrir a las diversas fuentes sobre software libre de la FSF así como al número 199 de *Novática* (mayo-junio de 2009) dedicado al Software libre para empresas.

No obstante, empezamos reflejando algunas de sus primeras respuestas que esbozan la idiosincrasia de un hombre, por demás pecu-

Resumen: con motivo de su visita a Valencia, entrevistamos a Richard Stallman, presidente y fundador de la Free Software Foundation y una de las principales referencias sobre software libre, y recabamos su opinión sobre el estado general del software libre y del software privativo, el papel de la educación, en particular la universitaria, en la enseñanza de uno u otro tipo de software, y sobre diversos aspectos del estado actual de la informática y de las regulaciones legales.

Palabras clave: código abierto, educación, software libre, Stallman.

liar, que para bien y para mal es una de las figuras que personifican el movimiento por el software libre.

La persona

Stallman se entiende razonablemente bien en castellano, pone sus condiciones, tiene un peculiar sentido del humor, insiste en que le tuteemos y nos saluda con un curioso "Happy hacking!"

¿Reivindicas el hacking?

Sí. Un *hacker* es alguien que se divierte usando la inteligencia. El *hacking* no tiene necesariamente relación con la seguridad informática, aunque una de las muchas cosas en las que es posible hacer de *hacker* es intentando romper la seguridad porque es difícil, y por tanto un reto. Pero hay más retos. El hecho es que hacia 1981 algunos periodistas descubrieron esta

actividad y asociaron *hackear* con romper la seguridad informática.

Ahora la connotación general es negativa.

Lo rechazo. No acepto la conversión de nuestro nombre en un insulto. Por tanto, cuando se hable de romper la seguridad informática para obtener beneficio o causar daño, utilicen el término *cracking*, y el que lo hace es un *cracker*.

¿Cómo empezó tu activismo en informática?

Trabajaba en un laboratorio del MIT (Instituto Tecnológico de Massachusetts) y prácticamente todo el software que teníamos era libre. Así aprendí a vivir en libertad y a apreciarlo. Después empezamos a perder esa libertad y me imaginé un futuro usando sólo software privativo, y me pareció un panorama muy desagradable, de manera que decidí rechazarlo y luchar por poder continuar usando los ordenadores en libertad. Para ello había que desarrollar todo un sistema operativo y después aplicaciones y así inicié el proyecto GNU y el movimiento por el software libre.

¿Qué hacías en el MIT?

Estudiaba física y al mismo tiempo trabajaba en el laboratorio del MIT: deseaba aprender de ambos campos. De física y matemáticas, sólo sabía aprender yendo a clase. Sin embargo, podía aprender a programar con la práctica; jugando.

¿Aprendiste informática de manera autodidacta?

Si uno tiene talento, si es un programador nato, no hace falta estudiarla, es suficiente con leer el manual. Yo leía muchos manuales cuando era joven, de diversos lenguajes de programación. De hecho, me interesaban mucho las diferencias entre los lenguajes. Ahora bien, recuerdo que recibí una clase de informática en la universidad. Sobre los compiladores. Y aprendí alguna cosa útil.



Richard Stallman (en el centro de la foto) en su visita a Valencia en febrero de 2010. A la izquierda Francisco Grimaldo Moreno y a la derecha Vicente Cerveron Lleó, autores de este artículo.

Entonces, ¿no es necesario enseñar ingeniería informática?

Tal vez hay algunas otras cuestiones especializadas que también es útil estudiar en clase como por ejemplo: los gráficos, la visión... aspectos en los que no soy experto. Pero a programar... se aprende programando. No es que esté en contra de enseñar programación, algunos quizás lo necesiten, pero tampoco debe sorprender que yo aprendiera sin estudiar.

El software libre en la educación

La Universidad coincide en dos de sus misiones fundamentales con el movimiento de software libre, ya que debe servir para promover el progreso de la sociedad así como ser el sustento de sus libertades. Sin embargo, el nuevo espacio europeo de educación superior orienta el diseño de los grados y másteres de ingeniería a la adquisición de competencias profesionales, las cuales a menudo están relacionadas con el uso de software privativo. ¿Debemos incluir en la enseñanza de ingeniería informática también algunos programas con los que se trabaja en la vida profesional?

No, eso es prever la derrota. No se debe hacer. Las escuelas y las universidades tienen una influencia que pueden utilizar para corregir el problema social del software privativo. Es su responsabilidad. No deben enseñar nunca, ni una sola vez, el uso de un programa privativo. No hay excusas para reforzar el dominio injusto que el desarrollador ejerce sobre la sociedad. La escuela debe enseñar únicamente software libre.

Es posible que eso cueste tiempo...

Hay universidades que ya lo hacen. Ésta es la educación ética. Al fin y al cabo, la misión social de la escuela y la universidad es educar a la siguiente generación como buenos ciudadanos de una sociedad fuerte, capaz, independiente, solidaria y libre. Enseñar los programas libres cumple con esta misión; enseñar programas privativos, no. Y no se debe hacer bajo ningún pretexto. De hecho, que la sociedad tenga una dependencia de un programa privativo no es excusa para que la escuela (universidad) actúe para mantenerla y extenderla. Al contrario: debe actuar para eliminarla.

