

EN DEFENSA

FMG

SOFTWARE

LIBRE

N^{RO} 0

En defensa del Software Libre

En Defensa del Software Libre es una revista de teoría sobre Software y Cultura Libres. Se edita en papel al costo de producción y se distribuye gratuita y libremente en formato digital.

<http://endefensadelsl.org>

Copyright © 2010 En Defensa del Software Libre.

Arte de tapa y diseño: Creative Commons

Atribución-CompartirIgual 2.5 Argentina[1].

El material contenido en esta publicación se distribuye bajo diferentes licencias Creative Commons. Para conocer cuáles son las libertades que los autores otorgaron, consultar al pie de cada capítulo.

Índice general

1. Libertad en la nube	1
2. ¡Hackers GNUíños!	17
2.1. La izquierda política y la política de los hackers	17
2.2. La máquina en el trabajo	20
2.3. Descalificación de obreros, recalificación de usuarios .	24
2.4. El software libre como una estrategia sindical	27
2.5. Reconocimientos	34
3. El Software-martillo	35
3.1. Introducción	35
3.2. La ciencia martillo	36
3.3. El software martillo	37
3.4. Conclusiones	40
4. Piratería y Software Libre	43
5. Richard Stallman: Qué es el Software Libre	47

Eben Moglen

1

Libertad en la nube: Libertad del Software, Privacidad y Seguridad para la Web 2.0 y Computación en la Nube¹

Es un placer estar aquí. Me encantaría pensar que la razón por la que estamos aquí en un viernes a la noche es porque mis charlas son muy buenas. En realidad no tengo idea de por qué estamos aquí en una noche de viernes pero estoy muy agradecido por la invitación. Soy la persona que no tenía cita esta noche así que fue bastante conveniente que me invitaran.

Así que, por supuesto, no tenía una cita esta noche. Todo el mundo lo sabe. Mi agenda está en la web.

El problema es ese problema. Nuestra agenda está en la web. Nuestra ubicación está en la web. Tenés un teléfono celular y un proveedor de telefonía celular y si tu proveedor es Sprint entonces podemos decir que varios millones de veces durante el año pasado, alguien con una identificación legal en su bolsillo fue al sitio web de

¹Discurso de Eben Moglen (<http://url.ca/1jy14>) en una reunión de la rama neoyorquina de la Internet Society (<http://www.isoc-ny.org/>) el 5 de Febrero de 2010. La grabación del evento se encuentra disponible en <http://url.ca/1ch5>. Esta es una traducción de su transcripción por Nicolás Reynolds, revisada y corregida por Leonardo G. De Luca, septiembre, 2010. Original disponible en <http://url.ca/uups>. El original y esta traducción se distribuyen bajo licencia Creative Commons Atribución-NoComercial-SinDerivadas 2.0 (<http://creativecommons.org/licenses/by-nc-nd/2.0/>).

Sprint y pidió la ubicación en tiempo real de alguien con un número de teléfono y ésta le fue dada. Varios millones de veces. Así nada más. Sabemos esto porque Sprint admite que tiene un sitio web donde cualquiera con una identificación puede ir y encontrar la ubicación en tiempo real de cualquiera que tenga un teléfono de Sprint. No sabemos eso de AT&T y Verizon porque no nos contaron.

Pero esa es la única razón por la que no sabemos, porque no nos contaron. Ese es un servicio que pensarían como un servicio tradicional, la telefonía. Pero el trato que obtenés con el servicio tradicional llamado telefonía contiene una cosa que no conocías, como el espionaje. No es un servicio para vos pero es un servicio y lo obtenés gratuitamente con tu contrato de servicio telefónico. Obtenés gratuitamente el servicio de publicidad con tu Gmail lo que significa que hay otro servicio detrás que no ha sido tocado por manos humanas, el análisis semántico de tu correo. Todavía no entiendo por qué alguien querría eso. Todavía no entiendo por qué alguien lo usa pero la gente lo hace, incluyendo la muy sofisticada y pensativa gente de esta sala.

Y obtenés servicio de correo electrónico gratuitamente y algo de almacenamiento que vale exactamente una moneda y media al precio actual de almacenamiento y obtenés espionaje permanente.

Y gratis también.

Y tu agenda está en la web y todos pueden ver si tenés una cita el viernes a la noche y tenés un estado ("buscando") y obtenés un servicio gratis, de publicidad ("soltero: buscando"). Espionaje gratis incluido. Y todo esto apareció así en un parpadeo y aquí estamos. ¿Qué tiene que ver con el Código Abierto? Bueno, de hecho no tiene nada que ver con el código abierto pero sí un montón que ver con el software libre. Otra razón por la que Stallman estaba en lo cierto. ¿Es la libertad, no?

Entonces tenemos que volver un poco y pensar en qué punto estamos y cómo llegamos aquí y probablemente algo más importante, si podemos salir y si podemos, ¿cómo? Y no es una linda historia, para nada. David tiene razón. Apenas puedo empezar diciendo que ganamos dado que el espionaje viene gratis con todo ahora. Pero no perdimos. Nos metimos en un lío nosotros mismos y vamos a tener que desliarnos realmente rápido o vamos a meter a otra gente inocente que no sabía que estábamos regalando su privacidad para siempre.

Empieza por supuesto con la Internet, por eso es bueno estar aquí hablando a la Internet Society, una sociedad dedicada a la salud, expansión y elaboración teórica de una red de par-a-par llamada "Internet" diseñada como una red de pares sin ninguna necesidad intrínseca de control jerarquizado o estructurado y asumiendo que cada punto de la Red es una entidad independiente y libre cuya voluntad es equivalente a la voluntad de los seres humanos que la quieren controlar.

Ese es el diseño de la Red, que, aunque estén pensando que está pegada con IPv4 o esa mejora maravillosa que es IPv6 que aparentemente nunca vamos a usar, todavía asume comunicaciones entre pares.

Por supuesto, nunca trabajó realmente de esa manera. No había nada en el diseño técnico para prevenirlo. En ningún punto en el diseño técnico de interconexión de los nodos y su comunicación. Fue un problema de software. Es un simple problema de software y tiene un simple nombre de tres sílabas. El nombre es Microsoft. Conceptualmente, había una red que estaba diseñada como un sistema de nodos pares pero el SO que usaba la red cada vez más - voy a usar la palabra, la usan con nosotros, ¿por qué no puedo usarla contra ellos? - viralmente durante una década y media. El software que empezó a ocupar la red estaba construido sobre una idea muy clara que no tenía nada que ver con pares. Se llamó "arquitectura cliente-servidor".

La idea de que la red era una red de pares fue difícil de percibir después de un tiempo, particularmente si eras, digamos, un ser humano ordinario. Esto es, no un ingeniero informático, científico o investigador. No un hacker, no un geek. Si eras un ser humano ordinario era difícil percibir que la arquitectura que sostiene a la Red estaba pensada para trabajar entre pares porque el SO con el que interactuabas ejemplificaba fuertemente la idea de la arquitectura cliente-servidor.

De hecho, si lo piensan, fue peor que eso. La cosa llamada "Windows" era una versión degenerada de una llamada "X Windows". Esta también pensó el mundo como una arquitectura cliente-servidor, pero lo que ahora diríamos al revés de como la conocemos. El servidor era la cosa del lado del ser humano. Esa era la concepción básica del mundo de X Windows. Servía comunicaciones con seres humanos en los puntos finales de la red a procesos ubicados en lugares arbitrarios cerca del centro, en la mitad, o en los bordes de la Red. Fue la gran idea de Windows en una forma extraña crear un arquetipo político en la red que redujo el ser humano al cliente y produjo una gran y centralizada computadora que podríamos llamar un servidor, que ahora proveía cosas al ser humano en términos de tómallo-o-déjalo.

Eran términos, por supuesto, bastante tómallo-o-déjalo y desafortunadamente, todos lo tomaron porque no sabían dejarlo una vez que entraron. Ahora la red estaba hecha de servidores en el centro y clientes en el borde. Los clientes tenían bastante poco poder y los servidores tenían mucho. Mientras el almacenamiento se abarata, mientras el procesamiento se abarata, y mientras los servicios complejos que sólo escalan en formas en que es difícil usar computadoras pequeñas -o en cualquier caso, estas colecciones agregadas de computadoras- para ello, siendo el más importante de todos la búsqueda. Mientras los servicios empezaban a poblar la red, la naturaleza jerárquica de la red empezó a verse como pensada para que así fuera. La red estaba

hecha de servidores y clientes y los clientes eran los tipos en el borde representando humanos y los servidores eran las cosas en el medio con montones de poder y montones de datos.

Ahora, una cosa más ocurrió alrededor de ese momento. No ocurrió en las computadoras de Microsoft Windows aunque sí ocurrió en los servidores de Microsoft Windows y también en SOs más sensibles como Unix y BSD entre otros. Es decir, los servidores mantenían registros². Es una cosa muy buena de hacer. Las computadoras deben mantener registros. Fue una decisión muy sabia la de mantener registros, cuando se crea software para sistemas operativos. Ayuda en la inspección, hace realizables los rendimientos, hace posible estudiar las operaciones de las computadoras en el mundo real. Es una muy buena idea.

Pero si tenés un sistema que centraliza servidores y los servidores centralizan los registros, entonces estás creando vastos repositorios de datos jerárquicamente organizados sobre gente en los bordes de la red, sobre el que estos no tienen control y, a menos que tengan experiencia en la operación de servidores, no comprenderán su extensión, su significado, no entenderán su agregabilidad.

Entonces construimos una red de una arquitectura de comunicaciones diseñada para trabajar entre pares a la que definimos en un estilo cliente-servidor, al que entonces definimos como el desempoderado cliente en el borde y el servidor en el medio. Agregamos cada vez más procesamiento y almacenamiento en el medio y nos quedamos los registros -esto es, información acerca del flujo de información en la red- en lugares centralizados lejos de los seres humanos que controlaban o pensaban que controlaban, la operación de las computadoras que dominaban, cada vez más, sus vidas. Esto fue una receta para el desastre.

Esto fue una receta para el desastre. Ahora, todavía no mencioné la palabra “nube” que me fue puesta sobre la mesa cuando recibí la noticia de que estaría aquí esta noche hablando de privacidad y la nube.

No mencioné la palabra “nube” porque la palabra “nube” en realidad no significa nada. En otras palabras, el desastre que estamos teniendo no es la catástrofe de la nube. El desastre que estamos teniendo es la catástrofe de la forma en que malentendimos la red bajo la asistencia del software no-libre que nos ayudó a entenderla. Lo que “nube” significa es que los servidores dejaron de estar hechos de metal. “Nube” significa que ha ocurrido la virtualización de los servidores.

Así que, acá afuera en los polvorientos bordes de la galaxia donde vivimos en clientelismo desempoderado, nada cambió mucho. Mientras nos acercamos al centro

²NT: *logs*.

de la galaxia, todo se vuelve más borroso de lo que era antes. Ahora vemos halos donde hubo estrellas. Servidores con perillas y botones que se pueden apretar y así. En su lugar, lo que pasó es que el metal ya no representa a un único servidor. El metal es solamente un lugar donde el servidor podría estar. Entonces “nube” significa que los servidores han ganado libertad, libertad de moverse, libertad de danzar, libertad de combinarse y separarse y volverse a agregar y hacer toda clase de trucos. Los servidores ganaron libertad. Los clientes no ganaron nada. Bienvenidos a la nube.

Es una modificación menor de la receta para el desastre. Mejora la operabilidad de los sistemas que controlan clientes que estaban pensados para ser pares en una red hecha de cosas iguales.

Entonces esta es la arquitectura de la catástrofe. Si lo piensan, cada paso en esa revolución arquitectónica: de una red hecha de pares a servidores que sirven comunicaciones con humanos, a clientes que son programas corriendo sobre metal macizo, a clientes que son computadoras que la gente usa en un estado bastante desempoderado y servidores con una alta concentración de poder en la Red, a servidores como procesos virtuales corriendo en nubes de metal en el centro de una galaxia cada vez más caliente con clientes que están por ahí en los brazos de espiral polvorientos.

Todas esas decisiones arquitecturales fueron hechas sin ninguna discusión sobre sus consecuencias sociales a largo plazo, parte de nuestra dificultad general para hablar de las consecuencias sociales de la tecnología durante el gran período de invención de Internet hecho por científicos informáticos que no estaban terriblemente interesados en sociología, psicología social o, con algunas excepciones brillantes, la libertad. Entonces obtuvimos una arquitectura que estaba bastante sujeta al abuso. Estaba pidiendo que la abusen y ahora estamos teniendo el abuso que creamos. Porque alivianamos los clientes más y más y más. En efecto, los hicimos móviles. Los pusimos en nuestros bolsillos y empezamos a pasear con ellos.

Hay muchas razones para hacer clientes desempoderados y hay todavía más razones para desempoderar a la gente que posee los clientes y que curiosamente podría pensarse que eran quienes los controlaban. Si piensan por un momento cuánta gente está interesada en desempoderar a los clientes que son los teléfonos móviles verán a qué me refiero. Hay muchos potentadores de derechos, como piensan de sí mismos, yuxtapuestos, cada uno de los cuales tiene interés en desempoderar al cliente en el borde de la red para prevenir la movilidad de un hardware particular de un punto de la red a otro. Para prevenir hardware en particular tocar música que no fue comprada en el gran monopolio de música en el cielo. Para deshabilitar servicios de entrega de video que hagan competencia en nuevos chips que encontré que no corren

estándares de video populares, sean buenos o malos. Hay un montón de modelos de negocio que se basan en fastidiar el control sobre el hardware y software cliente del borde para privar al humano que curiosamente pensaba que lo había comprado para ocupar el lugar que el capitalismo dice que ocupan los dueños, esto es, el de tener control total sobre ellos.

De hecho, lo que tenemos, como dije un par de años atrás entre otras apariciones aquí en otra instalación de la NYU. De hecho, lo que tenemos son cosas que llamamos plataformas. La palabra “plataforma” como la palabra “nube” no tiene un significado inherente. Se tira mucho en lenguaje de negocios. Pero, básicamente, lo que significa plataforma es lugar que no podés dejar. Cosas a las que estás clavado. Cosas que no te dejan. Esas son plataformas. Y la Red, una vez que se convirtió en una zona arquitecturada jerárquicamente con servidores en el centro y clientes cada vez más desempoderados en el borde, se convierte en la zona de plataformas y fabricar plataformas se convierte en la orden del día.

Algunos años atrás un muy perspicaz abogado que trabaja en la industria me dijo “Microsoft nunca fue una compañía de software. Microsoft fue un compañía de gestión de plataformas”. Y pensé “Sí, directo al corazón”.

Entonces tenemos un montón de gestores de plataformas en una red organizada jerárquicamente y empezamos a desarrollar servicios. “Servicios” es una palabra complicada. No carece para nada de sentido pero es engañoso describirla. La usamos para un montón de cosas distintas. Necesitamos desesperadamente una taxonomía analítica de “servicios” como señaló mi amigo y colega Phillippe Aigrain en París hace 2 o 3 años. Las taxonomías de los “servicios” incluyen preguntas acerca de simplicidad, complejidad, escala y control.

