

NEW LEFT REVIEW 113

SEGUNDA ÉPOCA

NOVIEMBRE - DICIEMBRE 2018

ARTÍCULOS

GÖRAN THERBORN	La nueva Suecia	7
DAVID KOTZ	¿Recuperación estadounidense?	31
PERRY ANDERSON	Una tarde con Althusser	61

ENTREVISTA

RICHARD STALLMAN	Hablando con el cartero	71
------------------	-------------------------	----

ARTÍCULOS

ALICE BAMFORD Y DONALD MACKENZIE	Contraformatividad	99
-------------------------------------	--------------------	----

CRÍTICA

DYLAN RILEY	Una metafísica para Occidente	127
ZÖE SUTHERLAND	La obra de arte como crítica	143
JOHN GRAHL	¿Más allá de la redistribución?	153

WWW.NEWLEFTREVIEW.ES

© New Left Review Ltd., 2000

Licencia Creative Commons

Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional (CC BY-NC-ND 4.0)

INSTITUTO
25M
DEMOCRACIA

ts
d traficantes de sueños

[SUSCRÍBETE](#)

RICHARD STALLMAN

HABLANDO CON EL CARTERO

Entrevista de Rob Lucas

Estás reconocido como el principal abanderado mundial de la libertad de software después de haber dirigido el desarrollo del sistema operativo GNU. Actualmente, el sistema operativo GNU/Linux, y de manera más general el software libre, fundamenta gran parte de Internet aunque hayan surgido nuevas estructuras que pueden ejercer un gran poder sobre los usuarios. Nos gustaría hablar contigo sobre el actual panorama informático y la relevancia política del software libre. Pero podemos empezar preguntándote por tu formación, como programador y como intelectual. ¿Cómo empezó todo?

NACÍ EN 1953 y crecí en Manhattan. Tenía problemas de comportamiento —no podía ir a una escuela pública sin meterme en líos— y empecé a trabajar con ordenadores desde muy joven. En 1969, durante mi último año en la enseñanza secundaria, un laboratorio de IBM me abrió las puertas dejándome utilizar sus ordenadores. Estuve trabajando allí durante el verano de 1970. Me dieron un proyecto, que consistía en desarrollar un determinado algoritmo para ver cómo podía funcionar. Lo acabé en unas semanas así que me dejaron pasar el resto del verano cobrando por hacer lo que yo quisiera. Fui a Harvard para estudiar física y seguí programando allí. Al final de mi primer año empecé a visitar laboratorios de informática para examinar sus manuales y ver en qué se diferenciaban sus ordenadores. Cuando visité el Artificial Intelligence Lab (AIL) del MIT, no tenían nada que se pareciera a un manual porque habían desarrollado su propio sistema de tiempo compartido. El director del centro decidió contratarme de manera más o menos inmediata, así que aunque me gradué en Harvard en 1974, real-

mente llevaba trabajando en el MIT desde hacía tres años. El ordenador de Harvard era mucho mejor para jugar que el de IBM, pero no tenía mucha memoria, mientras que el ordenador del MIT en el AIL tenía mucha. Pero además me dejaron cambiar el sistema de tiempo compartido; de hecho, ese fue mi trabajo, me contrataron para trabajar en ese sistema. Yo añadí montones de funciones a montones de programas diferentes, cualquier cosa que se me ocurriera, o que la gente me sugiriera y me pareciera una buena idea, la llevaba a la práctica y la gente la utilizaba. Eso era un verdadero placer; resultaba gratificante hacer cosas que la gente utilizaba y apreciaba, así que me quedé trabajando allí. A partir de ese momento me dediqué a programar utilizando el equipo del MIT.

¿Hubo alguien en el MIT que influyera en cómo aprendiste a programar?

Allí encontré a Richard Greenblatt, que más tarde puso en marcha el proyecto Lisp Machine; y tuve cierta relación con Don Eastlake y con Bill Gosper, aunque su inspiración se concretaba más en temas de *hackeo* y matemáticas que en cómo programar. Había un montón de gente inteligente allí. Pero también me resultaba motivadora la actitud que había en el laboratorio del MIT: los *hackers* decían: «No vamos a dejar que los administradores nos digan cómo hacer la cosas; vamos a trabajar sobre lo que ellos necesitan, pero nosotros decidiremos cómo y no dejaremos que pongan en práctica una seguridad informática con la que limitarnos». Esta era una decisión consciente de los *hackers* que habían escrito el sistema de tiempo compartido puesto en marcha un par de años antes de mi llegada. Su actitud era decir, sí, los administradores podrían despedirnos, pero no vamos a lamerles el culo. No iban a aguantar que se les tratase como empleados normales. Yo no hubiera tenido la fuerza necesaria para hacer eso por mi cuenta, pero como parte del equipo aprendí a hacerlo. Éramos los mejores y la mayoría de nosotros no cobrábamos cantidades extraordinarias, cualquiera podría haber tenido un empleo mucho mejor pagado en otro sitio si hubiera querido. Estábamos allí porque éramos libres para mejorar el sistema y hacer cosas útiles de la manera que queríamos, y no para que nos trataran como gente que tenía que estar obedeciendo todo el rato.

¿Y esto tenía el consentimiento de los directores del AIL?

Hasta cierto punto. Los directores del Laboratorio, Marvin Minsky y Patrick Winston, estaban plenamente satisfechos con eso. Me dijeron que a Minsky no le gustaba cerrar las puertas, porque tenía tendencia a perder

las llaves. Así que las puertas del Laboratorio y las oficinas que había dentro estaban siempre abiertas. No había contraseñas para el sistema de tiempo compartido. No había protección para los archivos; realmente cualquiera podía sentarse en cualquier consola y hacer cualquier cosa.

¿En qué medida piensas que el ethos colectivo del AIL en aquél momento —que es una característica sobresaliente de esta historia y se cita habitualmente en los relatos sobre los orígenes del software libre y de «código abierto»— se basaba en el hecho de que estabais trabajando con una clase diferente de tecnología?

Se debía al hecho de que utilizábamos un ordenador compartido. El PDP-10 tenía el tamaño de una habitación y valía un millón de dólares, así que obtener otro no era fácil. El sistema de tiempo compartido se puso en marcha en la década de 1960, pero incluso en la de 1980 nadie pensaba que podríamos permitirnos tener un ordenador para cada uno, no un verdadero ordenador. Sí, estaban esos PC de juguete, pero ¿qué podías hacer con ellos? La cuestión es que cuando la gente comparte un ordenador o bien lo hace como una comunidad en la que confían unos en otros y donde resuelven las disputas, o se dirige como un Estado policial, donde hay unos pocos que son los amos y ejercen un poder absoluto sobre todos los demás.

Entonces, ¿estás de acuerdo en que los orígenes del movimiento por el software libre tenían algo que ver con la utilización compartida de un ordenador?

Sí y no. En el laboratorio del MIT los hackers eran los autores del *software* y también estaban a cargo de la máquina. Quizá guiados por el espíritu de Marvin Minsky desarrollamos una cultura de dar la bienvenida a todos los que venían a trabajar y a compartir. Así que nos opusimos a las medidas de seguridad. A través del sistema cualquiera podía entrar en el terminal de los demás. Si tenías un verdadero trabajo que hacer no hacías eso demasiado, porque estabas ocupado. Pero los chicos, los quinceañeros que llegaban a través de Internet —entonces era Arpanet— podían ver y aprender cosas. Podían observar a los demás y ver si alguien estaba causando algún daño.

¿Así que podían ver a gente programando?

Sí, podían hacerlo. Nuestra manera de tratar a los chicos que llegaban a través de Arpanet era socializarlos. Todos participábamos en eso. Por ejemplo, había un comando mediante el cual podías hacer que el sistema

se cerrara en cinco minutos. Los chicos algunas veces lo utilizaban y cuando lo hacían simplemente cancelábamos el cierre. Se quedaban asombrados. Se enteraban de ese comando, pensaban que seguro que no iba a funcionar y lo ejecutaban. Inmediatamente recibían una notificación: «El sistema se cerrará en cinco minutos debido a...».