Insistimos, teniendo en cuenta que los ingenieros deben desenvolverse con soltura en un mercado de trabajo en el que el software privativo es importante, ¿qué postura debería adoptar una Escuela de Ingeniería como la ETSE-UV si los estudiantes quieren aprender aquello que consideran que después necesitarán?

La escuela debe hacer aquello que es bueno. Si el estudiante pidiera usar heroína, ¿la escuela tendría el deber de ofrecérsela? Enseñar

Richard Stallman, el fundador del movimiento del software libre

Es muy difícil condensar en un breve perfil biográfico todas las contribuciones de Richard Stallman al mundo de la informática en particular, y a la humanidad en general. Él se autodefine como "un desarrollador de software y un activista de la libertad del software", en un buen resumen de sus principales actividades públicas. Pero es conocido fundamentalmente por algo más concreto: ser el fundador del movimiento del software libre. En gran medida, Stallman fue quien tuvo la visión de imaginarlo como concepto deseable, y quien actuó como el catalizador que lo puso marcha.

Su actividad pública en este ámbito comenzó en 1983, cuando anunció el inicio del proyecto GNU, iniciando lo que con el tiempo daría lugar a un sistema operativo libre completo. Un poco más tarde, en 1985, anunció la formación de la Free Software Foundation, que constituyó la primera estructura institucional creada específicamente para apoyar y promover el software libre. Durante esta época sentó también varias de las principales bases conceptuales, filosóficas y legales del software libre. Y todo esto lo realizó de formas alternativas a las empleadas hasta ese momento por la comunidad informática, replanteándose conceptos como los derechos de autor, y aprovechando las nuevas oportunidades que ofrecía la naciente Internet para la creación de comunidades de desarrollo.

A Stallman se debe en gran medida el concepto de software libre tal y como lo entendemos hoy día, basado en las libertades de uso, comprensión, redistribución y modificación. También el concepto de copyleft, plasmado en la licencia GPL, que garantiza

a usar software privativo es como enseñar a usar drogas adictivas. Por eso los desarrolladores de este tipo de programas suelen regalar copias: la primera dosis es gratuita, ¿no? Quieren aprovecharse de las escuelas y de las universidades para implantar la dependencia, y después es difícil escapar.

El papel del software y de las libertades en el desarrollo de la informática

Sea como sea, ¿el software libre es tan práctico a la hora de conseguir resultados como el camino que usa software privativo?

Depende. A veces va rápido, a veces va lento. Pero va en una dirección ética. Y, en cualquier caso, vale la pena llegar a un buen destino más lentamente que llegar al infierno rápidamente.

que un programa libre seguirá siéndolo a lo largo de la cadena de redistribución. Y el software creado por el proyecto GNU, en colaboración con una amplia comunidad, organizado con la meta explícita de construir un sistema completo de computación formado sólo por software libre. Stallman hizo notables contribuciones a éste proyecto, entre las que cabe destacar las primeras versiones del editor Emacs, el compilador GCC y el depurador gdb, todos ellos muy populares en sus ámbitos.

Esta actividad había posibilitado, ya en 1989, un sistema operativo prácticamente completo (faltaba el kernel), y una primera ola de adopción de software libre. Aunque desde entonces ha seguido realizando importantes contribuciones técnicas y legales (por ejemplo, a la versión 3 de la GPL), a partir de este momento Stallman es conocido en gran medida por su intensa dedicación a la promoción del software libre, escribiendo varios ensayos que han tenido gran influencia, y dando charlas por todo el mundo.

Pero Richard Stallman no sólo se ha dedicado al software libre. Además de otros muchos intereses, ha tenido (y sigue teniendo) energías para desplegar también una gran actividad en otros temas. Entre ellos cabe destacar la explicación de los peligros que suponen las patentes de software, los sistemas de control de derechos de autor (DRM), y su activismo político, normalmente enfocado a la defensa de las libertades cívicas.

(Reseña biográfica a cargo de **Jesús González Barahona**, coordinador de la sección "Software libre" de *Novática*)

Porque el software privativo no es ninguna opción, sino una trampa. He trabajado muchos años para eliminarlo de mi vida.

Habéis demostrado que el trabajo en colaboración da frutos ...

Sí, pero hay que matizar eso. En nuestra comunidad cada uno es libre y puede cambiar el software como le parezca y puede colaborar en un grupo si quiere. Pero no es un requisito: la cuestión no es que colabores, sino que tengas la libertad de hacerlo. Porque la libertad de hacer algo no significa la obligación de hacerlo: la elección es tuya.

En algunos campos, como por ejemplo la banca, hay quien no utiliza software libre porque no lo considera suficientemente seguro...