Para dar un ejemplo, podríamos definir una dicotomía entre servicios complejos y simples en los cuales los servicios simples son cosas que cualquier computadora podría realizar y los servicios complejos son cosas que no se pueden hacer con una computadora. Hay que hacerlas en clústers o estructuras de cierta complejidad informática o administrativa. BUSCAR es un servicio complejo. En efecto, buscar es el servicio complejo arquetípico. Dada la naturaleza unidireccional de los vínculos en la Web y otros elementos en la arquitectura de datos en la que vivimos (esa es otra charla, en otro momento) buscar no es una cosa que podamos distribuir fácilmente. El poder en el mercado de nuestros amigos en Google depende enteramente del hecho de que la búsqueda no se distribuye fácilmente. Es un servicio complejo que debe ser organizado centralizadamente y entregado centralizadamente. Debe recorrer la Web unidireccionalmente, vínculo por vínculo, resolviendo el orden de todo para ayudarte a encontrarlo cuando lo necesites. Para hacer eso, por lo menos hasta ahora, no

desarrollamos buenas estructuras algorítmicas y de entrega en forma descentralizada. Entonces, buscar se convierte en el arquetipo de servicio complejo creándose a su alrededor un modelo de negocio para su monetización.

Hacer publicidad en el siglo XX era una actividad azarosa. Tirabas cosas y esperabas que funcionen. Hacer publicidad en el siglo XXI es una actividad exquisitamente precisa. Esperás a que un tipo quiera algo y le mandás publicidades de lo que quiere y ¡bingo!, funciona como magia. Por supuesto por debajo de un servicio complejo llamado búsqueda hay un servicio teóricamente simple llamada publicidad que, cuando se unifica con un servicio complejo, incrementa su eficiencia en órdenes de magnitud y el incremento de eficiencia del servicio simple al combinarse con el complejo produce una enorme flujo de plus ganancia que puede ser usado para fortalecer la búsqueda todavía más.

Pero esa es la parte inocente de la historia y no nos quedamos en la parte inocente de la historia por una variedad de temas. No voy a ser tedioso un viernes a la noche y decir que es porque la burguesía está envuelta en constante reinversión y mejora destructiva de sus propias actividades, y no voy a ser moralista un viernes a la noche, no podría hacer eso, y decir que es porque el pecado es inerradicable y los seres humanos son criaturas caídas y que la codicia es uno de los pecados que no podemos evitar cometer. Sólo voy a decir que como una especie de proceso social ordinario no paramos en lo inocente. Seguimos, lo cual es seguramente lo que deberían decir un viernes a la noche. Y entonces seguimos.

Ahora, donde llegamos es a descubrir que todo esto hubiera sido mejor si tuvieran los registros de todo porque una vez que tenés los registros de todo entonces cualquier servicio se convierte en una mina de oro esperando y la jodimos porque la arquitectura de la Red puso los registros en el lugar equivocado. Pusieron los registros donde la inocencia se tentaría. Pusieron los registros donde el estado fallido de los seres humanos implicaba malos problemas y los obtuvimos.

La nube significa que ya ni siquiera podemos apuntar en la dirección del servidor y porque no podemos apuntar en la dirección del servidor no tenemos ningún medio extra técnico o no técnico confiable para controlar este desastre en cámara lenta. Pueden hacer una regla sobre los registros o flujos de datos o preservación o control o acceso o publicación pero sus leyes son humanas y ocupan un territorio particular y el servidor está en la nube y eso significa que el servidor siempre está un paso adelante de cualquier regla que hagas o dos o tres o seis o ¡puf! Me acabo de dar cuenta que estoy sujeto a regulación, creo que me voy a Oceanía.

Lo que significa es que en efecto, perdimos la habilidad para usar regulación legal o cualquier otra cosa sobre la arquitectura física de la red para interferir en el proceso

de caída de la inocencia que era ahora inevitable en la etapa de la que hablo, que podríamos llamar Google tardío etapa 1.

Aquí es donde, por supuesto, entra el Sr. Zuckerberg.

La raza humana es susceptible al daño pero el Sr. Zuckerberg ha alcanzado un récord envidiable: ha hecho más daño a la raza humana que cualquier otro de su edad.

Porque ha aprovechado el viernes a la noche. Esto es, todos necesitan acostarse con alguien y él convirtió esto en una estructura para degenerar la integridad de la personalidad humana y hasta una extensión remarkable tuvo éxito con un trato bastante pobre. Es decir, “te voy a dar alojamiento web gratis y algunos pendorchos PHP y también obtenés espionaje gratis todo el tiempo”. Y funciona.

Esa es la parte triste, funciona.

¿Cómo pudo pasar esto?

No había una razón arquitectural, en verdad. No había una razón arquitectural. Facebook es la Web con “Me voy a quedar todos los registros, ¿cómo te sentís sobre eso?” Es un terrario de cómo se siente vivir en un panóptico construido de partes web.

Y no debería permitirse. Se reduce a eso. No debería permitirse. Es una manera muy pobre de entregar esos servicios. Están groseramente sobrevaluados para “espionaje todo el tiempo”. No son técnicamente innovadores. Dependen de una arquitectura sujeta al abuso y el modelo de negocio que lo sostiene es abuso. No hay ningún otro modelo de negocio para ellos. Esto es malo.

No estoy sugiriendo que deba ser ilegal. Tendría que ser obsoleto. Somos tecnólogos, deberíamos *arreglarlo*.

Estoy contento de estar con ustedes hasta ahora. Cuando llego a cómo debemos arreglarlo espero que todavía estén conmigo para que podamos hacerlo.

Pero digamos, por ahora, que ese es un muy buen ejemplo de cómo lo hicimos mal y lo que nos pasó por eso. Es más engañoso con Gmail por esa mágica cualidad de no haber sido tocado por manos humanas. Cuando le digo a mis estudiantes, “por qué dejan que otra gente lea sus correos?”, me responden “pero nadie está leyendo mi correo, ningún ser humano lo ha tocado. Eso me horrorizaría, me asustaría que los tipos de Google estén leyendo mi correo. Pero no está pasando así que no tengo problema.”

Ahora, no podemos decir esto de Facebook. En efecto, ellos saben demasiado sobre Facebook si se permitieran saberlo realmente. Ustedes leyeron y saben. Los trabajadores de Facebook saben quién va a tener un amorío antes que la gente porque pueden ver que X revisa obsesivamente la página en Facebook de Y. Hay algunas

investigaciones muy buenas hechas un par de años atrás por estudiantes del MIT que no voy a nombrar porque no respetaron los términos de servicio de Facebook durante su investigación. Sólo estaban revolviendo el contenido de algunas páginas, pero el propósito de esto era demostrar que podían encontrar homosexuales no declarados en Facebook.

No dicen nada sobre su orientación sexual. Sus amigos están declarados, sus intereses son los intereses de sus amigos declarados. Sus fotos están etiquetadas con sus amigos declarados y ellos están declarados sólo que no. Sólo están declarados en Facebook si alguien mira, que seguramente no es lo que tenían en mente y seguramente no es lo que teníamos en mente sobre ellos. En efecto, el grado potencial de desigualdad informacional y la alteración y dificultad que surgen de un malentendido, un error heurístico, en las mentes de los seres humanos acerca de qué es y qué no es descubrible acerca de ellos no es nuestro mayor problema con la privacidad.

Mis estudiantes, y sospecho que muchos de los estudiantes de los educadores de esta sala también, muestran constantemente esta dificultad en nuestro diálogo. Ellos piensan todavía en la privacidad como “el gran secreto que no quiero que se revele” y ese no es el problema. Su problema es todo el resto que forma el relleno, los datos que rellenan la vida, que no piensan de ninguna manera como un secreto pero que se agrega a lo que no quieren que nadie sepa. Lo que se agrega, en efecto, no sólo a lo que no quieren que otros sepan sino también a modelos predictivos sobre ellos de los que se horrorizarían si supieran que existen. La simplicidad con la que podés desanonimizar datos teóricamente anónimos, la facilidad con la que, de múltiples fuentes disponibles a través de terceras o cuartas partes, se puede ensamblar información, hacer mapas de la vida de la gente. La facilidad con la que empezás a juntar, con las pocas cosas que sabés de la gente, los datos que están disponibles para inferir inmensidad de cosas más.

Mi amigo y colega Bradley Kuhn que trabaja en el Centro Legal del Software Libre [SFLC] es uno de esos seres humanos arcaicos que cree que su número de seguridad social es una cosa privada. Y se toma grandes molestias para asegurarse que su número de seguridad social no sea revelada, lo cual es su derecho bajo nuestra ley, aunque suene raro. Pero, trató de obtener seguro médico o una caja de depósitos segura, o de hecho, operar en negocios siquiera. A veces hacemos malabares con nuestros asuntos porque el número de seguridad social de Bradley es un secreto. Un día le dije “sabés, ya está, Google ya conoce tu número de seguridad social”. Él dijo, “no, no lo saben, nunca se lo dije a nadie”. Yo dije, “si pero conocen los números de

seguridad social de todos los demás que nacieron en Baltimore ese año. El tuyo es el que falta”.

Y como ustedes saben, es verdad. Los datos que inferimos son los datos en los agujeros entre los datos que ya conocemos, si sabemos lo suficiente.

Así que el lugar donde vivimos se ha convertido en un lugar del que sería muy poco inteligente decir que no hay nada que no se sepa. Si sos bastante bien conocido en la Red, y cada uno de nosotros por una razón u otra es bastante bien conocido en la Red. Queremos vivir ahí. Es nuestro vecindario. Lo que no queremos es vivir con una cámara en cada árbol y un micrófono en cada arbusto y un minero de datos bajo nuestros pies mientras caminamos y la Red es así ahora. Este no es un juicio estético del '95 sobre cómo el vecindario se llenó de gente que no comparte nuestro etnocentrismo tecnogeek. No estoy lamentando el progreso de alguna forma democratizante. Al contrario, lamento el progreso hostil a la libertad humana. Todos sabemos lo que es hostil a la libertad humana. Todos comprendemos las posibilidades despóticas gracias a las distopías en que es fértil la ciencia ficción que leímos cuando éramos chicos. La Guerra Fría fue fértil en la invención fantástica de cómo vivimos ahora y es difícil aceptarlo aunque sea verdad. Afortunadamente, por supuesto, no es propiedad del gobierno. Bueno, lo es. Es afortunado. Es verdad. Es afortunado que sea propiedad de gente a la que podemos sobornar no importa quién seas. Si sos el gobierno hay formas muy fáciles de hacerlo. Llenás un formulario de citación [subpoena] y lo mandás por correo.

Pasé dos horas ayer en una clase de leyes explicando en detalle por qué la Cuarta Enmienda ya no existe porque fue un jueves por la noche y ¿quién haría eso un viernes por la noche? Pero la Cuarta Enmienda no existe más. Voy a poner la grabación en la Red y el FBI y ustedes pueden escucharla cuando quieran.

Tenemos que avivarnos si somos los que nos preocupamos por la libertad, el juego empezó hace tiempo y estamos atrasados. Hicimos muchas cosas buenas y tenemos un montón de herramientas dando vueltas que construimos durante los últimos 25 años. Yo mismo ayudé a la gente a construir las. Ayudé a la gente a mantenerlas a salvo, ayudé a la gente a prevenir que el monopolio ponga todas esas herramientas en su bolsa y se las lleve y estoy contento de que esas herramientas estén por ahí pero tenemos que admitir que no las hemos usado para proteger la libertad porque la libertad está decayendo y eso es lo que quiso decir David en su amable introducción.

De hecho, la gente que está invirtiendo en las nuevas empresas de la no-libertad es también la que escuchás si pasas tiempo en Silicon Valley en estos días diciendo que el código abierto se ha vuelto irrelevante. ¿Cuál es su lógica? Su lógica es que el

software como un servicio se está convirtiendo en la forma del mundo. Ya que nadie va a obtener software nunca más, las licencias que dicen “si le das software a la gente también tenés que darles libertad” ya no importan porque no le estás dando software a nadie. Sólo les estás dando servicios.

Bueno, está bien. El código abierto ya no importa. El software libre importa mucho porque, por supuesto, el software libre es código abierto con libertad. Stallman tenía razón. La libertad es lo que importa. El resto es sólo código fuente. La libertad todavía importa y lo que tenemos que hacer es hacer que el software libre importe en el problema en el que servicios no libres son proveídos en formas no libres que empiezan a deteriorar la estructura de la libertad humana.

Como un montón de no-libertades, el verdadero proceso social inmanente que fuerza esta no-libertad no es más que la conveniencia percibida.

Todas las formas de libertad sobrepasan la conveniencia percibida. Ustedes lo saben. Han dejado de pagar las cosas en efectivo. Usan una tarjeta que pueden agitar frente a un lector RFID.

La conveniencia dicta que necesitás alojamiento web gratuito y chirimbolos PHP como recompensa a dejarte espiar todo el tiempo porque los servidores web son terriblemente difíciles de correr. ¿Quién podría tener un servidor web propio y mantener los registros? Sería brutal. Bueno, lo sería si fuera IIS³. Era una autorrealización, estaba pensado para que así fuera. Estaba diseñado para decir “vos sos un cliente, yo soy un servidor, yo inventé Windows 7. Me voy a quedar los registros, muchas gracias.” Así era la industria. Nosotros construimos otra industria. Está aquí. Pero no está [en uso]. Bueno, sí, está aquí de alguna manera. Así qué ¿dónde no está? Bueno, no está en el servidor web personal que no tengo y que prevendría que cayera... bueno, ¿por qué no hacemos algo al respecto?

¿Qué necesitamos? Necesitamos un buen servidor web que puedas poner en tu bolsillo y enchufar en cualquier lado. En otras palabras, no debería ser más grande que un cargador de celular y deberías poder enchufarlo en cualquier tomacorriente del mundo y en cualquier cable cercano o sincronizarlo con cualquier router WiFi que esté en el vecindario. Tendría que tener un par de puertos USB para enchufarle cosas. Debería saber cómo levantarse solo. Debería saber cómo iniciar el servidor web, cómo recolectar todas tus cosas de las redes sociales en que las tengas. Debería saber cómo enviar una copia de seguridad cifrada de todo a los servidores de tus amigos. Debería saber cómo microbloguear. Debería saber cómo hacer un ruido que suene como tweet pero no infrinja la marca de nadie. En otras palabras, debería saber cómo ser vos... ah

³*Internet Information Server*, servidor web que viene con Microsoft Windows

perdónenme porque necesito usar una palabra peligrosa -tu avatar- en una red libre que trabaja para vos y que mantiene los registros. Siempre podés saber qué pasa en tu servidor y si alguien quiere saberlo tiene que conseguirse una orden de registro.

Y si querés mudar tu servidor a Oceanía o Sealand o Nueva Zelanda o el Polo Norte, te comprás un pasaje de avión y lo ponés en tu bolsillo. Lo llevás ahí. Lo dejás. Ahora hay un poco más que necesitamos hacer. Es trivial. Necesitamos DNS dinámico y todas cosas que ya inventamos. Está todo ahí, nadie necesita algo especial. ¿Tenemos el servidor que podés poner en tu bolsillo? En efecto, lo tenemos. De la góndola de hardware. Hermosos y pequeños aparatos de pared con chips ARM. Exactamente lo que especifiqué recién. Los enchufás, los conectás. ¿Cuál es el software que tiene? Bueno, es cualquier software que le quieras poner.

De hecho, te lo mandan con la distro⁴ más popular adentro, sólo tenés que decir la que querés. ¿Cuál querés? Bueno podés querer Debian GNU/Linux para Redes Sociales y va a ser entregado libremente, libremente como en libertad quiero decir. Lo que hace que todas las cosas que nombro -levantarse, correr su pequeño Apache o lighttpd o su diminuto httpd, hacer todas las cosas que necesito que haga -sincronizarse, obtener tus datos de redes sociales, absorberlos, buscar tus copias de seguridad, encontrar a tus amigos, registrar tu DNS dinámico. Todo es trivial. Todas son cosas que tenemos. Necesitamos ponerlas juntas. No estoy hablando de algo que es difícil para nosotros. Tenemos que armar un dispositivo de distribución de software libre. ¿Cuántos necesitamos?