Parece un caos

Pero no lo era. Siempre había un verdadero usuario que simplemente cancelaría el cierre y le diría a esa persona: «¿Por qué tratas de cerrar la máquina? Sabes que la estamos utilizando. Solamente se hace eso si hay una buena razón». Y la cuestión es que muchas de estas personas se sentían al margen de la sociedad, eran *geeks*; sus familias y sus compañeros de estudio no les entendían, no tenían a nadie. Y nosotros les dábamos la bienvenida a la comunidad y les invitábamos a aprender y a empezar a hacer algún trabajo útil. Les resultaba asombroso que no se les tratara como basura.

¿Y estos chicos llegaban principalmente a través de Arpanet?

Unos cuantos lo hacían físicamente, pero la mayoría llegaba a través de Arpanet.

Una cosa que resulta sorprendente sobre esa cultura –que es legendaria en la historia de la programación– es que surgió en una institución que en gran medida estaba financiada por el Departamento de Defensa. ¿No te parece paradójico que esta clase de libertad pudiera desarrollarse bajo el caparazón del Pentágono?

Resulta paradójico, pero realmente encaja perfectamente. Ellos querían financiar una investigación. No necesitaban que la realizaran unos patanes a los que se puede pisotear, simplemente querían que se realizara. Durante la década de 1970 había algunos *hackers* del AIL que estaban preocupados por el hecho de que el laboratorio estuviera financiado por los militares. Yo pensaba que lo importante era lo que estábamos haciendo, no de dónde viniera el dinero, y en un momento determinado llegué a la conclusión de que la financiación de las empresas era mucho peor.

¿Peor que la financiación de los militares?

Mucho peor, porque las empresas tratarían de limitar la utilización de lo que tú hicieras.

No obstante, se trataba del Estado que estaba bombardeando Vietnam.

Así es. Yo estaba en contra de la guerra en Vietnam, igual que todo el mundo en el laboratorio, pero nosotros no estábamos *ayudándoles* a bombardear Vietnam. Nuestro trabajo no estaba especialmente dirigido a los militares, ni siquiera era probable que se utilizara a corto plazo. Por ejemplo, Greenblatt trabajaba mucho sobre programas de ajedrez; yo trabajaba sobre todo en la mejora de varios programas del sistema, durante esa época desarrollé el primer editor de texto Emacs.

¿En qué momento cambió esa cultura y cómo lo hizo?

A finales de la década de 1970, incluso en el AIL, había presiones de la dirección para intentar controlar las cosas, algo a lo que me resistía explícitamente. Una vez me encontré en el ascensor con un directivo que había establecido algunos formularios que supuestamente teníamos que rellenar todos los usuarios. Yo no lo había hecho. «Parece que no has rellenado los formularios de usuario». Le contesté algo así como: «No, no veo razón para hacerlo». A lo cual me respondió: «Bueno, deberías hacerlo, de lo contrario alguien puede borrar tu directorio si no está claro cuál es su objetivo». Le repliqué: «Eso sería una pena, parte del código fuente del sistema se encuentra en mi directorio. Sería un problema para el Laboratorio si se borrara». La cuestión es que yo podía hacer cosas y él no. Pero en la década de 1980 alguien insistió en establecer contraseñas en las máquinas de tiempo compartido del Laboratory for Computer Science (LCS) del MIT. Mi sueldo lo pagaban a medias el AIL y el LCS, pero en aquel momento me quedé con el AIL y dejé de hacer cosas para el LCI.

¿El resto del personal tuvo la misma reacción?

Muchos sí, otros se cansaron y finalmente se rindieron. Durante la década de 1970 el espíritu de resistencia se fue agotando en algunos, mientras que en mi caso se fortaleció.

¿Cómo lo explicas?

No lo sé. Supongo que había encontrado algo a lo que merecía la pena seguir siendo leal y fundamentalmente no tenía otra cosa.

¿Fue también en ese momento cuando los proyectos comerciales empezaron a dividir a los programadores del AIL?

Fue un proceso que se produjo a lo largo de varios años. Greenblatt puso en marcha el proyecto Lisp Machine. Lisp era un lenguaje de programación especialmente potente en el que el programa tiene una representación simple y natural como datos con los que otros programas pueden operar. Todo el mundo en el Laboratorio estaba de acuerdo que sería estupendo crear una compañía para hacer Lisp Machines, para que otra gente también las pudiera tener, pero había desacuerdos sobre cómo hacerlo. Greenblatt quería crear una compañía sin inversores externos pero el resto de *hackers* cuestionaba su perspicacia empresarial. Así que trajo a Russell Noftsker, el hombre que me contrató a mí en 1971 cuando era director del AIL. Noftsker decidió deshacerse de Greenblatt y crear una compañía con inversores de la manera habitual. Así que alrededor de 1980 se crearon dos compañías rivales, la segunda, Symbolics, apuñalando a Greenblatt por la espalda. Yo no quise unirme a ninguna de las dos, simplemente quería seguir trabajando en el AIL que para mí era lo ideal.

¿Hubo dinámicas comercializadoras similares que afectaran a la cultura hacker en otras partes, por ejemplo en Standford?

No lo sé. Visité Standford, pero no estuve lo suficiente como para saber cómo iban las cosas allí.

¿Cómo afectó la rivalidad de estas dos compañías a la situación en el MIT?

Para empezar, tanto Symbolics como LMI, la compañía de Greenblatt, cooperaron en el mantenimiento del sistema de Lisp Machine en el MIT. Pero en 1982 –precisamente el 16 de marzo, el día de mi cumpleaños– Symbolics anunció que el MIT ya no podía incluir sus cambios en la versión de su sistema, lo que significaba que todo el mundo tenía que elegir bando: el sistema de Symbolics o el sistema del MIT que también utilizaba LMI. Yo, como un personaje neutral al que se había atacado y dado un ultimátum no tenía otra elección honorable que no fuera oponerme a ellos. Siempre consideré aquello como una guerra y pensé que mi papel estaba en la retaguardia. Mi objetivo era mantener viable la versión del sistema del MIT durante el tiempo suficiente para que LMI evitara ser destruida por Symbolics, como era su objetivo. Desde luego todas sus mejoras las estaban poniendo a disposición del MIT, pero yo no

quería leer su código y a continuación escribir algo similar. Así que veía su documentación y después ponía en marcha una función comparable, no necesariamente la misma, porque tenía oportunidad de mejorarla y a menudo encontraba una manera de hacerlo.

Excepto que ellos eran una docena y tú estabas solo.

Bueno había seis personas importantes, pero sí, tenía que igualar lo que ellos hacían, así que trabajé muy duro. Pero eso me hizo ver que podía hacer algo grande como el GNU.

Bien. ¿Así que en cierto sentido aquello resultó educativo?

Sí. Fue un aprendizaje para concentrarme por completo en crear funciones de *software*, en hacerlas lo más limpias que pudiera y hacer que funcionaran razonablemente pronto. De cualquier modo, aunque así no podían destruir a LMI podían trabajar en otra generación de ordenadores, algo que LMI no podía. Y en 1983 empezaron a llegar sus ordenadores, sus versiones más recientes, y el sistema del MIT no podía utilizarse en ellos. Vi que no podría continuar con lo que estaba haciendo, pero también que había hecho suficiente: había sido una acción de demora que había tenido éxito, había permitido a LMI mantenerse y empezar a desarrollar otro modelo de ordenador contratando a gente que creara el *software* para él. Entonces decidí que no quería pasarme el resto de mi vida castigando a Symbolics por su agresión, quería tratar de construir un sustituto para lo que ellos habían destruido.

¿Te refieres a la cultura en el AIL?

