

XIII Jornadas de Sociología. Facultad de Ciencias Sociales, Universidad de Buenos Aires, Buenos Aires, 2019.

El movimiento software libre. Un estudio exploratorio.

Milagros Dolabani.

Cita:

Milagros Dolabani (2019). *El movimiento software libre. Un estudio exploratorio. XIII Jornadas de Sociología. Facultad de Ciencias Sociales, Universidad de Buenos Aires, Buenos Aires.*

Dirección estable: <https://www.aacademica.org/000-023/615>

Acta Académica es un proyecto académico sin fines de lucro enmarcado en la iniciativa de acceso abierto. Acta Académica fue creado para facilitar a investigadores de todo el mundo el compartir su producción académica. Para crear un perfil gratuitamente o acceder a otros trabajos visite: <https://www.aacademica.org>.

El movimiento software libre. Un estudio exploratorio

Milagros Dolabani

Eje 6 “Cultura, Significación, Comunicación e identidades”. Mesa 102 “Tecnologías digitales, Comunicación y Sociedad”.

Centro de Estudios Históricos (CeHis), Facultad de Humanidades, Universidad Nacional de Mar del Plata (UNMDP), Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

milagrosdola@gmail.com

Resumen

Las transformaciones experimentadas en las últimas décadas en ámbitos diversos como la investigación académica, las actividades empresariales o las formas de sociabilidad e interacción, a partir de la creciente intromisión de las tecnologías de la comunicación y los entornos digitales en la vida cotidiana, demandan un análisis pormenorizado de los modos en que éstos fueron configurados, enfatizando en la reconstrucción histórica de su producción y desarrollo. La exposición de su carácter histórico debe acompañarse del desarrollo de un sentido crítico que grafique el carácter fetichista con la que muchas veces son presentadas, bajo la naturalización de su índole privada y comercial, recapitulando los pasajes que signaron la producción de servicios informáticos en los últimos cincuenta años.

El análisis del software libre entonces, se presenta como un caso rico para la consecución de dichos objetivos, considerando la ética no propietaria y los preceptos desarrollados en torno al copyleft como sus principios rectores, dispuestos al servicio de las necesidades de los usuarios o de las comunidades que lo demanden. El análisis del surgimiento, desarrollo y consolidación del software libre intentará de ese modo cuestionar la neutralidad de las tecnologías de información, en consonancia con las transformaciones experimentadas en la organización del trabajo y el desarrollo científico a partir de la globalización gestada en Occidente.

Palabras clave: Software libre- programación-servicios informáticos

I. Contexto histórico

La producción científica norteamericana en los '50: la red ARPA y la comunidad de “hackers auténticos”

Contextualizar los avances en las tecnologías de la comunicación e información que precedieron al desarrollo del software libre implica necesariamente considerar el impacto de la guerra fría en el desarrollo científico de los Estados Unidos, y su relación con el modelo de producción industrial vigente durante el estado de bienestar. La batalla de información desplegada contra la Unión Soviética constituyó en ese marco un estímulo para el desarrollo de la informática, propiciando importantes avances sobre la capacidad de los ordenadores, la sofisticación de sus sistemas operativos (conocidos como software) y de sus equipos físicos o hardware, en ese entonces concebidos como una unidad.¹ La idea de un desarrollo científico unido a los objetivos políticos, económicos y militares de la nación del norte cobró por entonces fuerza, e importantes partidas presupuestarias beneficiaron la instauración y fortalecimiento de centros de investigación estatales dedicados al desarrollo de servicios informáticos, generando importantes avances en el área.²

Resultó clave entonces un ambicioso proyecto que buscaba la construcción de una red de intercambio de información a salvo de los objetivos militares del enemigo rojo, una “red de redes” que conectara los distintivos centros de cómputos y comunicación a lo largo del país, y cuya concreción requería del despliegue de múltiples esfuerzos científicos y presupuestarios, capaces de volver a las máquinas y sus sistemas operativos en instrumentos aptos para resguardar información confidencial. La creación de la red ARPA (*Advanced Research Projects Agency Network*, Red de la Agencia de Proyectos de Investigación Avanzada) forjó así una comunidad de informáticos, técnicos y programadores en los distintos centros de investigación beneficiados con el proyecto, una generación de “programadores auténticos” que durante las décadas de 1950 y 1960 se dedicaría a trabajar sobre los alcances de la red y sobre la capacidad de las computadoras.³