Necesitamos darles un montón a todos nuestros amigos y necesitamos decirles, tomen jueguen con esto y mejórenlo. Necesitamos hacer lo que realmente somos buenos haciendo porque todo lo demás ya está hecho, en la bolsa, listo y barato. Esos servidores de pared están US\$99 ahora y están bajando a US\$79 y cuando haya cinco millones de ellos costarán US\$29.99.

Entonces vamos a la gente y les decimos US\$29.99 por única vez, muy buena red social, se actualiza automáticamente, el software es tan fuerte que no lo podrías voltear aunque lo patearas, usado en miles de millones de servidores en todo el planeta haciendo un trabajo maravilloso. ¿Y sabés qué? Obtenés no-espionaje gratis. ¿Quiéren saber qué es lo que tenés ahí? Que se consigan una orden de registro de tu casa, tu castillo, el lugar donde la Cuarta Enmienda todavía existe de alguna manera cada martes o jueves cuando la Corte Suprema no entra en sesión. Podemos hacerlo. Podemos hacerlo. Eso requiere que hagamos sólo lo que somos realmente buenos haciendo. El resto lo conseguimos libremente. ¿El Sr. Zuckerberg? No tanto.

⁴Distribución de GNU/Linux.

Porque por supuesto, cuando hay un competidor a “todo el espionaje permanente aunque no te guste”, la competencia va a ir realmente bien. No esperen que Google sea el competidor. Esa es nuestra plataforma. Lo que necesitamos es hacer algo tan grasoso que no haya otra plataforma de red social nunca más. ¿Podemos hacerlo? Si, absolutamente. De hecho, si no tienen cita el viernes a la noche, armemos un hackfest y hagámoslo. Está bien a nuestro alcance.

¿Vamos a hacerlo antes que el IPO de Facebook? ¿O vamos a esperar hasta después? ¿Realmente? ¿Honestamente? Seriamente. El problema que tiene la ley muy a menudo en el mundo en que vivimos y practicamos y trabajamos, el problema que la ley tiene muy a menudo, es el problema que la tecnología puede solucionar. Y el problema que la tecnología puede solucionar es el lugar al que vamos a la ley. Ese es el movimiento del software libre. Está el hacking de software por acá y está el hacking legal por allá y cuando juntás ambos el todo es superior a la suma de las partes. Entonces, no es que tenemos que vivir en la catástrofe. No es que empezemos a revertir la catástrofe es difícil para nosotros. Necesitamos rearquitecturar los servicios en la Red. Necesitamos redistribuir los servicios de vuelta hacia los bordes. Necesitamos devirtualizar los servidores donde nuestra vida está almacenada y necesitamos recuperar alguna autonomía para vos como el dueño del servidor.

Las medidas para tomar estos pasos son técnicas. Como es usual, los fabricantes de los aparatos se nos adelantaron. El hardware no nos limita. Como es usual, en estos días, el software no es una limitación porque hicimos mucho software maravilloso que de hecho es usado por todos los tipos que están del lado de la mala arquitectura. No quieren pasársela sin nuestro software. La mala arquitectura está habilitada, sostenida por nosotros. La rearquitectura también. Y tenemos nuestro beneficio mágico habitual. Si tenemos una copia de lo que estamos hablando, tenemos todas las copias que necesitamos. No tenemos limitaciones de manufactura o transporte o logística. Si terminamos el trabajo, ya está. Escalamos.

Este es un desafío técnico por una razón social. Es una frontera por explorar de la gente técnica. Hay un enorme retorno social por hacerlo.

El retorno es claro porque el daño a reparar existe y gente que conocen está sufriendolo. Todo lo que sabemos acerca de por qué hacer software libre dice que este es el momento en que estamos en nuestro ambiente. Es un desafío técnico alcanzable incrementalmente por extensión de dónde estamos parados ahora y de quiénes nos preocupamos mejor. Nunca en 25 años de hacer este trabajo, nunca nos he visto fallar al enfrentar un desafío que puede definirse en estos términos. Así que tampoco pienso que vayamos a fallar ahora.

El Sr. Zuckerberg merece la bancarrota por lejos.

Vamos a dársela. Gratis.

Y yo prometo, y ustedes deberían prometer también, no espiar en el proceso de bancarrota. No es nuestro asunto. Es privado.

Esta es una historia con un final feliz en potencia. Es una historia potencialmente feliz y si lo hacemos entonces habremos acallado un rumor más acerca de nuestra irrelevancia y todo el mundo en el Valle va a tener que encontrar otra palabra de moda y todos los tipos que piensan que Sandhill Road va a surgir en nuevo poder y gloria espiando a todos y monetizando el espionaje va a tener que encontrarse otra línea de trabajo, y todo esto está puramente del lado de los ángeles. Puramente del lado de los ángeles.

No nos vamos a sacar de encima todos nuestros problemas, de ninguna manera, pero solamente el mover los registros de ellos a ustedes es el mayor paso que podemos tomar hacia resolver un rango entero de problemas sociales que siento muchísimo sobre lo que queda de mi Constitución estadounidense y que sentiré muchísimo si estuviera viendo fallar la ley de protección de datos europea desde adentro en lugar de desde afuera y que me sentiré un poco esperanzado si fuera, no sé, un amigo mío en China. Porque ustedes saben que por supuesto tenemos que poner una VPN adentro de ese aparato.

Y probablemente tengamos que ponerle un enrutador Tor.

Y por supuesto, tenemos Bittorrent, y al momento de terminar todo esto, tenemos el aparato de la libertad. Tenemos un aparato que no solamente nos saca del pozo en el que estamos, tenemos un aparato que pone una escalera para la gente que está más profundo que nosotros en ese pozo, que es otra cosa que nos encanta hacer.

Lo que creo es que el Departamento de Estado de EEUU va a estar puteando [slanging] al Partido Comunista Chino por un año o dos acerca de la libertad en Internet y creo que el Partido Comunista Chino va a putear también y lo que van a decir es “¿Ustedes piensan que tienen una buena privacidad y autonomía en la Internet en su vecindario?” Y cada vez que hacen eso ahora como si lo hubieran estado haciendo desde hace 2 semanas solamente diría ¡ay! si fuera Hillary Clinton y supiera algo del tema porque no la tenemos. No la tenemos. Es verdad. Tenemos un tipo capitalista y ellos tienen uno de tipo vanguardia centralista del partido o tal vez algo así como marxista o tal vez marxista o tal vez sólo totalitaria pero no vamos a ganar la discusión sobre la libertad en la Red con Facebook sobre nuestras espaldas. No vamos a ganarla.

Pero empiecen a instalar esos servidores de pared en la sociedad norteamericana y empiecen a retomar los registros y ¿quieres saber con quién habló el viernes a la noche? Conseguite una orden de registro y dejá de leer mi correo. A propósito ahí

está mi llave GPG y ahora estamos realmente cifrando y así y así y así y empieza a parecerse a algo que querríamos convertir en una cruzada nacional. Realmente estamos creando libertad para otra gente. Para gente que vive en lugares en los que la web no funciona.

Así que no es un desafío que no queremos enfrentar. Es uno que queremos enfrentar con ganas. De hecho, estamos en un estado feliz donde todos los beneficios que podemos conseguir sobrepasan por mucho la complejidad técnica de hacer lo que tenemos que hacer, que no es mucho.

De ahí es de donde venimos. Venimos de donde nuestra tecnología era más libre de lo que pensamos y de donde regalamos un montón de libertad antes de que supiéramos que en realidad no existía más. Venimos de donde el software no libre tuvo malas consecuencias sociales más allá de lo que incluso los agitadores libertarios sabían. Venimos de metáforas no libres que tienden a producir mala tecnología.

En otras palabras, venimos de cosas a las que nuestro movimiento estaba diseñada para enfrentar desde el principio pero venimos de ahí. Y todavía estamos viviendo con las consecuencias de no haberlo hecho lo bastante bien la primera vez, aunque lo alcanzamos gracias a Richard Stallman y seguimos.

Donde vivimos ahora no es el lugar donde vamos a ver vivir a nuestros nietos. Donde vivimos ahora no es el lugar donde queremos hacer tours guiados. Antes les decía a mis estudiantes ¿cuántas cámaras hay entre el lugar donde viven y la Escuela de Leyes? Cuéntenlas. Ahora le digo a mis estudiantes ¿cuántas cámaras hay entre la puerta del colegio y este aula? Cuéntenlas.

Ahora les digo a mis estudiantes “¿pueden encontrar un lugar donde no haya cámaras?” Ahora, lo que pasó en ese proceso fue que creamos inmensos auxiliares cognitivos para el Estado -enormes máquinas para escuchar. Ustedes saben cómo es si viven en una universidad estadounidense gracias a las compañías de música y películas que te recuerdan todo el tiempo que vivís en el medio de una red enorme de vigilancia. Estamos rodeados por gente escuchándonos y mirándonos. Estamos rodeados de datos explotables.

No todo eso va a desvanecerse porque tomemos Facebook y lo dividamos y nos llevemos nuestros pequeños fragmentos de él. No va a desvanecerse porque ya no aceptemos alojamiento web gratis con espionaje adentro. Tenemos otro trabajo que hacer. Algo de ese trabajo es trabajo de abogados. Admito eso. Algo de este trabajo es escribir leyes y litigar y crear problemas y otras cosas de abogados. Está bien. Estoy listo.

Mis amigos y yo vamos a hacer la parte de los abogados. Sería mucho más simple hacer el trabajo de abogado si viviéramos en una sociedad que entendiera

mejor qué es la privacidad. Sería mucho más simple hacer el trabajo de abogado si la gente joven se diera cuenta cuando crezca y empiece a votar o ya esté votando ahora que es grande, que esto es un problema. Que necesitan terminar de hacer lo que queda de la misma manera que arreglábamos las grandes cosas cuando éramos chicos. Vamos a tenerlo mucho más fácil con la enorme confusión de solapamiento internacional de regímenes cuando hayamos deteriorado la inmensa fuerza del capitalismo norteamericano que nos fuerza a ser menos libres y más vigilados para provecho de otra gente todo el tiempo. No es que esto va a resolver todos los problemas pero el trabajo fácil es muy rico y gratificante ahora mismo.

Estos problemas son muy malos. Terminar con los más fáciles va a mejorar la política para resolver los difíciles y está justo en nuestro camino. La solución está hecha de nuestras partes. Tenemos que hacerlo. Ese es mi mensaje. Es viernes a la noche. Alguna gente no quiere irse derecho a programar, estoy seguro. Podemos atrasarlo hasta el martes pero ¿cuánto más quieren esperar? Saben que cada día que pasa se van más datos que nunca van a volver. Cada día que pasa hay más inferencias sobre los datos que no podemos deshacer. Cada día que pasa apilamos más cosas en las manos de la gente que tiene demasiado. Así que no es que debamos decir “uno de estos días me voy a poner a hacerlo”. No es que debamos decir “Creo que en cambio voy a pasar mi tiempo navegando noticias sobre el iPad”.

Es mucho más urgente que eso.

Es que no nos hemos dado a nosotros mismos la dirección a la que ir así que demosnos la dirección a la que ir. La dirección a la que ir es hacia la libertad usando software libre para hacer justicia social.

Pero ustedes saben esto. Ese es el problema de hablar un viernes por la noche. Hablás por una hora y todo lo que le decís a la gente es algo que ya saben.

Así que muchas gracias. Ahora acepto preguntas.

Johan Söderberg

2

¡Hackers GNUíños!¹

2.1. La izquierda política y la política de los hackers

En este artículo voy a mirar el hacking desde la perspectiva del sindicalismo. El significado político del hacking informático ha intrigado a la vieja izquierda, aunque existen algunos vasos comunicantes entre el movimiento hacker y movimientos sociales más tradicionales. Los más notables de estos grupos dentro del underground informático son los autodenominados 'hacktivistas'. Tratan de aplicar sus habilidades informáticas a una agenda política ya establecida, como el feminismo o el ecologismo[2]. Pero es más desafiante darle sentido a la agenda política del núcleo del movimiento hacker. De inmediato, uno se pregunta si el underground informático puede siquiera considerarse un movimiento social. Muchos hackers, tal vez la mayoría, dirían que este no es el caso. A lo más, la política es tenida como un punto secundario frente a la alegría de jugar con tecnología informática[3]. Aún así, de esta apasionada afirmación de las computadoras han crecido ideas con

¹Este artículo es la traducción al castellano de "Hackers GNUited!", publicado en Free Beer 1.0 (2008). El libro completo en inglés puede encontrarse en <http://freebeer.fscons.org>. Traducción por Nicolás Reynolds <fauno@kiwwwi.com.ar>, revisión y correcciones por Leonardo G. De Luca, agosto, 2010. Este artículo y el original están licenciados bajo Creative Commons Atribución-CompartirIgual 2.5 <http://creativecommons.org/licenses/by-sa/2.5/>.

ramificaciones políticas. Por caso, hackers que de otra manera no se considerarían ‘políticos’ tienden a oponerse a las patentes del software y la vigilancia estatal de Internet, para mencionar sólo dos ejemplos. De hecho, estos puntos de vista están tan compartidos en el underground informático que parecen más sentido común que consignas políticas. Algunos temas, como las campañas contra la expansión de las leyes de propiedad intelectual y la defensa de la libertad de expresión, han sido agregadas a agendas políticas y son promovidas activamente por grupos de lobby hacker, dos ejemplos de esto son la Fundación del Software Libre (FSF) y la Fundación Frontera Electrónica (EFF). Estas organizaciones están claramente involucradas en política, aunque sostienen que estos intereses cortan por ejes diferentes de los de la división derecha-izquierda tradicional. Cuando los científicos sociales han analizado los supuestos que descansan detrás de las declaraciones públicas de estos grupos de lobby hacker han encontrado sin embargo una cercana afinidad con el liberalismo[4].

Un par de escritores de izquierda han hecho un corte en el que no interpretan el hacking como una ideología liberal. Muy al contrario, creen que el movimiento hacker podría revitalizar las viejas luchas de la izquierda, no sólo por la libertad individual sino también en contra de la injusticia y la desigualdad. El más renombrado enterado que ha expresado tales opiniones sobre el hacking es Eben Moglen. Es profesor de leyes y fue por un largo tiempo una figura maestra en la Fundación del Software Libre (FSF). Moglen es además el autor del *Manifiesto puntoComunista*, donde predijo que el anarquismo del desarrollo de software reemplazará a las firmas capitalistas como el modo más eficiente de organizar la producción en el futuro[5]. El estudioso de los medios Richard Barbrook razonaba en forma similar cuando desbancaba el hype acerca de los ‘mercados libres en el ciberespacio’ creado en los ‘90. En su lugar presentó su propia visión de una economía anarquista del don de alta tecnología.² El impulso a dar se genera automáticamente del hecho de que en Internet la gente tiene un interés propio en compartir información libremente antes que comercializarla en un mercado[6]. Podría decirse que el surgimiento de Napster y generaciones posteriores de tecnologías de compartición de archivos probaría que Barbrook estaba en lo cierto. Aun más iconoclasta en su adopción de la retórica socialista es el filósofo esloveno Slavoj Zizek. Parafraseó el apoyo de Lenin a la electricidad diciendo, en tono de burla, que ‘el socialismo es igual a libre acceso a Internet más todo el poder a los Soviets’[7]. Por lo menos unos pocos comunistas de la vieja escuela están tomando esta idea con seriedad. Creen que la tecnología informática ha provisto el vínculo

²NdT: traduzco “gift economy” por economía del don ya que es el término más extendido en castellano, y podría decirse más exhaustivo que “regalo”. Ver Ensayo sobre el don, de M. Mauss, y El enigma del don, de M. Godelier

faltante que al fin podría hacer de una economía planificada una alternativa viable a la economía de mercado[8].