Sí, a la cultura, pero por encima de todo a la comunidad *hacker*, y a un sistema que podíamos desarrollar y compartir libremente. Pero tener una comunidad de *software* libre depende de tener un conjunto de *software* libre que puedas utilizar y por eso es por lo que tuve que desarrollar otro sistema operativo. La única manera de que hubiera un sistema operativo libre era que alguien lo escribiera para los ordenadores modernos, para los ordenadores normales más que para los especiales, como el Lisp Machine. Durante la década de 1980, montones de gente estaban comprando ordenadores IBM. Aunque en aquella etapa todavía eran débiles, me di cuenta de que los ordenadores del futuro podrían ejecutar el sistema que yo iba a desarrollar. Eso significaba que lo mejor era hacer un sistema compatible con Unix, lo que me dio los parámetros básicos del GNU.

Resulta interesante que lo describas principalmente en términos de crear una comunidad en vez de crear una tecnología.

La comunidad necesita la tecnología, no puedes tener una comunidad en la que tu modo de vida sea el *software* compartido a no ser que tengas *software* libre para hacer todo. La cuestión es que el *software* tenía el propósito de hacer que la comunidad fuera posible. Pero parte de esa idea era que yo quería que todo el mundo formara parte de esa comunidad: el objetivo era liberar a todos los usuarios de ordenadores. Al mismo tiempo recibía una lección tras otra sobre la injusticia del *software* no libre. El MIT había comprado una nueva máquina que utilizaba el sistema Digital de tiempo compartido, TWENEX, en vez del que nosotros habíamos estado desarrollando; tenía funciones de seguridad que permitían que un grupo de usuarios se apoderara del control de la máquina y se lo negara a otros. Vi cómo se introducían en Harvard las reglas represivas para los estudiantes de informática. Yo sugerí que se asignara un ordenador a cada grupo de estudiantes que vivieran juntos y que se les dejara hacerlo funcionar; los que estuvieran interesados desarrollarían los conocimientos de administración del sistema, y todos aprenderían a vivir juntos como una comunidad resolviendo sus propias disputas. En vez de ello me dijeron: «Hemos firmado contratos de *software* privativo, que dicen que no estamos autorizados a dejar que ninguno de los estudiantes acceda a ellos». Se trataba de un sistema operativo privativo que permitía tener programas que la gente podía manejar pero que no podían leer. Eso me enseñó que el *software* no libre era un factor que creaba un Estado policial dentro del ordenador, algo que en el AIL se daba por supuesto; no fui el primero en llamar a eso «fascismo». También fui la víctima de un acuerdo de confidencialidad, que me enseñó la naturaleza de esos acuerdos: son una traición para todo el mundo¹.

El sistema operativo GNU/Linux constituye ahora una parte muy importante de las infraestructuras informáticas globales y tiene incontables contribuyentes a escala mundial. ¿Cómo empezaste a escribir el GNU, hasta qué punto fue una empresa colectiva?

Al principio no participaba demasiada gente, pero poco a poco se fueron uniendo más personas. Me dirigí a varias empresas para ver si podían

¹ Véase Sam Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*, Sebastopol (CA), 2002, (también disponible *online* en una edición revisada por Stallman).

financiar el esfuerzo, pero ninguna lo hizo. En septiembre de 1983 yo había anunciado el proyecto GNU en Usenet, un sistema anterior a Internet para noticias en red establecido por AT&T. Inicialmente solo hubo una persona que quisiera trabajar, así que cuando empezamos en enero de 1984 éramos solamente dos. En aquél momento renuncié formalmente a mi trabajo en el MIT, pero el director del Laboratorio, Patrick Winston, me ofreció que pudiera utilizar las instalaciones del laboratorio para el trabajo de desarrollo, lo que fue una ayuda. En 1985 creamos la Free Software Foundation y publicamos el Manifiesto de GNU. Eso atrajo a más voluntarios. Conseguir recambios para todos los componentes de Unix era una gran tarea y tenía que encontrar gente que hiciera cada parte. Esta etapa diría que finalizó alrededor de 1990, cuando alcancé cierta fama, recibí un premio y conseguí más atención.

Esto coincidió más o menos con el desarrollo de la actual Internet, haciendo referencia a la publicación de la propuesta de Berners-Lee para la red mundial. ¿Qué explica la atención que recibiste en aquél momento, la atracción para los programadores del movimiento del software libre?

Habíamos redactado la Licencia Pública General (LPG) de GNU que garantiza que los usuarios finales tengan libertad para ejecutar, estudiar, compartir y modificar el programa del *software*. Fue la primera licencia *copyleft* para uso general, lo que significaba que el *software* que se derivara de ese programa tenía que ser distribuido bajo las mismas condiciones. Pero, sobre todo, estábamos produciendo un *software* que montones de programadores encontraban útil, porque era más fiable que las alternativas comerciales privativas que había. Alguien hizo un experimento probando los diversos programas –versiones de Unix o las alternativas libres que proponíamos nosotros– para ver si el programa colapsaba. Los nuestros fueron los más fiables.

¿Puedes explicar por qué eran tan fiables?

Creo que era porque realmente nos preocupábamos y arreglábamos los errores que nos señalaban los usuarios. Pero no solo eso: los usuarios podían enviarnos correcciones además de informes sobre determinados errores. Teníamos la política de agradecer a los usuarios por los informes de errores. El momento clave lo superamos en 1992 cuando teníamos un sistema completo utilizando el núcleo diseñado por Linus Torvalds. Empezamos a desarrollar un núcleo en 1990, pero el diseño

que elegí se convirtió en un proyecto de investigación, nos llevó seis años obtener una versión de prueba. De modo que Linux es el núcleo que por lo general utilizamos con el GNU.

¿Y Torvalds utilizó afortunadamente una licencia GNU para hacerlo?

No al principio. En 1991, Linux era un programa privativo. En 1992 lo liberó bajo las condiciones de la LPG de GNU.

¿Le convenciste?

No, no lo hice. En aquél momento no había oído hablar de él. Pero creo que él había asistido a una conferencia mía en la University of Technology de Helsinki. Cuando liberó el núcleo lo pudimos utilizar bautizando esa combinación como GNU/Linux.

¿Cuál es tu definición de software libre?

El *software* libre es un *software* que respeta la libertad y la comunidad del usuario. No es una cuestión de precio. Es *libre*, no gratis. Con cualquier programa hay dos posibilidades: o bien el usuario controla el programa o el programa controla al usuario. Cuando los usuarios controlan el programa tenemos un *software* libre, ellos controlan las cosas que hacen con él y de ese modo el programa respeta su libertad y a su comunidad. Si no tienen pleno control sobre el programa, entonces es un *software* que domina al usuario, un *software* privativo no libre. El programa controla a los usuarios y los propietarios del programa controlan el programa de manera que se convierte en un injusto instrumento de poder del propietario sobre los usuarios. Para que estos tengan el control necesitan cuatro libertades específicas, el criterio concreto de lo que es el *software* libre. La Libertad Cero es la libertad para ejecutar el programa como tú quieras, para el propósito que tengas. La Libertad Uno es poder estudiar el código fuente del programa y cambiarlo de manera que puedas ejecutar el programa de la manera que quieras.

¿Qué pasa con la gente que no puede hacer eso, con los usuarios normales que no son programadores?

No pienso que todo el mundo tenga que aprender a programar, no todo el mundo tiene capacidad para ello. Pero aun así merecen tener el control

de sus actividades informáticas. Eso solo lo pueden obtener mediante el control colectivo, que implica el derecho de los usuarios de trabajar juntos para ejercer el control sobre lo que el programa hace para ellos. De modo que para el control colectivo son necesarias más libertades. La Libertad Dos es la libertad para hacer copias exactas de un programa y pasarlas o venderlas a otros. La Libertad Tres es poder hacer copias de tus versiones modificadas y pasarlas o venderlas a otros. Esto hace posible que los usuarios trabajen juntos, porque uno de ellos puede hacer una versión modificada de un programa y distribuir copias a los demás, y ellos pueden hacer copias y pasarlas a su vez. Eso es el *software* libre y todo el *software* debería ser libre, porque siempre habría que respetar la libertad del usuario. Cualquier programa no libre es una injusticia. El hecho de que exista es un problema social y ético para la sociedad y el objetivo del movimiento a favor del *software* libre es acabar con ello.