El Instituto de Tecnología de Massachusetts (MIT, *Massachusetts Institut of Tecnology*,) y la Universidad de California en Los Ángeles (UCLA), fueron los encargados de efectuar las primeras conexiones entre los nodos dispuestos en esos centros, extendiéndose más tarde a los principales polos de investigación de E.E.U.U, configurando la red conocida como ARPA-NET o red ARPA. Con la paulatina extensión de la red por el territorio también emergió un método de trabajo colaborativo, mediante el intercambio de mensajes o ficheros de datos, o del desarrollo de programas o códigos de programación de manera conjunta y por ese medio,⁴

¹ En este punto conviene realizar una distinción entre la máquina o equipo, el sistema operativo, y el lenguaje de programación. En efecto, el lenguaje de programación es un conjunto de instrucciones que determinan las acciones que ejecutará la máquina, compuesto por símbolos y reglas sintácticas y semánticas, las que sumadas a un agregado de datos conforman un programa o software, determinante del comportamiento físico y lógico del equipo, conocido como hardware.

² Flichy, Patrice. *El imaginario de Internet*, Madrid, Tecnos, 2003.

³ Raymond, Eric S.; “Breve historia de la cultura hacker”, en Carlos Gradin (compilador) *Internet, hackers y software libre*, Buenos Aires, Editora Fantasma, 2004; pp. 27-44.

⁴ Flichy, *op. cit.*

formando colegios invisibles o redes de cooperación con su propia cultura, argot y códigos de trabajo. Como lo expresa Flichy “Arpanet no ha sido utilizado, como sus promotores lo habían imaginado, para suministrar recursos informáticos a distancia, sino para comunicarse (...) No solamente los creadores de Arpanet han sido durante mucho tiempo los usuarios del mismo, sino que el modelo cooperativo que estaba en la base de la concepción constituía el contenido mismo del uso.”⁵

Dicha forma de trabajo instituida entre programadores y técnicos situados a kilómetros de distancia perduraría hasta la década del ‘70, en paralelo a la emergencia de hackers,⁶ *hobbyistas* o aficionados a la programación que también desarrollaron avances en la informática por fuera de ámbitos académicos.⁷ Como sostiene Gabriella Coleman, aunque durante la mayor parte de la década de 1970 las computadoras estaban muy lejos de producirse en masa y de ser accesibles al gran público, la conformación de grupos de aficionados o *hobbyistas* atentos a los últimos desarrollos en informática y tecnologías electrónicas, constituyó un actor relevante en la producción de innovaciones en el área.⁸

La expansión y desarrollo de programas informáticos, aplicaciones, sistemas operativos y equipos informáticos así, tomó lugar dentro de espacios disímiles representados por la formación de una comunidad científica, y por el otro lado, de una comunidad de aficionados o entusiastas, no sin presentar heterogeneidades hacia el interior de cada espacio. En ese sentido, emergerán dos posiciones que, perdurables en el tiempo, distinguirán a los hackers que efectuaban cambios en los software o hardware por el simple placer de dominar la tecnología, de quienes veían en los cambios tecnológicos un medio para la consecución de sus propósitos políticos, disponiéndolos al servicio de la comunidad de pertenencia, parte de la herencia contracultural norteamericana de los años ‘60.⁹ Sin embargo, y más allá de esas diferencias, ambas agrupaciones coincidían en un modo de producir avances informáticos sin la persecución de fines comerciales, sin publicidad ni comercio de datos, en un contexto en el que el modelo de producción científica aún no estaba sometido a la aplicación de patentes o leyes de copyright y en el que las grandes compañías de comunicación aun no incursionaban en el terreno de la informática y en las tecnologías ligadas a las computadoras.¹⁰

⁵ Flichy, op. cit., p. 58.

⁶ Por hacker nos referimos a entusiastas de la programación y no al término comúnmente difundido asociado a personas dedicadas a quebrantar sistemas de seguridad.

⁷ Raymond, op.cit.

⁸ Un precedente clave fue *The homebrew computer club*, un grupo de aficionados y hackers del hardware que se reunía periódicamente en Silicon Valley en cuyo seno se desarrolló la primer computadora disponible comercialmente en EEUU, el modelo *Altair*. Coleman, Gabriella, *Coding Freedom. The ethics and aesthetics of hacking*, Princeton, Princeton University Press, 2013.