Pero estas afirmaciones positivas del hacking y la tecnología informática probablemente sean opiniones minoritarias dentro de la izquierda tradicional. Hay una sospecha profundamente arraigada entre los intelectuales de izquierda hacia la tecnología informática y, por extensión, a sus más celosos usuarios, es decir los hackers. El origen de Internet dentro de instituciones del EEUU de la Guerra Fría es suficiente para desanimar a muchos pensadores progresistas[9, 10]. Hay que agregar a esto el hype alrededor de Internet a mediados de los '90. Dió rienda suelta al viejo dilema de la 'Edad de la Información'. Esta noción data de los '50 y proviene de sociólogos conservadores estadounidenses que se propusieron desaprobar la relevancia y continuidad de los conflictos de clase. Al anunciar el fin de la sociedad industrial, buscaban probar que las tensiones entre clases se habían disuelto y la lucha ideológica entre liberalismo y socialismo se estaba volviendo obsoleta. En consecuencia, los académicos con tendencias de izquierda protestaron contra las nociones acerca del surgimiento de la Edad de la Información e insistieron en la existencia y continuidad del industrialismo, capitalismo y conflicto de clases[11]. Para probar este punto les basta con llamar atención sobre las condiciones inhumanas bajo las que la electrónica informática es manufacturada en zonas de exportación de países del tercer mundo[12]. Un informe de 2008 ha documentado cómo en China chicas de 16 años de edad trabajan de doce a quince horas diarias, seis o siete días a la semana, y apenas ganan para vivir[13]. Estos descubrimientos resuenan junto a la circunstancia histórica de que las tarjetas perforadas, maquinaria de control numérico, mainframes y otros embriones de computadoras modernas fueron instrumental para volver redundantes a los obreros y degradar sus calificaciones laborales en el momento de la producción[14, 15].

Ahora, habiendo delineado brevemente la relación confusa entre la izquierda tradicional y el empuje político de los hackers, este artículo procederá a examinar la importancia política de estos últimos a la luz de un viejo debate acerca de la maquinaria fabril y el trabajo asalariado. El Debate Braverman, como es conocido según el autor que inició la controversia, se retrotrae a los '70. Harry Braverman publicó un libro en el que argumentaba que la descalificación del trabajo era una cualidad inherente al capitalismo. La razón era que los gerentes tratan de independizarse de los obreros altamente calificados para mantener bajos salarios y debilitar políticamente a los sindicatos. Braverman encontró apoyo a su hipótesis en los escritos de los pioneros de la filosofía gerencial. La figura pivote entre ellos, Winston Taylor, fue el fundamento de lo que se conocería como 'gerenciamiento

científico' o 'taylorismo'. Una idea central del gerenciamiento científico es que la línea de producción debe reestructurarse de manera que las tareas puedan realizarse mediante rutinas simples que requieran un mínimo de calificación de los empleados. Taylor argumentó que esto podía lograrse con la introducción de maquinaria fabril. Braverman mostró cómo esta estrategia se estaba desplegando en la industria pesada a mediados del siglo XX.

Este punto de vista puede servir como lente con la que mirar el significado político de la maquinaria informática y el hacking de esta. La novedad de este argumento es que el análisis de los hackers está formulado desde una perspectiva orientada a la producción, en oposición a la perspectiva de los derechos del consumidor. Se argumentará que el surgimiento del Software Libre y Abierto (FOSS) puede trazarse hasta el conflicto industrial entre gerentes y trabajadores. Además, la similitud entre la lucha de los trabajadores contra la maquinaria fabril y la lucha del movimiento hacker contra el software propietario va a ser puesta en relieve. El libre acceso al código fuente, preocupación clave de los hackers, contradice el sistema fabril y la lógica del gerenciamiento científico en la programación informática[16]. Aunque la situación de los programadores comparada con la de los obreros es muy diferente en muchos aspectos, el artículo muestra que ambos grupos están preocupados en la meta de preservar sus calificaciones y la autonomía obrera frente al rápido cambio tecnológico. La demanda de los hackers de que el código fuente debe ser libremente accesible puede interpretarse como parte de una estrategia orientada a preservar el saber-cómo de los programadores y su control sobre las herramientas de su oficio.

2.2. La máquina en el trabajo

Los sentimientos ambivalentes de entusiasmo y miedo que a menudo evoca la tecnología informática entre la gente tienen un precedente histórico. En el amanecer de la revolución industrial, se debatió fuertemente en todos los segmentos de la sociedad qué efecto tendría la mecanización sobre el ser humano, tanto social como espiritualmente[17]. Incluso algunos de los adelantados de la teoría económica liberal, como David Ricardo, admitieron que la clase trabajadora tenía buenas razones para resentir de la maquinaria fabril[18]. La miseria que cayó sobre los trabajadores que estaban subyugados a la maquinaria y la disciplina fabril fue vívidamente descrita por James Kay, un reformista social que trabajó como doctor en las villas:

“Mientras la máquina anda la gente debe trabajar –hombres, mujeres y niños están uncidos juntos con hierro y vapor. La máquina animal

–frágil en el mejor de los casos, sujeta a mil fuentes de sufrimiento– está encadenada a la máquina de hierro, que no conoce sufrimiento ni fatiga.”[19]

Escritores sobre el gerenciamiento tempranos como Andrew Ure y Charles Babbage dieron la bienvenida a esta oportunidad y aconsejaron a los dueños de las fábricas sobre cómo diseñar maquinaria para mantener a los obreros dóciles e industriuosos [20, 21]. Sus testimonios informaron el análisis del capitalismo de Karl Marx, quien denunció la maquinaria fabril como el ‘modo material de existencia del capital’. Pero él también cualificó su crítica contra la tecnología agregando que: “Tomó tiempo y experiencia para que los trabajadores aprendieran a distinguir la maquinaria de su empleo por el capital, y en consecuencia a transferir sus ataques de los instrumentos materiales de producción a la formación social que utiliza esos instrumentos.”[22]. Así Marx renunció a la estrategia de rompimiento de máquinas que fue el sello de los Luditas. Los Luditas consistían en peñadores, tejedores y artesanos que sintieron que su oficio era amenazado por la introducción de nuevos telares y una subsecuente reorganización de la industria textil. Se hicieron ataques nocturnos para destruir molinos de lana y marcos de tejido de los que los ‘maestros tejedores’ eran dueños. Estas actividades alcanzaron su punto cúlmine en 1811-1813 y una vez la corona inglesa tuvo que desplegar 14.400 soldados en la región para poder aplastar las insurgencias nocturnas. Remarcablemente, fueron movilizadoss más soldados ingleses contra los Luditas de los que habían sido enviados a Portugal cuatro años antes para enfrentar al ejército de Napoleón[23]. En su clásico reexamen del levantamiento Ludita, Eric Hobsbawm mostró que el rompimiento de máquinas no fue una resistencia fútil contra la tecnología y el progreso, como se hizo ver después. En su lugar la interpretó como un método de ‘negociación colectiva mediante el disturbio’. Romper la maquinaria era una opción, pero los obreros también pudieron presionar a sus empleadores prendiendo fuego los lugares de trabajo o enviando amenazas anónimas. Hobsbawm concluye que, a juzgar por la habilidad de los trabajadores de preservar sus salarios y condiciones laborales, tuvieron un éxito moderado[24].

La lectura equivocada de la rebelión Ludita como alocada, irresponsable y, más importante, sin relación alguna con la política, se asemeja al retrato de los hackers que se hace en los medios de noticias actuales. Andrew Ross protestó contra la imagen del hacker como un criminal menor, un bromista juvenil o, alternativamente, un yuppie de la Era de la Información. Hace hincapié en que el sabotaje espontáneo hecho por empleados contribuye la mayor parte de tiempo muerto en las oficinas. Estos ataques a menudo no son reportados ya que los gerentes prefieren culpar a adversarios

externos. Con esta observación en mente, sugiere una definición de hacking más amplia:

“Mientras un pequeño número de usuarios de computadoras se categorizaría a sí mismo como ‘hacker’, existen razones de peso para extender la definición restringida de *hacking* por debajo y a través de la jerarquía de analistas de sistemas, diseñadores, programadores y operadores para incluir a todos los trabajadores de alta tecnología –sin importar cuán inexpertos– que puedan interrumpir, molestar y redirigir el suave flujo de comunicaciones estructuradas que dicta su posición en las redes sociales de intercambio y determina el paso de su agenda de trabajo.”[25]

La sospecha de Andrew Ross es confirmada por estudios conducidos por organizaciones de empleadores. Que el personal provoque la falla del equipamiento informático de sus empleadores es el escenario más común, más costoso y más temido por las firmas que la intrusión de usuarios informáticos externos. De acuerdo a una encuesta realizada en 1998 por la Iniciativa por la Seguridad Informática en conjunto con el FBI, el costo medio de un ataque informático exitoso en los EEUU por un usuario externo es de US\$56.000. En comparación, el costo medio de actos maliciosos hechos por usuarios internos (es decir, empleados) se estimó en US\$2,7 millones[26]. La afición de los empleados por atacar los sistemas informáticos de sus empleadores subraya el rol de la computarización en la transformación de las condiciones de trabajo de los trabajadores de cuello blanco. La comparación de Ross con el sabotaje seguramente despertará algunas objeciones entre los hackers ‘reales’. Aquellos en el movimiento hacker que quieren adecuarse a la definición tratan de contrarrestar el estereotipo negativo de los hackers que hacen los medios al diferenciar entre hackers originales y los así llamados crackers. El primer nombre está reservado a usos creativos que contribuyen a proyectos de software socialmente útiles. Las connotaciones negativas del crimen informático están reservadas para el último grupo³.

Estos esfuerzos por mejorar las relaciones públicas de los hackers meramente subrayan el paralelo histórico con la militancia laboral sugerido arriba. El movimiento sindicalista también ha rescrito su propia historia para que el sabotaje, las huelgas ilegales y los actos de violencia queden fuera de la foto. En efecto, los sindicatos

³Por ejemplo el Jargon File, que es considerada la fuente autorizada de argot hacker, hace malabares para distinguir entre crackers y hackers ‘reales’: <http://url1.ca/f6o3> (última visita: 27-05-2009)

han tenido bastante éxito en formalizar el conflicto entre trabajo y capital como un problema de negociación institucionalizada. No obstante, puede decirse que la negociación colectiva del trabajo todavía descansa sobre la amenaza velada del sabotaje, las huelgas y los disturbios[27]. De la misma manera, entiendo la distinción entre hackers y crackers como una construcción discursiva que no retrata certeramente las raíces históricas ni el solapamiento actual de esta subcultura. En cambio, busca redefinir el significado de hacking y dirigirlo en una dirección particular. A pesar del éxito de esta retórica, la liberación de warez, la rotura de cifrados, y el crackeo de servidores corporativos juegan sin embargo un papel en la lucha más amplia por mantener la información libre.

Habiendo dicho esto, el lector estaría en lo correcto al objetar que la motivación de los Luditas y trabajadores para rechazar la maquinaria fabril y de oficina es muy diferente de la motivación de los hackers que luchan contra el software propietario. Para el último grupo, las computadoras se revelan como bienes de consumo y fuentes de estímulo. Podría decirse que su relación con la tecnología es de pasión más que de hostilidad. Aún cuando los hackers (crackers) sabotean servidores corporativos, este es un acto de alegría. Los trabajadores de oficina descontentos podrían obtener placer al destruir la computadora de su empleador, pero todavía es significativo decir que su acto nace del resentimiento contra su situación. Esta diferencia en motivación, sin embargo, no descarta la posibilidad de que los hackers compartan terreno con los antiguos rompedores de máquinas. Ambos están atrapados en un combate luchado en el terreno del desarrollo tecnológico. Incluso podría ser que la apasionada afirmación de la tecnología hecha por los hackers ofrezca una línea de ataque aún más subversiva, en comparación a, por ejemplo, la insurgencia Ludita. Aunque es incorrecto decir que los Luditas estaban en contra de la tecnología *per se*, es verdad que defendían una tecnología desactualizada contra la nueva del sistema fabril. Así es que parece que su causa estaba perdida antes de empezar. Los hackers, en contraste, tienen una tecnología propia sobre la que trabajar. Pueden declarar plausiblemente que su modo de escribir código es más avanzado que el 'modelo fabril' de desarrollo de software propietario.

2.3. Descalificación de obreros, recalificación de usuarios

Es una dialéctica extraña la que llevó a la situación actual en la que los hackers recuperan tecnología informática de las compañías e instituciones gubernamentales. Las pistas de cómo se llegó a esta situación pueden encontrarse en lo que se llamó el Debate Braverman. La controversia tuvo lugar contra el telón de la idea acerca de la llegada de una era post-industrial[28]. Dos décadas después, la misma idea fue reempaquetada como 'el surgimiento de la Era de la Información' o la 'Sociedad en Red'. Esta noción ha tenido muchos tonos pero invariablemente pinta un futuro brillante donde el capitalismo avanzará más allá del conflicto de clases y el trabajo monótono. Crucialmente, esta transición no fue traída a través de la lucha social sino que se debe exclusivamente a la trayectoria interna del desarrollo tecnológico. Harry Braverman apuntó a uno de sus supuestos principales, la de que las capacidades de los trabajadores se modernizarían cuando los trabajos de obrero fueran reemplazados por trabajos de cuello blanco. Insistió en que la lógica del capital es la de descalificar la fuerza de trabajo, sin tener en cuenta si se emplea en una fábrica o en una oficina. En lugar de una modernización general de calificaciones en la sociedad, predijo que el crecimiento de la así llamada 'economía del servicio' pronto resultaría en que los trabajadores de cuello blanco confronten la rutinización y descalificación así como la que obreros fabriles habían pasado antes.

“Por lejos lo más importante en la producción moderna es la división de procesos complejos en tareas simples que son realizadas por trabajadores cuyo conocimiento es virtualmente nulo, cuyo así llamado entrenamiento es breve, y que entonces pueden ser tratados como partes intercambiables.” [29]

Su afirmación fue rebatida por los sociólogos industriales. Reconocían que la descalificación del trabajo está presente en industrias maduras, pero argumentaban que esta tendencia se contrabalanceaba por el establecimiento de nuevos puestos de trabajo con mayores calificaciones en otros lugares de la economía. A primera vista, la emergencia de la profesión de programador pareció haber probado que sus críticos tenían razón. Uno de ellos, Stephen Wood, reprochó a Braverman por idealizar al trabajador artesanal del siglo XIX. Wood señaló la difusión de la alfabetización para probar que las calificaciones también habían aumentado en la sociedad moderna[30]. Su comentario es intrigante ya que trae una sutileza que se perdió en el calor del intercambio. No es la descalificación *per se* el objetivo del capital, sino el hacer

reemplazables a los trabajadores. Cuando las tareas y cualificaciones se estandarizan, el trabajo será barato y falto de fuerza política. Desde este punto de vista, no importa realmente si las calificaciones de los trabajadores se nivelan en un equilibrio más bajo o más alto. La alfabetización universal es un ejemplo de lo último.