Tú has sostenido que este énfasis en la libertad es lo que diferencia radicalmente a tu movimiento del llamado movimiento de «código abierto», que empezó más tarde.

El código abierto es un sustituto amoral, despolitizado, del movimiento por el *software* libre y desde el principio tuvo esa intención. Fue una campaña de reacción puesta en marcha en 1998 por Eric Raymond –había escrito «The Cathedral and the Bazaar»– y otros, para contrarrestar el apoyo que estábamos obteniendo para el *software* libre. Cuando empezó, Eric Raymond me llamó para hablarme sobre este nuevo término y preguntarme si quería utilizarlo. Le dije que tenía que pensarlo, pero al día siguiente ya me había dado cuenta de que para nosotros sería un desastre. Significaba desvincular el *software* libre de la idea de que los usuarios merecen ser libres. Así que lo rechacé.

Resulta paradójico en la historia del software libre que su momento de mayor fama estuviera asociado con este término que tú rechazas, «código abierto».

Es que no es el nombre de una filosofía, sino que hace referencia al *software*, no a los usuarios. Puedes encontrar montones de organizaciones, cautelosas y tímidas, que hacen cosas que son útiles, pero que no se atreven a decir que los usuarios merecen ser libres. Un ejemplo es Creative Commons, que hace un trabajo útil y práctico, sobre todo elaborando licencias que respetan la libertad de compartir. Pero Creative Commons no dice que los usuarios *tienen derecho* a la libertad de compartir; no

dice que está mal negar a la gente la libertad de compartir. No defiende activamente el principio. Desde luego es mucho más fácil ser partidario del código abierto, porque no te compromete a nada. Podrías pasar diez minutos a la semana haciendo cosas que contribuyan al avance del código abierto, o simplemente diciendo que eres partidario de él y no serías un hipócrita porque no puedes violar tus principios si no has establecido ninguno. Lo que es significativo es que, en su intento de separar nuestro *software* de nuestras ideas, han reducido nuestra capacidad de convencer a la gente mostrando lo que estas han alcanzado. Gente que no está de acuerdo con nosotros –gente que cree que lo que importa es lo que es conveniente– tienen derecho a presentar sus propias opiniones. Pero lo hacen de una manera que también nos tergiversa a nosotros y creo que eso es injusto.

En ese sentido parece una cierta clase de recuperación política.

No utilizaría ese término. Recuperarse significa restablecerse de alguna clase de enfermedad y no veo quién se ha recuperado aquí. Yo lo llamaría cooptación, han cooptado nuestro trabajo. Lo han hecho intencionadamente y en gran medida han tenido éxito. Lo hubieran conseguido por completo, si no nos hubiéramos defendido.

¿Podrías explicar en concreto por qué el software no libre es injusto para los usuarios?

El simple hecho de que los usuarios no puedan agregar o cambiar funciones ni arreglar errores es una injusticia. Los usuarios de versiones antiguas de programas que ya no se actualizan se ven obligados a cambiar a una versión más reciente, lo quieran o no. El *software* no libre, privativo, es mucho más probable que contenga *malware*, funciones maliciosas de las que hay muchas clases. Los programas no libres pueden espiar a los usuarios, informar sobre ellos. Muchos están específicamente diseñados para limitar lo que los usuarios puedan hacer, ese es su propósito. Los partidarios de estas funcionalidades maliciosas tienen un término para esto, lo llaman «gestión de derechos digitales», GDD. Ese es un término para la propaganda, nunca lo utilizo. Yo lo llamo Gestión de Restricciones Digitales.

Después están las puertas traseras, que significan que alguien puede mandar al programa una orden desde el exterior y decirle que haga algo

con el usuario, como hizo Microsoft cuando obligó a los usuarios a actualizarse a Windows 10 –cuando su actualización o no actualización es una cuestión de opiniones– y luego hizo imposible cancelar la operación. Esto se hizo aparentemente a través de órdenes mandadas a Windows por Microsoft, que esencialmente posee los ordenadores de los usuarios porque puso una puerta trasera universal en Windows. Igual que un ordenador es una máquina computadora universal porque, con el programa apropiado, hace cualquier computación, estas puertas traseras son universales porque, instalando a la fuerza el código apropiado, pueden hacer cualquier cosa con el usuario.

Otra forma de funcionalidad maliciosa es atar el programa a un servidor remoto específico, como ha sucedido con la llamada «Internet de las cosas» y a la que yo llamo «Internet de los pinchazos». Si cierran el servidor, el producto –la nevera o el sistema de calefacción– deja de funcionar. Algunas veces estos productos tienen una puerta trasera universal y la compañía puede cambiar el *software* a la fuerza de manera que los usuarios ya no puedan hacer algo a no ser que vayan a través de una cuenta del servidor de la compañía, lo que significa que les están supervisando. Cualquier cosa para la que tengas que crear una cuenta te está supervisando.

¿Cuál sería tu cronología del desarrollo de estas funcionalidades maliciosas?

Creo que en la década de 1980 los diseñadores de *software* privativo tenían algunas normas éticas, en general podías contar con que ellos no introducirían nada malicioso en un programa y si se encontraba algo de eso se producía un auténtico escándalo. En la década de 1990, el sistema operativo de Windows de Microsoft sí espía a la gente de algunas maneras: informaba a Microsoft sobre los programas que se instalaban. Pero hubo tantas objeciones que Microsoft tuvo que rectificar: en aquél momento también Windows tenía competidores. Una vez que Microsoft estableció un monopolio efectivo consideró que tenía licencia para abusar de los usuarios y reinstaló el *software*. Windows XP tenía una puerta trasera universal cuando fue lanzado en 2001.

Apple y Microsoft, ¿cómo los compararías?

Microsoft y Apple han estado cambiando sus posiciones. Durante mucho tiempo Microsoft fue el principal enemigo de la libertad de los

usuarios y después, durante más o menos los últimos diez años, ha sido Apple. Cuando salieron las primeras iThings, alrededor de 2007, fue un tremendo avance en el desprecio a la libertad de los usuarios, porque imponía la censura de las aplicaciones: solamente podías instalar programas aprobados por Apple. Paradójicamente, Apple ha echado cierta marcha atrás a este respecto. Si un programa está escrito en Swift, ahora lo puedes instalar desde un código fuente. De ese modo, los ordenadores de Apple ya no son cárceles al 100 por 100. Las *tablets* igual. Una cárcel es un ordenador en el que la instalación de aplicaciones está censurada. Así que Apple introdujo el primer ordenador-cárcel con el iPhone. Entonces Microsoft empezó a hacer ordenadores que son cárceles y ahora podrías decir que Apple ha abierto una ventana en la cárcel, pero no la puerta principal. Un estudio de las aplicaciones de los móviles encontró que de media cada aplicación informaba de cien diferentes sitios sobre el usuario, la peor de todas informaba de dos mil².

¿Qué opinas de Google?

Google distribuye *software* privativo –partes de Android son privativas–, que incluye una puerta trasera. Con muy pocas excepciones, todos los servicios de Google exigen utilizar *software* no libre; es Javascript en una página web, pero siguen insistiendo en que utilices en tu ordenador *software* no libre. Y, por supuesto, Google hace otras cosas malas, incluyendo la recogida de montones de datos de los usuarios. Gmail funciona a través de Google Services y Google lo usa para tratar de aprender cosas de la gente.

¿Facebook, Instagram, YouTube, Skype?