⁹ Un ejemplo reconocido fue la Whole Earth ‘Lectronic Link (WELL, por sus siglas en inglés), una de las primeras comunidades virtuales creadas por fuera de la academia y de ARPA, congregando a distintos aficionados que, inspirados por las ideas de la contracultura, veían al ciberespacio como un lugar donde llevar a cabo sus ideales. Söderberg, Johan (2019) *Hackeando al capitalismo: el movimiento de software libre y de código abierto*. Traducción y edición por el Partido pirata. Extraído de https://utopia.partidopirata.com.ar/hackeando_al_capitalismo.html [Consulta: 27/06/2019]

¹⁰ Coleman, op. cit.

De esta manera, la lógica de producción y consumo establecida en aquellos espacios se opuso a la regente en el desarrollo de productos comerciales, donde las categorías de productor/consumidor se encuentran escindidas, así como también las de dirección/desarrollo de productos, conservando tanto los conocimientos como las herramientas requeridas para sus proyectos, disponiendo de la libertad para establecer sus propios objetivos y ritmos de trabajo.

Sin embargo, las reestructuraciones que hacia finales de los años '70 se expresaron en los planos económico y político a nivel internacional, trastocarían los usos y prácticas desarrolladas hasta entonces en esas comunidades, exponiendo sus heterogeneidades internas. En efecto, el agotamiento del estado de bienestar y a la emergencia de nuevas potencias comerciales que disputarían el rol desempeñado por los Estados Unidos en el mercado mundial, provocarían una reestructuración económica con el objetivo de sostener su competitividad en el mercado global, virando hacia las industrias de la comunicación, la informática y el desarrollo tecnológico.¹¹

La organización del trabajo experimentó también un cambio de paradigma, y el modelo fordista de producción fue reemplazado poco a poco por el Toyotismo, con la consiguiente informatización de un proceso productivo realizado “a demanda” y según los requerimientos del mercado, bajo la táctica empresarial de “justo a tiempo” y “stock cero”.¹² En este marco, las nuevas tecnologías de la información y comunicación colocaron al conocimiento y al cambio tecnológico en el centro de los procesos productivos, en un contexto signado por la creciente internacionalización del capital, corolario de la globalización que comenzaba a expandirse.¹³ No obstante, la reestructuración del capital respondió menos a las características intrínsecas de los avances informáticos, que a la necesidad de mantener su margen de ganancia, disminuida por la creciente participación de las masas trabajadoras dado el incremento del valor de su fuerza de trabajo, mediante las conquistas adquiridas en el terreno económico y social y las fuerzas contestatarias cristalizadas en mayo del '68.¹⁴

En este contexto, el sistema científico norteamericano debió también reestructurarse, pues su centralización en los departamentos de investigación universitarios, en los organismos públicos y en las oficinas de métodos y de I+D de la “gran empresa”,¹⁵ resultaban inadecuados para unos requerimientos de mercado en transformación, con la nueva organización del trabajo y la “moderna empresa”. Serán las administraciones de Reagan, Bush padre y más tarde Clinton, las encargadas de estimular los cambios introducidos en la orientación económica del país, mediante el despliegue de una serie de medidas destinadas a transformar el modo de producir ciencia, así como la legislación vigente respecto a la propiedad intelectual. El

¹¹ Srnicek, Nick. *Capitalismo de plataformas*, Buenos Aires, Caja Negra, 2018.

¹² Hartd, Michael y Negri, Antonio. *Imperio*. Cambridge, Harvard University Press, 2000.

¹³ Míguez, Pablo. “Del *General Intellect* a las tesis del ‘capitalismo cognitivo’: aportes para el estudio del capitalismo del siglo XXI”. *Bajo el Volcán*, año 13, número 21, septiembre 2013-febrero 2014.

¹⁴ Hartd, Michael y Negri, Antonio, *op cit*.

¹⁵ Ídem.

desfinanciamiento de los principales organismos de investigación pública, el estímulo a la vinculación academia-industria, y la sanción de una serie de leyes para habilitar el patentamiento y la aplicación de derechos de autor a productos, servicios y descubrimientos hasta entonces de uso y producción pública,¹⁶ fueron algunas de las medidas más resonantes aplicadas con ese objetivo.