En este sentido puede decirse que la alfabetización es análoga en el presente a las campañas por la alfabetización informática que llaman a cerrar 'la brecha digital'. En un sentido trivial, las calificaciones han aumentado en la sociedad cuando más gente sabe cómo usar computadoras. Uno puede sospechar que un fuerte ímpetu para esto es, sin embargo, que la alfabetización digital reduce una mayor inercia en el esquema de 'aprendizaje de por vida', esto es, el tiempo que toma a los humanos aprender nuevas habilidades. Una vez que los trabajadores adquieren habilidades básicas para navegar en un ambiente digital, toma menos esfuerzo aprender una nueva ocupación cuando su oficio anterior se vuelve redundante. Esta interpretación de alguna manera cínica de la alfabetización informática puede ilustrarse con una referencia a la industria de la imprenta. Los oficios gráficos tradicionales toman muchos años para dominarse y requieren grandes y costosas instalaciones. La militancia sindical que caracterizó la industria de la imprenta se fundaba sobre este monopolio del conocimiento por parte de los trabajadores. La introducción de procesos informáticos fue decisiva para romper la fuerza de los trabajadores gráficos[31]. Las computadoras personales pueden verse como una extensión de este desarrollo. La mediación por el software permite que la simple capacidad de navegar una interfaz gráfica se convierta en múltiples capacidades. Con una computadora corriendo GNU/Linux y Scribus, por ejemplo, el usuario puede comandar el lenguaje de máquina de la computadora e imitar los oficios de la imprenta y la tipografía. Se requiere muy poco entrenamiento para usar estos programas, comparado al tiempo que le toma a un trabajador gráfico dominar su oficio. Esto sugiere que la alfabetización informática reduce la inercia del aprendizaje humano y vuelve las calificaciones de los trabajadores más intercambiables. Los escritores liberales interpretan este desarrollo como un ejemplo del crecimiento lineal del aprendizaje y la educación correspondiente a la llamada 'sociedad del conocimiento'. Desde la perspectiva de la teoría del proceso laboral, muy al contrario, el mismo desarrollo es visto como una degradación de las capacidades de los trabajadores y en última instancia apunta a debilitar la posición negociadora de los sindicatos.

El clásico estudio de David Noble sobre la introducción de maquinaria de control numérico en la industria pesada a mediados del siglo XX provee el vínculo faltante entre el argumento de Braverman sobre la descalificación y la discusión actual sobre computadoras y hackers. Una cosa sobre la que su estudio arroja luz es cómo la

universalidad de la herramienta informática estaba pensada para trabajar en ventaja de los gerentes. Su esperanza era la de debilitar la posición de los maquinistas calificados. La maquinaria de propósito especial había fallado en reemplazar a estos obreros, ya que aún tenían que tomarse iniciativas en la línea de producción para integrar los estadios separados de la producción especializada. En contraste, las máquinas de propósito general simulaban la versatilidad de los seres humanos, por lo que estaban mejor capacitadas para reemplazarlos[32]. Esta conexión histórica es importante de enfatizar porque ahora es un lugar común que la universalidad de las herramientas informáticas se asuma como una cualidad inherente de la tecnología de la información en sí misma. De ahí que la trayectoria hacia herramientas universales se haya desprendido de la lucha y se atribuya en cambio a la gracia del desarrollo tecnológico.

Decir eso no nos obliga a condenar la tendencia a un nivelamiento de las calificaciones productivas y el crecimiento de herramientas universales como las computadoras. Al contrario, en fuerte contraste con el retrato negativo de Harry Braverman como un neo-Ludita, Braverman reconoce que la unificación de la fuerza de trabajo causada por la maquinaria acarrió un potencial positivo.

“El proceso re-unificado en el que la ejecución de todos los pasos es construida en el mecanismo de trabajo de una sola máquina parecería ahora volverlo adecuado a un colectivo de productores asociados, ninguno de los cuales necesita gastar toda su vida en una sola función cualquiera y donde todos pueden participar en la ingeniería, diseño, mejora, reparación y operación de estas máquinas aún más productivas.”[33]

Con una herramienta universal, la computadora, y la casi universal calificación de usarla, el público puede involucrarse en cualquier cantidad y tipo de actividades productivas. Es desde este ángulo que podemos empezar a dar sentido a la tendencia actual de ‘empoderamiento del usuario’. En otras palabras: El desplazamiento del trabajo organizado de los baluartes del aparato de producción capitalista, a través de la combinación de descalificación y recalificación, ha preparado el terreno para esquemas de innovación asistidos por computadoras y centrados en el usuario. Porque programas como *Inkscape* y *Scribus*, y sus equivalentes propietarios, están substituyendo formas tradicionales de imprenta y tipografía, una multitud de gente puede producir pósteres y panfletos, instantáneamente aplicables a sus luchas locales. Las compañías tienen más difícil el control de la actividad productiva ahora que cuando los instrumentos de trabajo estaban concentrados en las manos de unos

pocos, aunque relativamente poderosos, empleados. Lo que es cierto para el diseño gráfico se aplica igualmente a la escritura de código de software y el desarrollo de la tecnología informática. Aquí la cara de Jano del software se pone en primer plano: con la misma flexibilidad y precisión con la que el código de software puede diseñarse para controlar trabajadores subordinados, esta misma facilidad permite a muchos más tomar parte en el proceso de escribirlo. Aunque formas embrionarias de tecnología informática, como la maquinaria de control numérico, fueron introducidas en los lugares de trabajo por los gerentes para independizarlos de trabajadores sindicalizados y calificados; como un efecto lateral, la tecnología informática ha contribuido al establecimiento de procesos productivos centrados en el usuario, parcialmente independientes de gerentes y fábricas. La comunidad de desarrollo de software libre puede tomarse como ilustración de esto.

2.4. El software libre como una estrategia sindical

El apoyo corporativo hacia la comunidad de Software Libre y Abierto (FOSS) debe verse contra el trasfondo de un mercado de trabajo reestructurado. Durante las últimas décadas, los sociólogos industriales han documentado una tendencia donde la fábrica está perdiendo su anterior estatus como el modelo de producción principal. La producción se vuelve cada vez más descentralizada y dispersa en una red de subcontratistas, trabajadores independientes, esquemas de trabajo en casa y franquicias[34]. Las compañías ahora pueden agregar a las comunidades voluntarias de desarrollo a la lista de formas heterogéneas de contratación de trabajo. O, para decirlo con una frase pegadiza, el trabajo es *outsource* y *opensource*. La oportunidad para recortar drásticamente los costos laborales para el mantenimiento de software ha atraído a instituciones estatales, vendedores, proveedores de servicio y manufactureras de hardware hacia el FOSS. Los ahorros que hacen gigantes como IBM, el ejército de EEUU y la ciudad de Munich, para mencionar un par de casos de alto perfil, ha creado el espacio para que firmas de software especializadas vendan productos y servicios de software libre. Este análisis es consistente con la crítica de Tiziana Terranova en la que el involucramiento de trabajo libre se ha vuelto estructural a la economía cultural. Ella protestaba contra los muchos ahelos y reclamos hechos sobre la tendencia al consumo mediático activo, primero celebrada en la disciplina de los estudios culturales desde los '80 en adelante y

más recientemente actualizada con el hype alrededor de la Web 2.0. En respuesta a estas declaraciones a menudo infundadas, Terranova dijo que el capital siempre ha anticipado al consumidor activo en sus estrategias de negocio[35] (2000). Su argumento provee un correctivo a los aplausos acríticos a la subcultura del *fan fiction*, la licencia Creative Commons y otras expresiones de ‘medios participativos’. No obstante, en mi opinión, los críticos de izquierda como Terranova han estado ansiosos de denunciar la explotación económica del trabajo voluntario y han fallado así en ver el cambio político en potencia que también existe en algunos de esos casos.

La relevancia de mi objeción debe decidirse caso por caso. Mientras concedo que la interactividad de los videojuegos y los esfuerzos voluntarios de escritores de *fan fiction* probablemente no resulten en ningún cambio político sustancial, la interactividad y el donar de los desarrolladores de software libre no puede describirse con el mismo pincel. Aquí debe tomarse en cuenta que el código de software es regalado conjuntamente con una meta política claramente articulada: hacer del software libre el estándar en computación. Es verdad que este punto de vista no es anti comercial en un sentido concreto. Como el lector probablemente sepa, la Licencia Pública General (GPL) protege el derecho del usuario a correr el software con cualquier propósito, incluido el comercial[36]. En la práctica, por supuesto, esta opción está limitada por el hecho de que la GPL también permite que las copias vendidas sean copiadas y distribuidas gratuitamente. Mientras la licencia libre reside perfectamente en un mercado libre idealizado, difícilmente lo sea dentro del mercado existente que presupone siempre los cuasi monopolios y las regulaciones estatales[37].

Esto explica de alguna manera por qué la derecha política tiene dos visiones sobre las licencias de software libre. Los autoproclamados *libertarians*, como Eric Raymond, ven el crecimiento de los modelos de negocio basados en código abierto como una mejor aproximación al libre mercado. Detrás de esta afirmación descansa un entendimiento del capitalismo como básicamente idéntico a sus instituciones, es decir propiedad privada, libre mercado y contratos. Pero esa mirada desecha otras posibles definiciones de capitalismo que hacen énfasis en el capital como autoexpansión del dinero, o, en otras palabras, en la acumulación. Este último punto de vista es central al análisis que hace Marx del capitalismo, pero también es muy cercano a las preocupaciones de los ‘capitanes de la industria’. Con esto en mente, puede ser interesante notar investigaciones de mercado que señalan que la adopción de aplicaciones de FOSS en negocios está comiéndose los ingresos anuales de vendedores de software propietario en US\$60 billones por año. Crucialmente, las pérdidas del mercado de software propietario son desproporcionadas con respecto al tamaño de los nuevos

mercados de FOSS, por la simple razón de que mucho de este no se paga.⁴ De ahí que la oposición al FOSS por parte de sectores de la industria no sea necesariamente una postura equivocada, como generalmente se quiso hacer pasar. Esta oposición alcanzó su clímax en la corte en el caso del Grupo SCO contra vendedores corporativos de GNU/Linux que terminó en 2007. Durante el caso, el oficial ejecutivo del Grupo SCO, Darl McBride, escribió una carta abierta al Congreso de EEUU donde acusaba a sus competidores de ser ingenuos al apoyar las licencias FOSS: 'A pesar de esto, estamos determinados a llevar estos casos legales hasta el final porque estamos firmes en nuestra creencia de que la difusión incontrolada del software de Código Abierto, bajo la GPL, es una amenaza mucho más seria para nuestro sistema capitalista de lo que las corporaciones estadounidenses creen.'⁵

Por lo menos, estas preocupaciones entre algunas partes de la industria informática muestran que los desarrolladores de software libre no pueden adscribirse como meras víctimas insospechadas de la explotación comercial. Tal vez esté más justificado decir que los hackers, al ofrecer su trabajo libremente, están chantajeando a las corporaciones para que adopten y difundan el modelo de desarrollo de FOSS. Ninguna compañía que responda al imperativo del mercado de reducir costos puede argumentar contra el trabajo libre (libre como en cerveza gratis). Mi hipótesis es que la militancia por las licencias libres puede interpretarse a la luz de la profesión emergente del programador de computadoras. Esta sugerencia está lejos de ser obvia ya que la identidad hacker está atada a la noción de ser un hobbista, o, en otras palabras, un no-profesional, no-empleado. Contradiendo esta autoimagen, sin embargo, los números dicen que la mayoría de la gente que contribuye a proyectos de software libre está trabajando en la industria informática o se está entrenando para convertirse en profesionales informáticos [39]. Por lo tanto, no sería inverosímil conectar los puntos entre los hackers y el mercado laboral que los espera. En efecto, esta línea de razonamiento ya ha sido intentada en el famoso artículo de Josh Lerner y Jean Tirole[40]. Ellos trataron de enmarcar el supuesto altruismo de los desarrolladores de software libre en la presunción de la teoría económica neoclásica acerca del 'hombre racional económico'. Los dos autores concluyen que los hackers regalan código a cambio de nada para crearse una reputación y mejorar sus chances de ser empleados más tarde. Sin negar que estos casos puedan existir, no estoy de acuerdo con la presunción del individualismo metodológico que sostiene su

⁴El reporte de investigación de mercado al que se hace referencia es Tendencias en Código Abierto y fue publicado por el Grupo Standish. Dado que el acceso a este informe es restringido, la información sobre este viene de los medios de noticias[38]

⁵<http://url.ca/f6o4> (fecha de acceso: 01-11-2009)

pensamiento. Cuando digo que las licencias de software libre pueden beneficiar los intereses laborales de los programadores informáticos, no quiero decir que es una estrategia racionalmente calculada o que es una explicación exhaustiva de por qué los hackers licencian su software bajo la GPL. Además, en contraste con Lerner y Tirole, no pienso que estos intereses laborales se persiguen exclusivamente a través de estrategias individuales. Adicionalmente a mejorar su propia reputación, los hackers individuales contribuyen a cambiar el mercado de trabajo para los programadores como un colectivo.

Suena contraintuitivo que los programadores mejoren su fuerza de negociación vis-à-vis las firmas al regalar su trabajo a empleadores potenciales. Déjeme empezar por regresar a Harry Braverman. Él enfatizó que el mismo diseño de la fábrica ponía al operador de la maquinaria en desventaja. El obrero sólo podía emplear sus calificaciones cuando le daban acceso a la maquinaria. Desafortunadamente, la escala y el modo de organización de la fábrica ya estaban sesgados jerárquicamente. El capitalista tenía una ventaja gracias a la propiedad de las máquinas y los edificios, sin los cuales los obreros no podían emplear sus habilidades. Los únicos pedacitos de negociación que tenían los obreros eran sus calificaciones y su conocimiento íntimo del proceso de producción. Fue así como Braverman explicó también la tendencia de los capitalistas a desarrollar nuevas tecnologías que reducen el trabajo calificado. Lo que pasó desde que Harry Braverman hizo su análisis en los '70 es que la fábrica fordista de gran escala se volvió obsoleta en muchos sectores de la economía. Esto es particularmente verdadero para la industria informática. Las herramientas productivas (computadoras, redes de comunicación, algoritmos de software e información) están disponibles en tales cantidades que se han convertido en el estándar común en lugar de ser una ventaja competitiva contra otros propietarios (capitalistas) y un umbral para los no propietarios (obreros). Una horda de sociólogos industriales y filósofos gerenciales han escrito sobre esta tendencia desde principios de los '80[41]. Es un truismo de este cuerpo literario afirmar que los empleados, no el parque fabril, son en la actualidad el recurso más valioso en una corporación moderna. Esta afirmación está nublada por la retórica, pero su validez puede probarse contra la adopción de 'acuerdos de no divulgación' dentro de la industria informática. Es aquí donde se establece que el empleado no tiene permitido divulgar información sensible de la firma. Otro tipo de cláusulas que a veces se incluyen en el contrato de empleo suelen producir el mismo efecto, es decir, prevenir fugas, prohibir al programador trabajar en tareas similares para un competidor después de haber dejado al empleador actual. Estos acuerdos pueden tomarse como testimonios de que el conocimiento y las calificaciones de los programadores se están volviendo

cada vez más preciosos para que las firmas quieran controlarlos. Argumentaré que estas prácticas, aunque tienen muy poco que ver con la regulación del copyright, sin embargo apoyan mi afirmación de que las licencias propietarias y libres afectan la posición negociadora de los desarrolladores de software.