Nunca he sido un «utilizado» de Facebook, hablo de «utilizado» en vez de «usuario» porque Facebook los está utilizando *a ellos*. Si la gente me saca fotos les pido que no las cuelguen en Facebook. Por lo que a mí respecta, Instagram es lo mismo, ya que se trata de la misma compañía. Lo malo de YouTube es que tienes que ejecutar *software* no libre para ver algo en esa página de la forma habitual, y no ofrece la opción de descargar nada a no ser que utilices *software* especializado. Skype está diseñado para que Microsoft pueda fisgonear en él. Pero hay alternativas de *software* libre como Linphone, Ekiga, Jitsi. Entre los proyectos de

² Luigi Vigneri *et al.*, «Taming the Android AppStore: Lightweight Characterization of Android Applications», Eurecom Research Report RR-15-305, 27 de abril de 2015.

mayor prioridad de la Free Software Foundation está el desarrollar un chat de audio y vídeo en tiempo real, así como un sistema operativo libre para los teléfonos.

¿Cómo se relaciona la periodización de estos desarrollos tecnológicos con la de la vigilancia del Estado?

Comencé a preocuparme por la vigilancia en la década de 1990, cuando vi cómo los teléfonos móviles estaban siendo vigilados y descubrí el espionaje que se estaba produciendo a través de Windows. A finales de la década de 1990, se amplió la legislación estadounidense que permitía las escuchas telefónicas para incluir a los servidores digitales. Pero el gran cambio vino después de los segundos ataques del 11 de septiembre, esto es, los atentados terroristas perpetrados en Estados Unidos en 2001, no el golpe militar de Pinochet en Chile organizado en 1973 contra Salvador Allende.

En aquel momento la NSA recibió una gran cantidad de dinero para vigilancia digital.

Sí, la *Pat-Riot Act* fue propuesta entonces. Yo siempre divido el acrónimo de esa manera, no voy a aplicar la palabra «patriota» a una ley antiestadounidense que explícitamente autoriza la vigilancia masiva. Cuando un programa espía a sus usuarios y envía los datos a una empresa estadounidense amparándose en esa ley, el FBI puede reunir todos los datos personales sin tener que ir a los tribunales. Peor aún, los datos pueden utilizarse para obtener el perfil de la gente y manipularla.

¿Fue desde este punto de partida cuando los llamados «big data» empezaron a convertirse en algo que podía ser recogido y escaneado?

Puede que esto fuera el comienzo, pero creo que la verdadera recogida de enormes cantidades de información sobre la gente y la manipulación de la tecnología para convertirla en un plan para recoger datos de ella, no empezó hasta aproximadamente cinco años después. Desde luego había otras formas de vigilancia, la vigilancia no solo se realiza a través de *software* no libre. Al rechazar el *software* no libre, algo para lo que tenemos innumerables razones, bloqueamos ciertas vías de vigilancia pero no todas. Por ejemplo, las cámaras de la calle que reconocen las matrículas de los coches y puede que ahora reconozcan las caras,

no podemos bloquear esto insistiendo en tener *software* libre en nuestros ordenadores. La única protección contra esa clase de vigilancia es política. Necesitamos exigirla y hacer campaña a favor de partidos que protejan nuestra privacidad de la opresión del gobierno.

Pero, ¿piensas que la cuestión de la vigilancia se entrecruza con el tema del software libre?

Son temas relacionados, no sé qué significa que se entrecrucen.

De acuerdo. ¿Cómo se relaciona políticamente el movimiento por el software libre con otros temas, tiene aliados naturales?

El movimiento por el *software* libre no exige que tengas ninguna posición política concreta sobre otros temas. Y la Free Software Foundation no toma posición sobre otras cuestiones políticas, excepto para defender los derechos humanos en el terreno de la informática, porque la libertad para controlar tus actividades en ese terreno debe considerarse un derecho humano. Eso también incluye no entregar tu actividad informática al servidor de nadie, porque no puedes controlar cómo se llevará a cabo; esos servicios que te ofrecen realizar tus actividades están invitándote a renunciar a tu libertad. Nos oponemos a la vigilancia general, porque es una violación de los derechos humanos básicos. Pero, por ejemplo, hay personas de derechas que apoyan el movimiento por el *software* libre. Les damos la bienvenida. No estoy de acuerdo con ellos en otras cosas, pero me alegro de contar con su apoyo para defender el *software* libre.

Básicamente, el *software* libre combina ideas capitalistas, socialistas y anarquistas. La parte capitalista es que el *software* libre es algo que las empresas pueden utilizar, desarrollar y vender. La parte socialista es que desarrollamos un conocimiento que se pone a disposición de todos y que mejora la vida de todos. Y la parte anarquista es que con él puedes hacer lo que quieras. Yo no soy anarquista, necesitamos un Estado para que podamos tener un Estado del bienestar. No soy un «libertario» en el habitual sentido estadounidense, a ese tipo de libertario yo lo llamo «antisocialista», porque su principal objetivo es una economía de *laissez-faire, laissez-mourir*. La gente como yo somos los auténticos libertarios. Apoyé a Bernie Sanders para presidente –Clinton era demasiado derechista para mí– y al Partido Verde.

¿Sería correcto decir que no hay una dinámica anticapitalista en el software libre? Después de todo, el propio capitalismo supone excluir a la mayoría de la población de los medios de producción, y el software libre hace que esos medios sean de fácil acceso para cualquiera. El intercambio mercantil es otra cuestión y, por ejemplo, también podría caracterizar al socialismo libertario.

Tal y como entiendo el término «capitalismo», no significa necesariamente que haya cuasi monopolios u oligopolios que dominen políticamente al mundo. Yo condeno enérgicamente el actual sistema de plutocracia. Cuando hablo de capitalismo hablo de empresas privadas. Hay una diferencia entre el sistema económico que tiene Estados Unidos actualmente y el que tenía en 1970. Podrías decir que hay dos formas diferentes de capitalismo, a la actual la llamo capitalismo extremo o plutocracia, que significa que las empresas dominan al Estado. Estoy absolutamente en contra de la plutocracia, pero no quiero identificar al capitalismo con la plutocracia porque en mi vida he visto que hay otras formas de capitalismo. Lo fundamental es que las empresas no deberían decidir nuestras leyes.

Entonces, ¿tu orientación política sería determinado tipo de socialdemocracia dotada de una economía mixta?

Sí. Pero esto atañe a la economía y mi orientación política es la democracia y los derechos humanos. Pero tener una democracia significa que la gente controle al Estado, lo que implica que las empresas no lo hagan. Así que en términos de economía, sí, estoy a favor de contar con empresas privadas, no creo que unos restaurantes públicos nos pudieran dar las comidas que hemos tenido ahora.

Esto plantea la cuestión de cómo se relaciona el software libre con cuestiones más amplias de control y propiedad.

Control y propiedad no son lo mismo. Hay quien propone una solución mercantil-capitalista para la recogida y vigilancia de los datos, que supone que la gente debería «poseer» sus datos y cobrar una pequeña cantidad por entregarlos a una empresa³. Esa es una completa trampa y no resuelve ningún problema práctico. Otro enfoque, adoptado por la nueva Directiva Europea de Protección de Datos, es exigir que la gente otorgue su «consentimiento» para que se recojan sus datos. Eso

³ Véase, Rob Lucas, «Xanadu como falansterio», *NLR* 86, mayo-junio de 2014.

ayudará contra algunas cosas, especialmente contra la recogida de datos por parte de sistemas que tú no sabías que estabas usando, o nunca aceptaste usarlos. Pero para ellos es tan fácil obtener el consentimiento formal que se trata de una barrera inútil. Cualquier página que tenga un botón de «Me gusta» de Facebook, o un rastreador de Google Analytics, solo necesita incluir en sus Términos y Condiciones –algo que nadie lee– una declaración que diga: «Consiento que mi visita sea registrada por otras páginas que tengan un acuerdo con esta». Yo llamo a eso «consentimiento industrial». Las regulaciones que exigen el consentimiento pueden ser útiles contra Facebook, porque ha estado supervisando durante años a los visitantes de miles o millones de páginas web y no se ha molestado en obtener esa clase de consentimiento, así que Facebook tiene problemas si utiliza esos datos. Pero en el futuro probablemente eso sea una simple restricción formal y no una cuestión de fondo. No deberíamos permitir ninguna recogida de datos que vaya más allá del mínimo que es inevitablemente necesario para que la página haga su principal trabajo y deberíamos desarrollar una tecnología para reducir aún más ese mínimo.