Centros de investigación como el MIT, con un papel destacado en el desarrollo de avances informáticos durante décadas previas, deberán enfrentarse ahora con el reclutamiento de sus científicos y científicas por parte de grandes compañías de comunicación, así como con la rúbrica obligatoria de acuerdos de confidencialidad y patentamiento de las producciones desarrolladas en ese centro.¹⁷ El patrimonio de grandes corporaciones se vio de ese modo engrosado por la capitalización de trabajo financiado con fondos públicos, mediante el reclamo de derechos de propiedad intelectual y por la aplicación de patentes, en una cruzada mercantilizadora que también avanzaría sobre el diseño y la programación de servicios informáticos.

La aparición de empresas destinadas a la comercialización de sistemas operativos por fuera de ámbitos académicos o bien la incursión de compañías destinadas a la producción de otro tipo de servicios en ese área, fueron un síntoma de cambio de época respecto al modo en que hasta el momento se diseñaba y programaba software. En este marco, el denominado caso Unix muestra de modo gráfico la puja establecida entre empresas y usuarios/programadores, sentado un precedente para el desarrollo de las resistencias que posteriormente emergerían en las comunidades de hackers a nivel internacional.

II. El software libre

El caso UNIX y la privatización de servicios informáticos

El sistema operativo Unix, diseñado y programado por dos empleados de los laboratorios de la compañía telefónica norteamericana AT&T, fue difundido libremente y a costos reducidos en universidades, centros de investigación y en el sector privado, de modo independiente a esa compañía pues enfrentaba un fallo antimonopólico que le impedía diversificarse comercialmente más allá de los teléfonos.¹⁸ Esa situación, junto a las características técnicas del sistema operativo, produjo que rápidamente cobrara popularidad entre los usuarios, puesto que además de la accesibilidad en su distribución, era compatible con una gran variedad de computadoras de bajo costo. Asimismo la posibilidad de editar, reprogramar y modificar las funciones desempeñadas por Unix habilitaron de esa manera a que múltiples usuarios anónimos escriban y re-escriban el software, apropiándose de él colectivamente. Sin embargo, una vez expirado el fallo antimonopólico en

¹⁶ Hurtado, Diego y Zubeldía, Lautaro. “El rol de las Universidades en el desarrollo económico”, *Política universitaria, Instituto de estudios y capacitación. Federación nacional de docentes universitarios*, número 4, octubre de 2017, páginas 24-33.

¹⁷ Coleman, *op.cit.*

¹⁸ Raymond, *op.cit.*

1982, AT&T reclamará su propiedad intelectual apelando a las leyes de copyright recientemente sancionadas, apropiándose de miles de contribuciones anónimas que desarrollaron y mejoraron el sistema operativo.¹⁹

En ese sentido, las posibilidades que tenían los usuarios de intervenir en Unix deriva de la noción misma de software y del modo en el que está escrito, tratándose de un plan de funcionamiento que habilita la relación de las computadoras con el ser humano, así como a conectarse entre sí. Un software o sistema operativo en efecto, es un conjunto de instrucciones realizadas mediante cálculos y algoritmos para “ordenarle” a la máquina cómo debe funcionar,²⁰ utilizando para su escritura lenguajes formales muy abstractos llamados código-fuente. Para que la máquina interprete las instrucciones, deben traducirse a código binario, mediante el empleo de programas especiales llamados compiladores, encargados de convertir un lenguaje a otro y de crear el “código-objeto”, los “binarios” o “ficheros ejecutables”: las instrucciones efectuadas desde la programación, traducidas y luego grabadas en el programa.