La justificación de estos diferentes tipos de acuerdos contractuales es la necesidad de prevenir la fuga de secretos de negocio hacia la competencia. No obstante, como un efecto lateral, previene que los programadores se muevan libremente a otras posiciones similares de su oficio. Dado que el programador se convierte en un especialista en el campo en el que está trabajando, puede hacerse difícil encontrar trabajo en un puesto distinto. El significado de esta observación se vuelve más claro si se lo contrasta con el estudio etnográfico de Sean O'Riain de un grupo de técnicos de software de una firma informática de Irlanda. Aquí se prueba que es muy difícil para los sindicatos organizar a estos trabajadores. Ya que los trabajos se proveen a destajo, las estrategias colectivas de los sindicatos carecen de agarre. Una de las conclusiones de O'Riain es que la movilidad se convierte en el medio principal por el que los empleados negocian sus condiciones laborales y salarios[42]. Al tener en cuenta este hecho, el significado de los acuerdos contractuales mencionados arriba debe ser reconsiderado. Las limitaciones que ponen a la capacidad de los empleados a 'votar con los pies' significa que la firma regana la ventaja. En cuanto al alcance en que los acuerdos de no divulgación y otras cláusulas son utilizados en la forma maquiavélica que se esboza aquí, es algo que queda por investigar empíricamente. Lo que me interesa en este artículo, no obstante, es que el mismo argumento puede aplicarse a las licencias propietarias en general.

La propiedad intelectual⁶ también se justifica por la necesidad de las firmas de proteger su conocimiento de los competidores. Una justificación complementaria es que la propiedad intelectual es requerida para que los productores puedan cobrar por la información a los consumidores del mercado. Pero la propiedad intelectual también afecta la relación entre la firma y sus empleados, tema menos discutido. Puede decirse que las licencias propietarias previenen la movilidad de los empleados. Asegura que los conocimientos de los programadores queden encerrados en un estándar propietario del que la firma es dueña. Puede trazarse un paralelo en cómo

⁶Muchos críticos de la regulación del copyright y las patentes rechazan las palabras 'propiedad intelectual'. En su opinión, estas palabras están cargadas de connotaciones que confunden al público. En su lugar utilizan las palabras 'monopolio intelectual'. No estoy convencido por este argumento aunque no queda espacio para desarrollar mi contraposición aquí. Es suficiente decir que uso las palabras 'propiedad intelectual' en el artículo porque pienso que la asociación a otras formas de propiedad está enteramente justificada.

el obrero depende del parque de maquinarias del que el industrialista es dueño. Sin acceso a la fábrica el trabajador no puede emplear sus habilidades productivamente. En la industria de la programación, como se mencionó anteriormente, muchas de las herramientas con las que trabaja el programador están disponibles como bienes informáticos baratos (computadoras, etc.). De ahí que la compañía no tiene ventaja sobre el trabajador para proveer estas instalaciones. Pero cuando el código fuente está encerrado por el copyright y las patentes de software, se requieren grandes cantidades de capital para acceder a las herramientas de programación. Como consecuencia, la licencia de software garantiza a la compañía una ventaja sobre el trabajador/programador. Este razonamiento teórico es tanto más difícil de probar empíricamente que la afirmación hecha antes de que las cláusulas en el contrato de empleo podrían usarse para restringir la movilidad de los trabajadores. Aún más, podría ser en un orden de magnitud superior en importancia para las condiciones de trabajo en el sector informático. En efecto, este aspecto de las licencias propietarias orientado a la producción podría ser tan significativo como las justificaciones oficiales a favor de la regulación de la propiedad intelectual, es decir a favor de regular la relación entre la compañía y sus clientes y competidores. Si mis razonamientos hasta ahora son correctos, entonces la Licencia Pública General debería leerse bajo la misma luz. Fui llevado a este pensamiento cuando leí el estudio autorizado de Glyn Moody sobre el modelo de desarrollo del FOSS. Hace la siguiente observación acerca de las condiciones excepcionales de las compañías especializadas en vender servicios relacionados al software libre:

“Porque el ‘producto’ es código abierto, y está disponible libremente, los negocios deben necesariamente basarse alrededor de un tipo distinto de escasez: las capacidades de la gente que escribe y provee ese software.”[43]

En otras palabras, cuando el código fuente se hace disponible públicamente bajo la GPL, las únicas cosas que se mantienen escasas en el mercado son las habilidades que se requieren para emplear productivamente las herramientas de software. Y este recurso es inevitablemente la facultad del ‘trabajo vivo’, para seguir la terminología de Karl Marx. Es así que los programadores pueden obtener una ventaja sobre el empleador cuando están negociando sobre su salario y condiciones laborales. La licencia libre nivela el campo de juego al asegurar que todos tengan acceso al código fuente. Terranova y estudiosos similares están en lo cierto cuando señalan que las compañías multinacionales están en mejor posición para explotar el valor comercial de las aplicaciones de software libre que cualquier programador individual. El

ahorro que hace IBM al usar Apache en sus servidores es, medido en números absolutos, muchas veces mayor a los golpes de suerte de cualquier programador que ha contribuido al proyecto. Aún así, en una segunda lectura, el programador podría estar mejor si existiera un mercado de trabajo para desarrolladores de software libre, en comparación a que no hubiere ocupaciones de este tipo disponibles. Al publicar software bajo licencias libres, el hacker individual no solo está mejorando su reputación y expectativa de empleo, como han enfatizado Lerner y Tirole. También contribuye al establecimiento de un mercado de trabajo donde las reglas del juego se han reescrito, para él y para todos los demás de su oficio. Puede interpretarse como una forma de acción colectiva adaptada al tiempo del individualismo rampante.

Queda por verse si el establecimiento de un mercado de trabajo en desarrollo de software libre se traduce en mejores condiciones laborales, salarios más altos y otros beneficios asociados al activismo sindical. Tal hipótesis debe sustanciarse con datos empíricos. Se necesita investigación comparativa entre gente trabajando independientemente como programadores de software libre y aquellos que trabajan con software propietario. Tal comparación no debe, sin embargo, enfocarse en aspectos monetarios. Tanto más importante es el costado subjetivo de la programación. Un ejemplo de esto es el descubrimiento consistente de que los hackers consideran más divertido participar en proyectos de software libre que trabajar con código propietario[39]. Tampoco creo que las estrategias sindicales escondidas son la única explicación de que los hackers publiquen bajo la GPL. Muy probablemente, la preocupación por las libertades civiles y un ethos antiautoritario dentro de la subcultura hacker sean factores más importantes. Los hackers son un grupo muy heterogéneo para incluirlos a todos en una sola explicación. Pero me atrevo a decir que la expectativa de trabajo merece más atención de la que se le ha dado en la prensa popular y la literatura académica hasta ahora. Aunque no faltan críticas hacia la regulación de la propiedad intelectual, estas objeciones tienden a formularse como una defensa de los derechos del consumidor dentro de la tradición política liberal.

Hay, por supuesto, algunas excepciones dignas de notar. Gente como Eben Moglen, Slavoj Zizek y Richard Barbrook han reaccionado contra la ideología liberal implícita en mucha de la charla acerca de Internet y temas relacionados. Lo han hecho al cortejar la retórica revolucionaria de la Segunda Internacional. Sus ideas son originales y atractivas y a menudo ricas en entendimiento. Sin embargo, la retórica revolucionaria suena extrañamente fuera de lugar cuando se aplica a hackers pragmáticos. Los militantes del software libre podrían estar mejor si buscaran un contrapeso a la hegemonía del liberalismo en la rama reformista del movimiento obrero, es decir en el sindicalismo. Creo que tal estrategia tendrá más sentido

mientras más madure la industria informática. De acuerdo con la línea general de argumentación de Harry Braverman, la profesión de ingeniería de software ya ha sido privada de mucho de su estatus anterior. En efecto, desde los tempranos '60, los escritores de revistas de gerenciamiento han llamado repetidamente al subyugamiento de los programadores bajo el mismo régimen fabril que previamente, y en parte a través de la introducción de la maquinaria informática, había sido impuesto a los trabajadores de cuello azul[44]. Con esta historia en mente me gustaría proponer que la militancia por el software libre, en lugar de caer en la enmienda de libre expresión de la Constitución estadounidense, pueda tomar su credo de la 'Declaración de Derechos Tecnológicos'. Esta declaración fue escrita en 1981 por la Asociación Internacional de Maquinistas en el medio de un conflicto industrial rabioso:

"Las nuevas tecnologías de automatización y las ciencias que las sostienen son el producto de [un proceso] mundial de acumulación de conocimiento que duró siglos. De este modo, los trabajadores y sus comunidades tienen el derecho a compartir las decisiones y el provecho sobre la nueva tecnología."[45]

2.5. Reconocimientos

El autor quisiera agradecer al editor, Stian Rødven Eide, así como a Michael Widerkrantz y Don Williams, por sus comentarios constructivos a los borradores de este artículo.

El Software-martillo¹

3.1. Introducción

El Software Libre surge como movimiento hace alrededor de 25 años, como continuación de una práctica de producción de software que estaba desapareciendo desde los '70s. Era un momento en el cual los primeros hackers² informáticos empezaban a convertirse en comerciantes de software y, por lo tanto, dejaban de compartir sus creaciones en forma de código fuente de programas de computadora.

La utilización de dispositivos legales en la forma de licencias libres que revierten los efectos del copyright –transformándolo en *copyleft*– ha permitido introducir en la lógica de producción de software una ética de cooperación cuyos alcances sociales a gran escala apenas empiezan a entretarse.

La adopción de Software Libre por parte de empresas, dependencias estatales e individuos para sus computadoras personales contradice los monopolios del software, principalmente el de Microsoft. Sin embargo, estas prácticas colaborativas

¹Este artículo se libera bajo la licencia CC-BY-SA[1]

²El sentido correcto del término no es ni pirata ni criminal informático. *Hacker* es todo aquel que utiliza su creatividad para resolver un problema de una forma novedosa. Existen elaboraciones de corte ético de cuáles prácticas pueden considerarse como *hacking* y cuáles no. Cometer intrusiones maliciosas, por ejemplo, no. Este tipo de actividad sería propia de un *cracker*.

y sus producciones están siendo utilizadas para revalorizar el capital en formas distintas a las “tradicionales”.

Este trabajo intenta establecer algunos puntos en común entre discusiones epistemológicas acerca de la neutralidad tecnocientífica y sus alcances sociales y los del movimiento que nos ocupa, tomando los aportes de Enrique Marí y Oscar Varsavsky.

3.2. La ciencia martillo

Enrique Marí nos recuerda que las formas dominantes de la filosofía son producto del juego de fuerzas sociales de una época, siendo la epistemología, la filosofía de las ciencias, la rama preponderante en la actualidad[46]. Su función principal ha sido la de enunciar este juego al nivel del pensamiento, forjar un mundo del pensamiento que proporcione instrumentos de justificación teórica del desarrollo del capitalismo [47].

La epistemología de corte enunciativo ha contribuido a dividir y compartimentalizar la historia ‘interna’ de la historia ‘externa’ de las ciencias (o de la Ciencia en este caso). Historia del desarrollo de las ideas y del conocimiento puros, por un lado, e historia de lo demás, es decir de las condiciones de posibilidad de las teorías científicas y de las consecuencias, no sólo éticas sino también prácticas de la aplicación del conocimiento científico. No es de extrañar, entonces, que las ciencias y sus productos se presenten muchas veces como *neutrales*, un martillo que a veces es usado para clavar clavos y otras para romper cabezas cada vez más eficientemente. En esta concepción, los científicos no deberían ser responsables de los usos que se hagan de su producción, puesto que ésta no es en modo alguno política. Se establece entonces una escisión entre la ética del investigador y la producción tecnocientífica.

Sin embargo, las ciencias no dejan de ser prácticas sociales aunque el análisis que se haga de ellas se base sólo en la lógica y adecuación de sus procedimientos. Siguiendo a epistemólogos no-enunciativos como Bachelard y Althusser, Marí indica que las condiciones de aplicación de una teoría científica, su *para qué*, se encuentran siempre y en cada nivel del proceso de conceptualización científico.

Bajo el título de sociologismo o reduccionismo sociologista, lo que queda soslayado son los modos de producción científico-tecnológicos. La idea de una ciencia martillo cumple en este sentido una función ideológica dentro de la producción capitalista de conocimiento. Las creaciones de la ciencia actual, dice Varsavsky, no son neutras ni de todo tipo, sino sólo las que el sistema le estimula a crear [48]. La

distribución diferencial de capital en distintos sectores de la investigación científica se realiza según las necesidades del mercado global, que imprimen en consecuencia sobre las ciencias la misma lógica productivista, de constante obsolescencia de productos, que lo mueve y reproduce. Si la llamada libertad de investigación está dirigida hacia las necesidades del mercado, no puede hablarse en ningún momento de neutralidad científica. Para una ciencia nueva, revolucionaria según Varsavsky, se torna indispensable pensar y practicar las ciencias en base a la importancia local de sus productos, poniéndolas en línea con una visión y práctica transformadora de la sociedad.

Otro tema que queda soslayado en las posiciones estándar acerca de las ciencias es la incidencia de los sistemas de propiedad privada de las ideas, la llamada 'propiedad intelectual', que convierte por arte de magia los bienes intangibles en bienes materiales –legalmente hablando, en bienes rivales[49]. Los descubrimientos científicos no son del dominio público, pertenecen a alguien, generalmente a quien espera obtener un rédito económico por ellos. No es este el caso sólo para las ciencias prácticas, ya que sucesivos cambios en las legislaciones de distintos Estados han permitido no sólo el patentamiento de seres vivos, desde bacterias hasta especies vegetales, pasando por el genoma humano, sino también, el de algoritmos matemáticos y otros que podrían ser catalogados dentro del conocimiento puro. Problemática que en los últimos años intenta imponerse al resto del mundo desde organizaciones internacionales como la OMPI o tratados internacionales “bilaterales” como el ACTA.

Vemos entonces, que un análisis epistemológico completo no puede ni debe dejar de lado el carácter social de las ciencias ni el de la responsabilidad ética de sus actores. Comprender la producción científica como una práctica social nos ayuda a recuperar la historia de sus prácticas y producciones y entender el lugar que ocupan en el desarrollo de la sociedad.

3.3. El software martillo

Libertad 0: Libertad de ejecutar un programa con cualquier propósito;

Libertad 1: Libertad de estudiar su funcionamiento y adaptarlo a tus necesidades –su condición indispensable es tener acceso al código fuente;

Libertad 2: Libertad de distribuir copias y ayudar a tus vecinos;

Libertad 3: Libertad de mejorar el programa y distribuirlo para el bien de toda la comunidad –su condición indispensable es tener acceso al código fuente; [50]

Fue el movimiento del Software Libre el que puso sobre la mesa el problema de la neutralidad tecnocientífica en el campo de la programación y algunos años más tarde, la cultura en general. La definición lógica de lo que significa que una pieza de software sea libre, resumida en las cuatro libertades fundamentales que encabezan este apartado, se enfoca en las consecuencias sociales de la práctica de los programadores. Si el software es libre cuando sus usuarios pueden hacer uso de esas herramientas con cualquier fin -estudiarlo, compartirlo y adaptarlo a sus necesidades con la condición de no sustraer estas capacidades-, este vuelco ético quitaría necesariamente la apariencia de neutralidad al proceso de producción y consumo de software. En un mundo en el que los dueños de la tecnología informática se arrojan cada vez más el derecho a un control absoluto de sus productos antes y después de vendidos, las aplicaciones informáticas libres en tanto productos tecnológicos publicados con licencias libres³ permiten mayores niveles de participación a los usuarios y una evaluación de importancia y adaptación local diferentes de las necesidades del mercado global.