¿Supone ello que, idealmente, la gente intervendría más en sus propios dispositivos?

Sí, debería hacerlo. Las personas que no son programadores es probable que piensen: «Bueno, yo nunca tendré el control, no sé cómo cambiar estos programas». Y eso puede ser cierto, ellos nunca cambiarán directamente los programas. Pero cuando estás utilizando un programa libre para hacer el trabajo detrás hay una comunidad de muchos usuarios y algunos de ellos son programadores. *Ellos* pueden corregir errores y *ellos* pueden añadir funciones; eso significa que el programa nunca quedará desfasado, porque la comunidad de usuarios será capaz de arreglar los problemas y tú te beneficiarás de ello. Y no solo eso, el hecho de que el programa sea libre supone una poderosa disuasión contra las funcionalidades maliciosas: los programadores son conscientes de que no tienen ningún poder sobre los usuarios y eso significa que no se corrompen de la manera que lo hacen los que desarrollan *software* privativo.

Esto nos lleva de vuelta a la cuestión de la propiedad y el control. Con el *software* libre siempre posees tu propia copia que supone el control sobre ella. No posees el programa en sentido abstracto, bueno, si tú lo has escrito sí tienes la propiedad. Pero normalmente el programa ha

sido escrito por otros de manera que, aunque no te pertenezca, sí te pertenece tu copia. En cambio, con el *software* privativo, los promotores normalmente dicen que los usuarios *nunca* pueden tener la propiedad de una copia. El *software* privativo lleva típicamente lo que yo llamaría un contrato antisocializador, lo que normalmente se conoce como acuerdos de licencia de usuario final. Nosotros hablamos de la socialización de los niños, de enseñarles a ser amables con otra gente, a ayudar y cooperar con otros. Bueno, este contrato supone exactamente lo contrario, la gente tiene que comprometerse a *no* hacer copias para otras personas, a *no* prestarlo o pasar sus copias a otros. Nunca he estado de acuerdo con semejante cosa. Resulta triste decirlo, pero la mayoría de la gente lo ha aceptado sin pensar en lo que estaban cediendo.

¿Eso sugiere que debería haber cierta clase de imperativo para que todos los usuarios se comprometieran con actos de desobediencia civil?

Absolutamente. Pero, aparte de ello, los gobiernos deberían aprobar leyes que dijeran que esa clase de acuerdos no son válidos, que no tiene fuerza legal en esta jurisdicción sin importar cuándo se firmaron. Esto se aplica a contratos no negociados en los que los términos simplemente se imponen: «Si quieres esto, acepta el contrato», no hay posibilidad de negociar y ahí está el gran abuso. Es diferente cuando las partes hablan realmente sobre las condiciones que quieren, en ese caso puede no haber necesidad de restringirlos. Pero cuando el poderoso acosa al débil para que acepte un contrato tenemos que rechazar esa estrategia en el plano moral. Creo que las libertades del *software* libre deberían ser derechos inalienables de todos los usuarios de *software*. Pero quizá también la libertad de dar, prestar o compartir copias de cualquier trabajo publicado debería ser un derecho inalienable. Los gobiernos deberían bloquear o prohibir activamente todos los métodos que *niegan* ese derecho a la gente. Estos contratos antisocializadores son un método, jurídico, para negar esos derechos y debería considerarse que carecen de fuerza legal. En cuanto a la gestión de restricciones digitales, ello debería llevarte a la cárcel durante muchos años, debería ser un delito grave hacer, vender, alquilar o importar productos con ese tipo de restricciones.

En términos materiales, el mayor impacto que ha tenido el software libre en el mundo ha sido su utilización por la comunidad técnica, en servidores y grandes empresas...

Y ese no es mi objetivo. Yo quería dar libertad a los usuarios, no a las empresas. Pero sí, creo que una empresa debería tener el control de su propia actividad informática. Si los ordenadores de la empresa A están controlados por la empresa B eso no hace que el mundo sea mejor. Pero liberar a las empresas de ese abuso no es mi prioridad. Yo quiero liberar a los seres humanos, a la gente.

Pero la cuestión sigue en pie: las distribuciones de GNU/Linux se utilizan ampliamente para servidores web, probablemente una mayoría de los servidores las utilizan. Así que cuando la gente está interactuando con una página web, en algún sentido estará normalmente interactuando con el software libre, pero eso no evita la clase de abusos que tú has estado detallando aquí, puertas traseras, recolección de datos.

Cuando visitas una página web no estás utilizando el *software* que está en el servidor de la página, la organización propietaria de esa página está utilizando ese *software* para hablar contigo. Los expertos en informática están acostumbrados a analizar sistemas diciendo que el hilo de la ejecución puede moverse de un ordenador o un proceso a otro. Técnicamente eso tiene sentido en algunos supuestos, pero no creo que lo tenga para una discusión sobre ética, no lo tiene cuando el supuesto implica a entidades que no pueden confiar las unas en las otras y cuyos intereses están en conflicto. Cuando yo visito tu página web, ello implica a dos entidades, tú y yo. Los programas en el servidor de tu web están trabajando para ti. El navegador y otros programas de mi portátil están trabajando para mí. No se trata de un único sistema, son dos sistemas que se comunican. Desde luego tú no controlas el *software* en la web de alguna otra entidad, ¿por qué deberías hacerlo? ¿Cómo lo podrías hacer? No lo haces y no debes hacerlo, y eso está bien. Yo hablo contigo, pero no estoy usando tu cerebro. Pero estamos hablando y *tú estás* utilizando tu cerebro.

Sin duda es probable que los usuarios que visitan una página web estén interactuando con el ordenador de alguien que está ejecutando *software* libre, y eso no ayuda necesariamente a esos usuarios. Simplemente porque el servidor utilice *software* libre y la empresa propietaria del servidor pueda cambiarlo, no significa que los visitantes reciban un trato ético. Y el que una página web esté en un ordenador que está ejecutando *software* libre no impide que pueda recoger datos de los usuarios. También es muy probable que esté mandando *software* no libre a sus navegadores, para que se ejecute en sus dispositivos. Muchas páginas web lo hacen.

Ese *software* realiza las funciones informáticas del visitante y el visitante no lo controla, porque no es un *software* libre y porque se está ejecutando directamente cuando se envía desde el servidor. Para el usuario no es fácil desarrollar un control sobre ello.

Supongamos que parte del *software* en el servidor de la página web *no* es libre. ¿A quién perjudica esto? A la entidad propietaria de la página web, así que yo le recomendaría que liberara sus páginas web cambiando a *software* libre; eso va en su interés. Pero no tengo ninguna razón para boicotear una página web solamente porque hay *software* no libre ejecutándose en ella, de la misma manera que no tengo razones para boicotear un restaurante porque su caja registradora tiene *software* no libre, yo no utilizo esa máquina.

Así que, de manera más general, ¿cómo deberíamos abordar esa cuestión de las entidades en Internet que bien pueden estar utilizando software libre en sus propios servidores, pero que interactúan con su base de usuarios de maneras muy opacas, manejando mucho poder sobre recogida de datos y todo lo demás?