Pero es un error, porque el software privativo tiene muy mala seguridad. Windows, por ejemplo, tiene mala seguridad y puertas secretas deliberadas. Es decir, que la seguridad que ofrece Windows contra Microsoft no es que sea mala: es nula. ¿Quieres más? El iPhone tiene una puerta secreta: Apple puede imponer cambios de software cuando desee, el iPad es igual... Si quieres seguridad, el software libre es la mejor opción: la seguridad es un aspecto del control de tu informática.

Algunos investigadores de la ETSE-UV desarrollan software científico que está siendo utilizado por la comunidad científica, siendo su distribución gratuita a quien garantice por escrito (y con firma) que no hará uso comercial de los programas. Para proteger la autoría del mismo, se realiza lo que se conoce como "patente software" del código, quedando así reconocida la autoría del software. ¿Cómo se ve desde el punto de vista del movimiento de "software libre" este procedimiento de "patentar" los programas?

Las patentes informáticas no deberían existir, son un peligro para todos los desarrolladores y usuarios del software. De hecho, es necesario que los investigadores en el campo de la informática firmen declaraciones donde denuncien la idea de patentar los programas y prometan no solicitar ninguna patente nunca. Y el Estado tampoco debería autorizarlo. No se trata de hacer una mera sugerencia, sino de una campaña para eliminar las patentes de la informática: no es suficiente con tratar de no usarlas, es preciso rechazar la idea de patentar algoritmos y métodos informáticos. Hace falta legalizar el hecho de compartir. Quien ataca la práctica de compartir lo hace porque quiere dividir a la gente para dominarla.

¿Qué opinas de la censura y de los ataques sufridos por Google, supuestamente relacionados con su posición en China?

No sólo es la censura gubernamental. También hay empresas que con el argumento de que "para hacer negocios en ese país hay que cumplir las leyes", apoyan la censura en China, Australia o Dinamarca. En Dinamarca hay una lista de páginas bloqueadas; esa lista es secreta. Y Australia todavía no bloquea el acceso a las páginas pero prohíbe enlaces a esas páginas, y ya ha dicho que este año impondrá el bloqueo. (En el momento en el que realizamos esta entrevista, el cierre de webs en España todavía no estaba en su punto álgido).

Abogas por estados neutrales que se preocupen por los ciudadanos.

Sí, exacto, y no por un estado corrupto. De hecho, sabemos que los estados, en lugar de servirnos, se ponen al servicio de las empresas para mantener el poder que ejercen sobre nosotros.

En otro orden de cosas, ¿cuál es el papel del Software libre, en el desarrollo del *Cloud Computing*?

Es una pura confusión. El término *Cloud Computing* es tan difuso que no se debe utilizar nunca. Es imposible utilizarlo para decir algo claro o útil; por tanto, no vale la pena intentarlo. Conviene usar términos más específicos y aclaratorios. Eso es lo que yo hago. No deseo comentar nada que incluya la expresión "*Cloud Computing*", porque pienso que es necesario discriminar según las características de quien tenga el control de los datos, el control de la computación... Hay diversos escenarios que merecen opiniones diferentes y me gusta opinar sobre cosas concretas.

¿Por ejemplo?

Por ejemplo, del software como ser...vicio (y hace una pausa para que se entienda el juego de palabras). No es servicio sino ser-vicio: envías tus datos al servidor de alguien, el cual hace tu informática y después te devuelve los resultados. Pierdes totalmente el control de tu informática. No se debe hacer.

Planteas un mundo donde, al ceder nuestros archivos a un servidor que no sabemos dónde está para que los gestione y nos los devuelva bajo petición ...

¡No! ¡Eso es un error! ¡No lo hagáis!

En definitiva, ¿es posible eliminar completamente el software privativo?

Pienso que todo el software ha de ser libre, pero no propongo eliminar todo el software privativo ahora y de golpe, ya que prohibir una cosa que mucha gente quiere hacer causa problemas sociales. Por eso propongo tomar medidas para fomentar la migración de la sociedad hacia el software libre. Por ejemplo, el Estado no debería poder comprar software privativo. Es más: si el Estado subvenciona el desarrollo de un programa, se le debería de proporcionar al Estado como libre (y, claro está, debería de poder funcionar en un entorno totalmente libre).

Hay quien te tilda de utópico ...

Yo no propongo maneras de llegar a un mundo perfecto, sólo propongo eliminar un problema artificial. Y eso es necesario y posible.

Conclusiones

Nuestro encuentro con Richard Stallman nos confirma que es un personaje peculiar, con sus extravagancias y sus gestos característicos, pero con un pensamiento claro y estructurado. Se puede estar más o menos de acuerdo, pero no se le puede descartar por utópico o extremista; bien al contrario, es realista y sus propuestas son viables. Los profesionales relacionados con la ingeniería informática debemos valorar sus propuestas e incorpo-

rarlas dentro de nuestra perspectiva, coincidiendo en que el movimiento del software libre que Stallman proclama tiene un papel relevante en el futuro de la informática.