En términos generales, el proceso ha dado muy buenos resultados. Hoy día es posible realizar todas las tareas comunes y no-comunes sobre una computadora personal con herramientas completamente libres y de bajo (o nulo) costo, empezando por el sistema operativo, siendo GNU/Linux el más conocido.⁴ Incluso en múltiples y muy variadas formas.

Pero aparte de sus logros meramente técnicos, el movimiento del Software Libre ha dado lugar e institucionalizado una ética de la colaboración, que algunos autores señalan como un nuevo modo de producción entre pares paralelo pero imbricado al capitalista [51], capaz de auto-organizar el trabajo [52]. Sin embargo, este proceso no es unívocamente anti-capitalista; ha sido también aprovechado por

³Una licencia libre, como la Licencia Pública General de GNU (GNU GPL), es un dispositivo legal que bajo la forma de un contrato de uso permite mantener las cuatro libertades invirtiendo el sistema de copyright en copyleft.

⁴Habitualmente conocido como 'Linux'. Existe toda una discusión dentro de la comunidad de usuarios, dividida entre quienes consideran desde un punto de vista técnico que un sistema operativo es su núcleo básico, es decir Linux, y quienes consideran que el proyecto GNU merece el reconocimiento de haber iniciado y sostenido el concepto de sistema operativo libre, es decir desde un punto de vista político.

grandes compañías para capitalizar la creatividad y el trabajo voluntario del público mientras se despoja al software de su componente ético.

El Software Libre se convierte en Software de Código Abierto⁵ para el mercado. Para algunos tecnólogos de esta corriente, ha provocado un cambio de paradigma en sentido kuhniano en cuanto a las formas de producción de valor de la industria informática al que hay que adaptarse [53]. Al poner el énfasis sobre aspectos meramente técnicos acerca de la producción y distribución de software, se desvía la discusión hacia la neutralidad de las herramientas, donde el debate acerca de la utilidad social de la tecnología queda reducido a la capitalización de las contribuciones individuales.

En un reciente discurso acerca del problema de la libertad de los usuarios en la época de 'la nube',⁶ Eben Moglen señala el nuevo desafío del movimiento del Software Libre: volver obsoleto el sistema de explotación de la privacidad levantado por compañías como Google y Facebook [54]. Las empresas de este tipo capitalizan información del comportamiento de las personas al recolectar todos y cada uno de sus movimientos en la red. Los usuarios obtienen servicios, como correo electrónico y almacenamiento de archivos, con el agregado de espionaje gratuito que permite enviar y mostrar avisos publicitarios finamente adaptados a cada uno de ellos.

Paradójicamente, el surgimiento de estos modelos de negocio fue facilitado en gran medida por desarrollos de software libre o abierto de gran calidad y costo de reproducción cero, sumado a una estructuración jerárquica cliente-servidor de una red que originariamente estaba pensada como y compuesta de pares iguales; mientras que la concentración de los registros de uso y flujo de la información en servidores centrales permitió su explotación. Para Moglen, el problema social al que se enfrenta el movimiento es producto de haber obviado por mucho tiempo las consecuencias sociales a largo plazo del desarrollo tecnológico en su conjunto.

La posibilidad de hacer uso de Software Libre, esto es, software con una gran carga ética, en ámbitos militares⁷ ha despertado diversas críticas señalando

⁵Open Source en inglés.

⁶Término difuso para referirse a una nueva administración de Internet en que los datos se almacenan en medios virtualizados. Dice Moglen: "La nube significa que los servidores dejaron de estar hechos de metal. 'Nube' significa que la virtualización de los servidores ha ocurrido [...] ya no podemos señalar en la dirección del servidor y por eso no tenemos medios extra-técnicos o no-técnicos confiables para controlar este desastre en cámara lenta." Se refiere a que no hay forma de controlar dónde y cómo se almacenan los datos y en consecuencia para qué son usados. Ver el primer artículo de esta edición.

⁷En Argentina se está desarrollando una distribución del sistema operativo GNU/Linux para uso del Ejército Nacional.

esta contradicción. Si volvemos a revisar la libertad 0, podemos observar que se convierte en una garantía de neutralidad tecnológica. Su elección pragmática ha sido explicitada en el artículo “*Censorship Envy and Licensing*” [Envidia de la censura y licenciamiento] [55], en el que la Fundación del Software Libre (FSF) asume que las licencias de software podrían impedir legalmente el mal uso del software que protegen, pero de esa forma las licencias se convertirían en una lista de prohibiciones gracias al efecto *envidia de la censura*.⁸ Desde otro punto de vista, las licencias libres se encontrarían con una imposibilidad lógica dentro de su propia formulación fundamental: al prohibir usos no-éticos otorgarían a los programadores el poder de veto sobre los usos posibles de este software-ahora-martillo. Proveer herramientas políticamente neutrales se convierte entonces en una estrategia para atraer la mayor participación posible.

Pero como vimos en la reflexión de Moglen, una planificación positiva de las metas sociales del movimiento podría volver obsoleto en principio cualquier intento de uso no-ético (en el sentido de nocivo para la sociedad) del software sin modificar el dispositivo legal que lo protege. La orientación hacia la justicia social es una estrategia que los hackers no han desarrollado todavía.

3.4. Conclusiones

El objetivo de este texto es demostrar cómo la crítica de las posiciones neutralistas sobre las ciencias pueden ser de ayuda para comprender los procesos que está atravesando el mundo de la informática y el movimiento del Software Libre en particular.

Si las ciencias y su filosofía son producto del juego de fuerzas dominantes, el modo en que se organiza su producción no puede estar en contradicción con esos condicionamientos. Sin embargo, vemos que en el caso de la tecnología informática surge como contra-práctica de la producción centralizada, dictada por el mercado, una forma de entender el software que no lo desliga de sus consecuencias sociales. El movimiento del Software Libre, no sin algunas contradicciones, ha puesto en duda los alcances sociales de la adopción masiva de tecnología desde sus comienzos, dándole un cariz ético-político.

La implementación desigualitaria de tecnologías de la información percibidas o elegidas como neutrales, permite a compañías como Google o Facebook concentrar

⁸“Si mi vecino puede prohibir los símbolos que no le gustan, ¿por qué no puedo hacer lo mismo?”

el flujo de información de la red en desmedro de la privacidad de una sociedad *conectada*.

El problema del software que, como un martillo que a veces devuelve la soberanía tecnológica a las sociedades y otras la despoja de su creatividad y privacidad, se resuelve en la práctica de programar e implementar servicios cuyo énfasis esté orientado hacia una comprensión ética de las costumbres y necesidades de las personas. En la praxis hacker, se trata de arreglar los problemas, como si se tratara de un *bug*, o como propone el título del libro de Söderberg, *hackear al capitalismo*.

Benjamin Mako Hill

4

Piratería y Software Libre¹

Este ensayo es un resumen de mi presentación en el taller “Inlaws and Outlaws”[56] realizado los días 19 y 20 de agosto, 2010 en Split, Croacia. El taller reunió militantes de la piratería con participantes de los movimientos por la Cultura y el Software Libres.

En “Por qué el software no debería tener dueños”[57], Richard Stallman explica que, si un amigo te pide una pieza de software y la licencia de ese software te impide compartirlo, vas a tener que elegir entre ser un mal amigo o violar la licencia del software. Stallman sugiere que los usuarios tendrán que elegir el menor de dos males y elegirán violar la licencia. Enfatiza que es injusto pedir a un usuario que realice tal elección.

En los últimos años, los partidos piratas[58] han crecido a través del mundo desarrollado. Por supuesto, la piratería es el medio principal de distribuir medios entre todos los demás. Los militantes del acceso a la información se han juntado y organizado bajo la bandera ‘pirata’, representando la elección de compartir con amigos sobre el cumplimiento de los términos de las licencias.

¹Este artículo es la traducción al castellano de “Piracy and Free Software”. El artículo original en inglés puede encontrarse en <http://mako.cc/copyrighteous/20101011-00>. Traducción por Nicolás Reynolds , septiembre, 2010. Este artículo y el original están licenciados bajo Creative Commons Atribución-CompartirIgual 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>.

Tanto el movimiento del Software Libre y Abierto como el de la Cultura Libre parecen tener reacciones confusas y contradictorias a esto. Por un lado, los mayores proponentes de varios partidos piratas[59, 60] son también militantes por el Software Libre y varios partidos piratas han hecho de la militancia por el Software Libre uno de los componentes de sus plataformas políticas. La clara oposición del Partido Pirata a las patentes del software y el DRM resuena junto a la de las comunidades del Software y de la Cultura libres. Por otro lado, líderes del movimiento del Software Libre, incluyendo a Stallman, nos han prevenido acerca de las políticas 'piratas' anti-copyright[61]. Líderes de la Cultura Libre, como Lawrence Lessig, han repetida y vociferadamente denunciado la piratería[62, 63, 64], hasta tratado la sugerencia de una asociación con la piratería como una afronta, y se han distanciado sistemáticamente de ella.

¿Deberían los militantes del Software y la Cultura libres aceptar a los piratas como camaradas de armas o condenarlos? ¿Debemos elegir entre estar con los piratas o contra ellos? Nuestras comunidades parecen no haber podido articular un consenso claro y consistente.

Creo que, inintuitivamente, si se toma una fuerte posición de principios [65] a favor de la libertad de la información y se distingue entre los principios y las tácticas, es posible llegar a un 'término medio' más matizado en respuesta a la piratería. A la luz de un convencimiento de que los usuarios deben poder compartir información podemos concluir que no hay nada éticamente malo con la piratería. Las licencias tienen el poder de la ley pero están protegidas por leyes de 'propiedad intelectual' injustas. Habiendo dicho esto, los principios no son la única razón por la que los activistas deciden hacer las cosas. Muchos intentos políticos son malas ideas no porque estén mal, sino porque no van a funcionar y tienen efectos negativos. Las tácticas importan también. Aunque no haya nada éticamente malo con la piratería desde la perspectiva del software o la cultura libres, todavía puede ser una mala idea. Hay al menos tres razones tácticas que pueden motivar al software y la cultura libres a no apoyar la piratería o a participar en movimientos y políticas pro-piratería.

Primero, el sistemático no respeto al copyright mina el respeto por todas las licencias que tienen un enorme beneficio táctico para el Software Libre y un factor cada vez más importante para el éxito de la Cultura Libre. Las licencias copyleft como la GPL[66] o CC-BY-SA[67] tienen poder sólo porque el copyright lo tiene. Como Stallman ha sugerido[61], las acciones anti-copyright son también acciones anti-copyleft. Esto no tiene por qué ser un argumento contra los intentos para limitar el copyright. En efecto, pienso que debemos limitar y reducir el copyright. Pero debemos hilar cuidadosamente. En el clima actual del copyright, debemos sopesar

los beneficios de promover la violación sistemática de las licencias propietarias con los beneficios de adherir al Software y la Cultura libres.

Segundo, la piratería es fundamentalmente reaccionaria. Parte de su resonancia como un símbolo político viene del hecho de que la piratería representa una manera en que los consumidores de medios pueden luchar contra un grupo de compañías que los han atacado –con demandas judiciales, sistemas DRM, y demonización y propaganda– por compartir en formas que los consumidores piensan como naturales y socialmente positivas. Pero la piratería se enfoca en la reacción antes que en la importancia fundamental de compartir que la conduce. Como resultado, muchos piratas no apoyan, o ni siquiera están familiarizados, con un acercamiento de principios al acceso a la información. Como resultado, muchos militantes piratas que hablan contra el DRM en DVDs estarían felices de usar Netflix para descargar películas con DRM por US\$5 al mes así como lo estarían por descargarlas gratuitamente. Los mejores gritos de guerra no siempre se traducen en los movimientos más robustos.

Tercero, en este foco sobre la reacción, un diálogo acerca de la piratería evita el compromiso con las preguntas difíciles sobre con qué vamos a reemplazar el actual sistema de copyright. Una posición de principios sugiere que es nuestra prerrogativa ética crear modelos alternativos. El movimiento del Software Libre ha triunfado porque creó tal prerrogativa y entonces, lentamente, proveyó ejemplos de alternativas funcionales. Una posición de principios sobre el Software Libre no requirió que se provea un nuevo sistema en funcionamiento inmediatamente, pero hace del desarrollo de acercamientos creativos y sostenibles una prioridad. Atacar al sistema sin siquiera *tratar* de hablar sobre modelos alternativos de producción no es sustentable. El Software y la Cultura libres piden una revolución. La piratería sólo pide una revuelta.

La piratería, en estos tres sentidos, puede verse como tácticamente imprudente, sin ser necesariamente anti-ética. Al tomar una posición de principios, uno puede construir e ir más allá, del comentario de Richard Stallman. En términos de software y cultura libres, podemos sugerir que la piratería no es éticamente mala, pero que es una manera imprudente de promover el compartir. Sin ser hipócrita, podemos decir “No pienso que la piratería sea anti-ética. Pero tampoco la apoyo”.

Lucy Watts

5

Richard Stallman:
Qué es el Software Libre¹

¹Las ilustraciones de Richard Stallman son obra de Lucy Watts[68] y están licenciadas bajo CC-BY-SA[67].

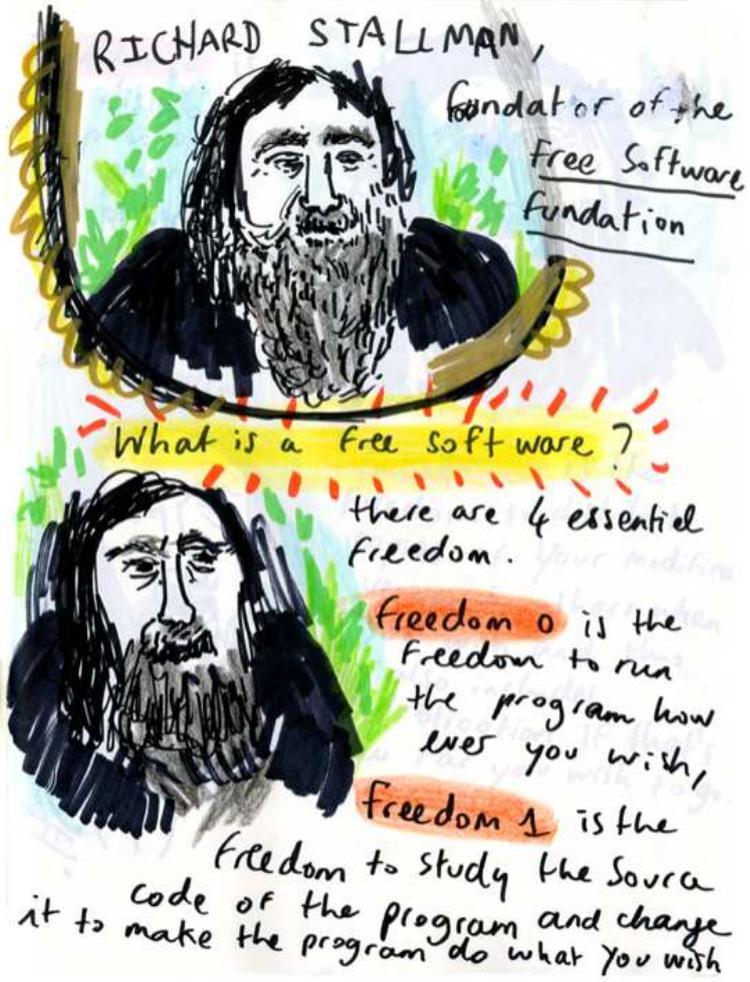


Figura 5.1: RICHARD STALLMAN, fundador de la Fundación por el Software Libre (FSF) ¿Qué es un software libre? Hay 4 libertades esenciales. Libertad 0 es la libertad de usar el programa como quieras, Libertad 1 es la libertad de estudiar el código fuente del programa y cambiarlo para que haga lo que quieras.