Recoger datos es una cuestión separada y es una responsabilidad de leyes distintas. El *software* libre evita un particular mecanismo de injusticia, pero no resume toda la ética de la informática. Y esto no debería sorprender a la gente; en la vida hay varias cuestiones éticas. Si te fijas en una tienda, por ejemplo, pueden surgir diferentes cuestiones éticas y no esperas que una norma las soluciones todas. Por ejemplo, la tienda puede abusar del personal. La solución a eso podría ser favorecer la sindicalización o tener leyes estrictas para impedir el robo de salarios. También está la discriminación, que es un abuso hacia aquellos que no pertenecen a la plantilla pero que querían hacerlo. Hay otras cuestiones que solo afectan a los clientes: por ejemplo, ¿los productos son lo que dicen ser? Un sindicato podría asegurar que el personal cobrara sus salarios, pero podría no preocuparse porque se engañe a los consumidores. Hay múltiples cuestiones éticas; no deberíamos esperar que corregir una clase de injusticia automáticamente corrija las demás.

De nuevo, dado que ese es el caso, ¿hay campañas políticas relacionadas con esto que tú apoyarías?

Las que busquen poner fin a la vigilancia masiva que pone en peligro derechos humanos. Para ello necesitamos exigir que todos los sistemas

sean diseñados para limitar la cantidad de datos que puedan recoger. La realidad es que una vez que se recogen los datos, se hará un uso indebido de ellos. La organización que los recoge puede hacer un uso indebido de ellos, los empleados deshonestos de esa organización pueden utilizarlos igualmente; terceras partes, delincuentes informáticos, pueden entrar en el sistema del ordenador, robar los datos y hacer una utilización abusiva de ellos; y el Estado puede tomarlos y hacer un uso indebido de ellos. Si se hicieran cumplir adecuadamente las leyes que limitan el uso de los datos se limitarían los abusos de la organización que los recoge y puede que, en cierta medida, la utilización indebida que pueda hacer el gobierno, aunque no lo suficiente porque el Estado normalmente se otorgará a sí mismo excepciones para hacer lo que quiera. Redactará leyes que le permitan hacerse con esos datos y utilizarlos para encontrar a denunciantes y disidentes.

Por otro lado, si el sistema está diseñado para no recoger datos no puede utilizarse para hacerlo, a no ser que el Estado intervenga para cambiarlo. Así que a no ser que confíes en que el Estado respetará la democracia –lo que difícilmente es el caso– y respete el derecho a disenter, deberías asegurarte que no se está produciendo la vigilancia masiva. He propuesto varias aproximaciones técnicas para diseñar sistemas que recojan muchos menos datos, se pueden ver en la web de GNU⁴. La cuestión clave es cómo nos aseguramos de que cuando haya otro Snowden el Estado no pueda encontrar a esa persona.

¿Estarías a favor de iniciativas legislativas para limitar este tipo de vigilancia?

Legislativas y jurídicas; hay gente que va a los tribunales tratando de detener la recogida de datos o de limitar su uso. Mi punto de vista es que si realmente queremos asegurar nuestra privacidad, tenemos que detener la *recogida* de datos. Las reglas para limitar cómo se puedan utilizar los datos recogidos pueden resultar positivas, pero no son una protección muy sólida. La cuestión de la privacidad es más amplia, por ejemplo, las cámaras que reconocen matrículas y rastrean automóviles. Eso se ha utilizado para reprimir la disidencia. En Gran Bretaña, por ejemplo, había una manifestación en una central eléctrica alimentada con carbón. La policía controló a los coches sospechosos de dirigirse a la central y los mantuvo retenidos en la carretera hasta que se acabó la protesta sin molestarse en

⁴ R. Stallman, «How Much Surveillance Can Democracy Withstand?», disponible en gnu.org; publicado originalmente en *Wired*, 14 de octubre de 2013.

acusarles de nada⁵. Está claro que no eran sospechosos de ningún delito, no había ningún motivo para acusarles de nada. Fue un ataque digital sobre la disidencia, la vigilancia utilizada para sabotear la democracia. Estaría bien crear un archivo de todos estos incidentes.

¿Cuál es tu opinión sobre las plataformas cooperativas, sobre la idea de crear cooperativas alternativas, por ejemplo, a Uber?

En general estoy a favor, pero eso no soluciona automáticamente todos los males. El abuso sobre los trabajadores podría evitarse por medio de una cooperativa de trabajadores, los trabajadores se tratarán a sí mismos todo lo bien que permitan las circunstancias. Pero eso no significa que ellos vayan a tratar éticamente al cliente. ¿Qué hay de malo en Uber? Lo primero es que paga una miseria a los conductores, por lo que yo la llamo Guber [pringado] y eso es una buena razón para negarse a utilizarlo. Pero todavía peor es la manera en que maltrata a los clientes, haciéndoles ejecutar *software* no libre y manteniendo una base de datos del lugar al que ha ido cada pasajero. Uber ha tratado activamente de eliminar las demás alternativas soportando unas pérdidas enormes, con una política de precios bajos dirigida a expulsarlas del negocio. Si eso se sustituyera por una cooperativa propiedad de los trabajadores, ellos podrían seguir dirigiéndola de la misma manera, con un salario mayor para los conductores que entonces serían propietarios, algo que en mi opinión no haría que la empresa fuera más aceptable. No es solo una cuestión de trabajadores frente a directivos, o de que la empresa abuse de sus trabajadores o no, los derechos de los clientes son igualmente importantes y simplemente tener una cooperativa propiedad de los trabajadores no garantiza que se respeten esos derechos.

¿Qué dirías de una cooperativa de trabajadores y usuarios? Ha surgido la idea de que la base de usuarios de Twitter comprara la compañía.

Puede ser. No estoy seguro de que eso tenga sentido aplicado a una empresa de transporte. ¿Qué significa para los clientes de una compañía de estas ser copropietarios? No puedes esperar que todos los pasajeros lo sean. Y si lo fueran, eso no les protegería en absoluto a no ser que insistieran en el anonimato y, actualmente, la mayoría de la gente no piensa lo suficiente en ello como para insistir.

⁵ Paul Lewis y Rob Evans, «Activists repeatedly stopped and searched as police officers “mark” cars», *The Guardian*, 25 de octubre de 2009.

Quizá el ejemplo de Twitter tenga más sentido; se podría imaginar un Twitter que fuera propiedad de sus usuarios y trabajadores. Y se puede imaginar que la comunidad de usuarios fuera capaz de asegurar que, por ejemplo, Twitter ejecutara solamente software libre.

Bueno, pero eso es una cuestión diferente. Desde luego deberían hacerlo para controlar sus propias operaciones. Twitter probablemente ya ejecute en gran medida *software* libre, porque esa es una práctica estándar en los servidores. Probablemente estén ejecutando el sistema GNU/Linux, además de un montón de *software* que ellos han escrito, que es un *software* libre en sentido superficial. Pero como he explicado, eso no es lo que determina el comportamiento justo con los usuarios.

¿Qué piensas sobre la idea de crear alternativas socializadas a los gigantes tecnológicos y a su acumulación de datos, como sugería en estas páginas Evgeny Morozov?⁶

Creo que hay poca diferencia en que los datos sean recogidos por gobiernos o por empresas, de ambas maneras los datos amenazan a denunciantes y disidentes. El Estado puede hacerse fácilmente con los datos que recojan las empresas.

¿Qué pasa con los diversos intentos de crear alternativas federadas a Facebook y Twitter, con el software de comunicación social como GNU social y Mastodon?

No son solo intentos, funcionan. Y por lo que he oído, Mastodon, que es un determinado tipo de producto altamente compatible con GNU social, está empezando a ser bastante popular. Sin embargo, la dificultad de una red social es su «efecto red». Supongamos que Facebook convirtiera todo su *software* en *software* libre: podrías crear un servidor que hiciera exactamente las mismas cosas que los servidores de Facebook, pero inicialmente no tendrías ningún usuario. La cuestión principal que presiona a la gente para que pasen a ser utilizados por Facebook es que la gente que conocen, o con la que quieren comunicarse, también están utilizados por Facebook. Mi análisis ético de un sistema de comunicación dotado de un efecto red es que las personas utilizadas por Facebook son víctimas-corresponsables. Pueden reconocer que Facebook abusa de la gente que utiliza, sin embargo, no se detienen y se convierten en uno de ellos porque otra gente les presiona para que lo hagan. Pero como

⁶ Evgeny Morozov, «¡Socializad los centros de datos!», *NLR* 91, marzo-abril de 2015.

resultado de la rendición, que les convierte en víctimas, también se convierten en parte de la presión sobre otros para que hagan lo mismo. Dicho en otras palabras, se convierten en víctimas corresponsables. Lo mismo sucede con Skype y con cualquier otro sistema semejante que abusa de la gente, pero que permite que las personas se comuniquen entre sí. Normalmente, si estás utilizando un programa no libre, eres una víctima de él, y yo no te lo reprocharía, solamente te animaría para que dejaras de serlo. Pero ahí donde haya un efecto red, diría que es un deber moral dejar de presionar a otros para que utilicen y sean utilizados por ese sistema.

He visto que GNU social está cerca de ser un sustituto de Twitter mientras que Diaspora se parece más a Facebook.

Creo que es así. Estoy a favor de proyectos como estos, porque sé que son útiles para otra gente, pero no encajan en mi modo de vida. Yo solo utilizo el correo electrónico. Me puedes llamar «el Cartero», como en *True Names*, el relato de ciencia ficción de Vernor Vinge.

Ahora se sostiene a menudo que los intentos de frenar la recogida masiva de datos de las corporaciones estadounidenses, en nombre de la privacidad o de las libertades civiles, permitirán a China ganar la nueva «carrera espacial» en el aprendizaje automático y la inteligencia artificial. ¿Qué dices sobre esto?

La libertad y la democracia son más importantes que el avance de la tecnología. Si China y Estados Unidos están echando una carrera en pos de una tiranía orwelliana espero que pierda Estados Unidos. Realmente, Estados Unidos debería abandonar la carrera lo antes posible. A nuestra sociedad se le ha enseñado a sobreestimar la importancia de la «innovación». Las innovaciones pueden ser buenas y también malas. Si dejamos que las empresas decidan qué innovaciones vamos a utilizar, elegirán las que les proporcionan más ventajas sobre nosotros.

*Un razonamiento complementario es que Silicon Valley puede tener ventaja sobre China en términos de innovación, pero que la próxima etapa en el campo de la inteligencia artificial será su puesta en marcha –la cantidad de datos que una organización pueda manejar por medio de algoritmos y la magnitud de su capacidad de procesarlos– y ahí China tiene la ventaja de una población cuatro veces mayor que la de Estados Unidos. (Esto parece ser la afirmación de Kai-fu Lee en *AI Superpowers*). ¿Qué opinas de estos factores?*

¿Cuáles son las implicaciones en términos de las libertades del software, en primer lugar, y, en segundo, de las libertades civiles?

Esa clase de inteligencia artificial trabajará para las empresas; su utilización será para ayudar a las empresas a manipular mejor a la gente y dominar más a la sociedad. Creo que debemos restringir la recogida de datos sobre la gente, independientemente de los acusados señalados por los sospechosos que designen los tribunales. Si eso evita que las compañías desarrollen técnicas de inteligencia artificial que les ayuden a dominar a la sociedad, mucho mejor.

La computación cuántica está siendo publicitada como la próxima innovación digital —«qubits» variables en vez de los ceros y unos de la computación convencional—, que supuestamente abre la puerta para un aumento exponencial de la potencia y del aprendizaje automático. ¿Crees que eso tendrá implicaciones para la libertad del software?

Eso significa que los ordenadores podrían hacer ciertas tareas más rápidamente (no todas). En su mayor parte, eso estaría bien pero fundamentalmente no cambiaría nada. Sin embargo, en un caso concreto podría hacer mucho daño: desvelarían nuestros actuales algoritmos de encriptación de clave pública. Hay gente tratando de desarrollar unos recambios adecuados antes de que los ordenadores cuánticos sean lo suficientemente potentes como para que los utilicen para eso.

Los principios del movimiento a favor del software libre que se aplican a la computación convencional, ¿se aplicarían igual a la computación cuántica?

Sin duda. Son principios generales que no dependen de los detalles del funcionamiento del ordenador.

¿Cuáles piensas que han sido las principales victorias y reveses del movimiento del software libre?

Entre las victorias, el tener un sistema operativo libre, en otras palabras, la existencia de GNU/Linux. Antes de ello era imposible hacer nada en un ordenador sin un *software* privativo. Internacionalmente, ha habido algunas victorias legislativas parciales; varios países de América Latina han aprobado leyes para encaminar a los gobiernos hacia el *software* libre, pero varía mucho la fuerza que tienen. Las que se han aprobado

en Perú son bastante débiles. Argentina no ha aprobado ninguna, pero algunas provincias sí lo han hecho. Ecuador es el mejor ejemplo de un sistema diseñado para provocar un cambio, pero la persona a la que Correa puso al frente del organismo no acabó la tarea. Creo que han cambiado las cosas en los colegios públicos, algo que es muy importante. Algo similar sucedió en Venezuela, solo que allí la política no fue diseñada de manera coherente, algunos funcionarios medios se opusieron, los ministros cambiaban continuamente y los activistas tenían problemas para conseguir apoyo político de estos. En India, el estado de Kerala migró hace una década al *software* libre en algunos niveles educativos, en aquél momento no pudieron migrar los últimos años de la escuela secundaria, porque estaban controlados por un consejo que dictaba los planes de estudios. India adoptó una ley nacional diciendo que tenía que haber una preferencia por lo que ellos llamaban «código abierto», pero de hecho están usando la definición de *software* libre.

¿Qué pasa con el sistema de escuela pública de Estados Unidos?

Simplemente es malo. En cuanto a los reveses, los tres más importantes son la informática móvil, los teléfonos y tablets diseñados desde el principio para utilizar *software* no libre. Las aplicaciones que ahora tienden a ser *malware* no libre y el motor de gestión de Intel, y más en general el *software* de bajo nivel, que no podemos reemplazar porque las cosas no nos permiten hacerlo.

¿Puedes imaginar una situación en la que no hubiera ya ningún software que no fuera libre, o eso es esencialmente una utopía? ¿Hay alguna manera de llegar a ello?

La gente decía que tener un sistema operativo libre era una utopía y una imposibilidad. Sostenían que no merecía la pena ni intentarlo, porque era muy difícil. Pero yo creo que hay un error fundamental en ese planteamiento y es que asume que rendirse estaría bien. Yo no utilizo *software* no libre. No utilizo los servicios que requieren que los usuarios ejecuten *software* no libre. Así que el *software* que tenemos ya es útil, y estoy seguro de que podemos llegar mucho más lejos si lo intentamos que si nos rendimos. No digo que el *software* libre sea más importante que derrotar a la plutocracia, o más importante que frenar el calentamiento global y no intentaría argumentar que la gente debería trabajar en una cuestión en lugar de la otra. Pero tenemos que tener gente trabajando

en este y la gente activa en el campo del *software* no puede eludir el tema del *software* libre frente al privativo, el *software* respetuoso de la libertad frente al que la pisotea. Si estamos haciendo cosas en el campo del *software* tenemos una responsabilidad: hacerlas de una manera que sea ética. No sé si llegaremos a conseguir liberar a todo el mundo, pero claramente esa es la dirección que hay que seguir.

Los textos inmediatamente anteriores publicados en este ciclo «Nuevas masas, nuevo medios» han sido Bhaskar Sunkara, «Proyecto Jacobin» (NLR 90) y Francis Mulhern, «Un partido de recién llegados» (NLR 93).