Freedom 2 is
to distribute
copies of
the program
to others
when you wish,
now this includes
republication
of the program



FREEDOM 3 is the
freedom to distribute
copies of your modified
version to others when
you wish and this
also includes
publication if that's
how far you wish to go.

Figura 5.2: Libertad 2 es la de distribuir copias del programa a otros cuando quieras, esto incluye la republicación del programa. Libertad 3 es la libertad de distribuir copias de tu versión modificada a otros cuando quieras y esto también incluye la publicación si es lo que quieres hacer.

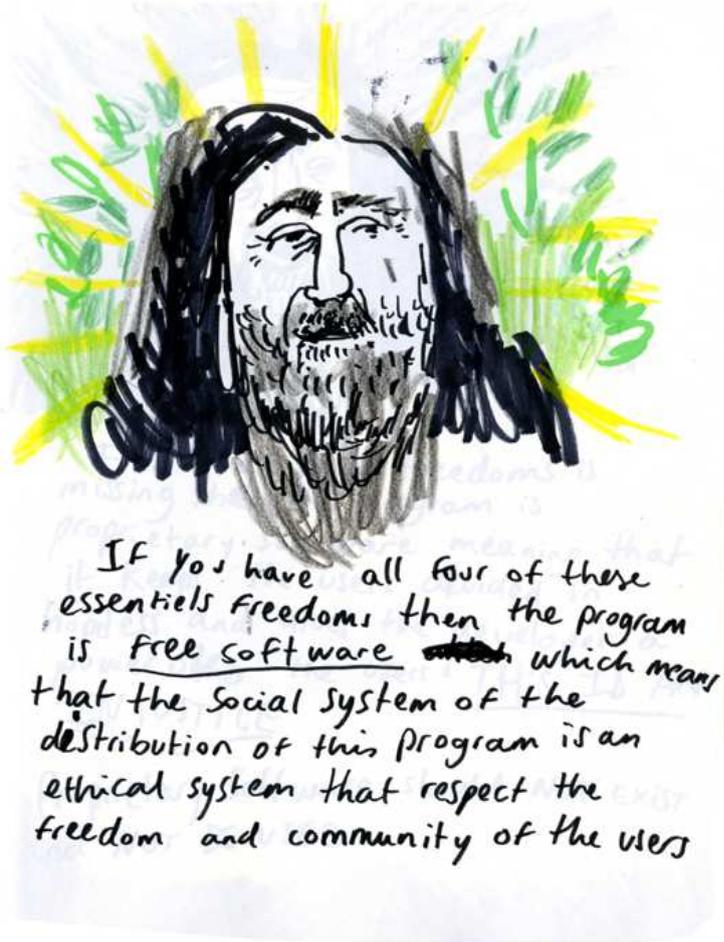


Figura 5.3: Si tenés estas cuatro libertades esenciales el programa es Software Libre lo que significa que el sistema social de la distribución del programa es un sistema ético que respeta la libertad y comunidad de los usuarios.

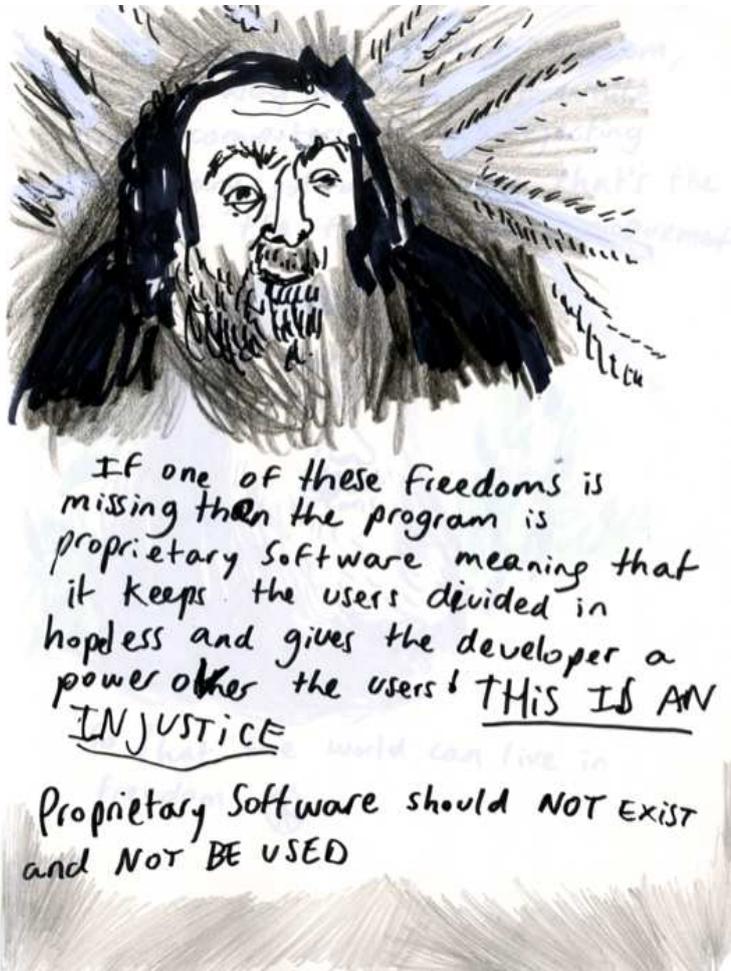


Figura 5.4: Si falta una de estas libertades entonces el programa es software propietario lo que significa que mantiene a los usuarios desesperanzadamente divididos y da al desarrollador poder sobre los usuarios. ESTO ES UNA INJUSTICIA. El software propietario NO DEBE EXISTIR y NO DEBE USARSE

If you want to have freedom,
the only way of having it while
using computers is by rejecting
proprietary software and that's the
aim of the free software movement



So that the world can live in
freedom. ☺

Figura 5.5: Si querés tener libertad, la única manera de hacerlo al usar computadoras es rechazando el software propietario y ese es el objetivo del movimiento del Software Libre. Para que el mundo puede vivir en libertad.

Bibliografía

- [1] Creative Commons Corporation. Licencia Creative Commons Reconocimiento-CompartirIgual 2.5 Argentina.
<http://url.ca/2djpm>.
- [2] T. Jordan. *Activism!: Direct Action, Hacktivism and the Future of Society*. Reaktion Books, London, 2002.
- [3] L. Torvalds and D. Diamond. *Just For Fun – The Story of an Accidental Revolutionary*. HarperCollins Publisher, New York, 2001.
- [4] G. Coleman. Hacker practice: Moral Genres and the Cultural Articulation of Liberalism. *Anthropological Theory*, 8:255–277, 2008.
- [5] E. Moglen. Anarchism Triumphant: Free Software and the Death of Copyright.
<http://url.ca/f6nx>, 1999.
- [6] R. Barbrook. The Regulation of Liberty: Free Speech, Free Trade and Free Gifts on the Internet. *Science as Culture*, 11.(2.):150–177, 2002.
- [7] S. Zizek. A Cyberspace Lenin: Why Not? *International Socialism Journal*, 95, 2002.
- [8] A. Pollack. in (ed.) Robert McChesney, Ellen Meiksins Wood & John Bellamy Foster, *Capitalism and the Information Age – The Political Economy of the Global Communication Revolution*. 1998.
- [9] P. Edwards. *The Closed World: Computers and the Politics of Discourse in Cold War America*. MIT Press, Cambridge, 1996.
- [10] D. Shiller. *Digital Capitalism: Networking the Global Market System*. MIT Press, London, 1999.
- [11] F. Webster. *Theories of the Information Society*. Routledge, New York, 2002.
- [12] G. Sussman and J. Lent. *Global Productions – Labour in the Making of the 'Information Society'*. Hampton Press, Cresskill, 1998.

- [13] Weed. The Dark Side of Cyberspace – Inside the Sweatshops of China’s Computer Hardware Production.
<http://url1.ca/f6ny>, 2008.
- [14] H. Braverman. *Labor and Monopoly Capital*. Monthly Review Press, New York, 1974.
- [15] P. Kraft. *Programmers and Managers: The Routinisation of Computer Programming in the United States*. Springer-Verlag, New York, 1977.
- [16] G. Hannemyr. Technology and Pleasure: Considering Hacking Constructive. *First Monday*, 4(2), 1999.
- [17] M. Berg. *The Machinery Question and the Making of Political Economy*. Cambridge University Press, Cambridge, 1980.
- [18] D. Riccardo. *Principles of Political Economy and Taxation*, 3rd edition, 1821.
- [19] J. Kay. *Moral and Physical Conditions of the Operatives Employed in the Cotton Manufacture in Manchester*, 1832.
- [20] A. Ure. *The Philosophy of manufactures: or, an exposition of the scientific, moral and commercial economy of the factory system of Great Britain*. Knight, London, 1835.
- [21] C. Babbage. *On the economy of machinery and manufactures*. Kelley, New York, 1971.
- [22] K. Marx. *Capital vol. I*. Penguin Books, London, 1976.
- [23] K. Sale. *Rebels Against the Future – The Luddites and Their War on the Industrial Revolution, Lessons for the Computer Age*. Addison-Wesley Publishing Company, Reading, Mass., 1995.
- [24] E. Hobsbawm. The Machine Breakers. *Past and Present*, 1:57–70, February 1952.
- [25] A. Ross. *Strange Weather – Culture, Science, and Technology in the Age of Limits*. Verso, London, 1991.
- [26] B. Shell and J. Dodge. *The Hacking of America: Who’s Doing it, Why, and How*. Quorum Books, Westport, CT., 2002.
- [27] G. Brown. *Sabotage – A study in Industrial Conflict*. Spokesman, Nottingham, 1977.
- [28] D. Bell. *The Coming of Post-Industrial Society: A Venture in Social Forecasting*. Basic Books, New York, 1973.
- [29] H. Braverman. *Labor and Monopoly Capital*. Monthly Review Press, New York, 1998.

- [30] S. Wood. *The Degradation of Work? Skill, Deskilling and the Labour Process*. Hutchinson, London, 1982.
- [31] A. Zimbalist. in: (ed.) Zimbalist, A. *A Case Studies of the Labor Process*. Monthly Review Press, New York, 1979.
- [32] D. Noble. *Forces of Production – A Social History of Industrial Automation*. Alfred A Knopf, New York, 1984.
- [33] H. Braverman. *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century*. Monthly Review Press, New York, 1998.
- [34] R. McChesney, E. Wood, and J. Foster. *Capitalism and the Information Age: The Political Economy of the Global Communication Revolution*. Monthly Review Press, New York, 1998.
- [35] T. Terranova. Free Labour: Producing Culture for the Digital Economy. *Social Texts*, 18:33–57, 2000.
- [36] J. Gay. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Cop, Boston, 2002.
- [37] K. Polanyi. *The Great Transformation: The Political and Economic Origins of Our Time*. Beacon Press, Boston, 2001.
- [38] M. Broersma. Proprietary Vendors Lose £30bn to Open Source, April 2008.
- [39] K. Lakhani and R. Wolf. *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Projects in: Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K., Cusumano, M. & Shirky, Clay (eds.) Perspectives on Free and Open Source Software*. MIT Press, Cambridge, 2005.
- [40] J. Lerner and J. Tirole. Some Simple Economics of Open Source. *Journal of Industrial Economics*, 50 (2):197–234, 2002.
- [41] S. Zuboff. *In The Age Of The Smart Machine: The Future Of Work And Power*. Basic Books, New York, 1998.
- [42] S. O’Riain. Net-Working for a Living: Irish Software Developers in the Global Workplace in: Ash, A. & Thrift, N. (Eds.). *Cultural Economy Reader*, 2004.
- [43] G. Moody. *Rebel Code: Linux and the Open Source Revolution*. Penguin, London, 2001.
- [44] G. Dafermos and J. Söderberg. The Hacker Movement as a Continuation of Labour Struggle. *Capital & Class*, 97:53–73, 2009.

- [45] H. Shaiken. *Work Transformed: Automation and Labor in the Computer Age*. Lexington Books, Lexington, Mass., 1986.
- [46] Enrique Marí. *Elementos de epistemología comparada*. Puntosur Editores, 2000.
- [47] Silvia Rivera. "La epistemología y sus formas cambiantes.^{en} Filosofía, política y derecho. Un homenaje a Enrique Marí. Prometeo, 2003.
- [48] Oscar Varsavsky. *Ciencia, política y cientificismo*. Centro Editor de América Latina, 1969.
- [49] Beatriz et. al. Busaniche. *Monopolios artificiales sobre bienes intangibles*. Fundación Vía Libre, 2007.
- [50] Richard Stallman. "La definición del Software Libre", en *Software Libre para una sociedad libre*. Traficantes de Sueños, 1996.
- [51] Michel Bauwens. The political economy of peer production.
<http://ur1.ca/21pb0>, 2005.
- [52] Johan Söderberg. *Hacking Capitalism: The Free and Open Source Software (FOSS) Movement*. Routledge, 2007.
- [53] Tim O'Reilly. The Open Source Paradigm Shift.
- [54] Eben Moglen. Freedom In the Cloud: Software Freedom, Privacy, and Security for Web 2.0 and Cloud Computing.
<http://ur1.ca/uups>, 2010.
- [55] David Turner. Censorship envy and licensing. 2010.
- [56] Ništa se neće dogoditi. Nothing will happen again in Split with Inlaws & Outlaws.
<http://ur1.ca/21mhi>, 2010.
- [57] Richard M. Stallman. La definición del Software Libre.
<http://ur1.ca/3ogz>, 2010.
- [58] Wikipedia. Partidos Pirata Internacional.
<http://ur1.ca/21mhg>.
- [59] Piraten Partei. Party manifesto of the Pirate Party of Germany.
<http://ur1.ca/21mhe>, 2006.
- [60] Pirate Party UK. Copyright and Patents.
<http://ur1.ca/032j0>.

- [61] Richard M. Stallman. Las propuestas del Partido Pirata sueco y el software libre: el tiro por la culata.
<http://url.ca/21mhx>.
- [62] Lawrence Lessig. there he goes again.
<http://url.ca/21mh5>, 2008.
- [63] Lawrence Lessig. News flash: I don't "defen[d] piracy".
<http://url.ca/21mh3>, 2008.
- [64] Lawrence Lessig. ASCAP's attack on Creative Commons.
<http://url.ca/0o9cb>, 2010.
- [65] Benjamin Hill. Taking a Principled Position on Software Freedom.
<http://url.ca/21mh4>, 2009.
- [66] Free Software Foundation. GNU General Public License.
<http://url.ca/k>, 2007.
- [67] Creative Commons Corporation. Licencia Creative Commons Reconocimiento-CompartirIgual 3.0.
<http://url.ca/21mi5>.
- [68] L. Watts. Richard Stallman: What is Free Software?
<http://url.ca/2aq6i>, 2010.