Hasta la década de 1970, los códigos fuente de los distintos software eran abiertos, es decir, presentados en el mismo lenguaje de programación en el que fueron escritos, posibilitando su re-escritura y modificación, pues como señala Richard Stallman:

No llamábamos “software libre” a nuestro software porque ese término todavía no existía, pero era exactamente eso. Cuando alguien de otra universidad o de una empresa quería adaptar un programa para utilizarlo, se lo permitíamos de buen grado. Si se veía a alguien usar un programa que era desconocido e interesante, siempre se le podía pedir que nos mostrara el código fuente para poder leerlo, modificarlo o tomar partes del mismo para hacer otro programa.²¹

La privatización segmentada de los elementos constituyentes del software significó que el conjunto de instrucciones recayera sobre la égida de las leyes de propiedad intelectual, expresándose en el modo en que su código fuente se presenta a los y las usuarias. El lenguaje binario, muy difícil de descifrar e interpretar bajo el ojo humano, será la forma en la que aparecerá el código, acompañando por un proceso análogo aplicado a las herramientas de edición y programación como los compiladores o editores, impidiendo de ese modo no sólo su libre distribución, sino también la posibilidad de intervenirlo.²²

Estas transformaciones tendrán consecuencias en la configuración de las comunidades de hackers, particularmente en sus formas de trabajo, escindiendo la unidad establecida entre las categorías de productores/usuarios, despojándolos no sólo de las herramientas necesarias para su programación, sino también de la posibilidad de imponer su propio ritmo de trabajo y de la libertad para crear las versiones de software deseadas. Las compañías de software en ascenso se encargarán de relegarlos al rol de empleados o

¹⁹ Söderberg, *op. cit.*

²⁰ Vidal, *op.cit*

²¹ Richard Stallman, “El proyecto GNU”, extraído de <https://www.gnu.org/gnu/thegnuproject.html> [Consulta: 24/5/2019]

²² Söderberg, *op. cit.*

consumidores, tal como sucedió con el caso Unix: “El intento por parte de AT&T para privatizar UNIX califica como uno de los cierres de código más conocidos que saludaban el amanecer de la era de la información”,²³ provocando el ocaso de una etapa signada por la libertad de editar, re-programar y difundir los aportes realizados individual o colectivamente.

No obstante, también se abrirá un nuevo capítulo en la historia del hacking, a partir de la emergencia de nuevas generaciones que se organizarán para defender la transparencia del código fuente, entendida como parte fundante de su libertad de expresión,²⁴ manteniendo una forma de trabajo cooperativa, sin la excluyente persecución de fines de lucro, regida por una serie de principios y una ética cimentada por el “último hacker auténtico”, Richard Stallman.

Richard Stallman y el proyecto GNU

Como empleado del Laboratorio de Inteligencia Artificial del MIT, Stallman experimentó en primera persona las transformaciones que afrontaba el sistema científico norteamericano a inicios de los ‘80, particularmente el reclutamiento de investigadores por compañías informáticas en ascenso y la rúbrica obligatoria de acuerdos que prohibían el acceso al código fuente y la divulgación de los software allí programados y/o utilizados.²⁵ Su renuncia en 1984 significó el comienzo de un proyecto que reivindicaría la libertad para compartir y modificar código fuente, mediante la construcción de una réplica del sistema Unix pero completamente libre. GNU fue el nombre con el que se conoció el proyecto, acrónimo recursivo que significa “GNU’s Not Unix” o en español “GNU no es Unix”, y que también designa al sistema operativo desarrollado por Stallman y sus colaboradores. El peligro de reclamos por la titularidad de la obra, la imposición de sus propios términos de distribución, o incluso su transformación en software privativo, provocó que Stallman se alejara definitivamente del MIT, aunque se le permitió el uso de los laboratorios para comenzar el proyecto.

Junto al proyecto GNU, la creación de la *Free Software Foundation* (Fundación de software libre, FSF en inglés) en 1985 tomó como objeto recaudar fondos para solventar el desarrollo de la réplica de Unix, principalmente mediante la venta de copias de ese y otros servicios relacionados, como CD-ROM con código fuente, CD-ROM con binarios, manuales impresos, siempre bajo la libertad de redistribución y modificación.

El proyecto GNU fue de ese modo cobrando forma, desarrollándose en principio un compilador, un editor y un depurador completamente libres, tarea reseñada por Vidal de la siguiente manera “Stallman tuvo que empezar casi desde cero (...) pues no existía la universalizada red Internet tal y como hoy la conocemos;

²³ Ídem, p. 30.

²⁴ Coleman, *op. cit.*

²⁵ Richard Stallman, “El proyecto GNU”, extraído de <https://www.gnu.org/gnu/thegnuproject.html> [Consulta: 24/5/2019]

tampoco existía una comunidad de desarrolladores lo suficientemente grande y ni siquiera se disponía de un compilador libre para empezar el trabajo.”²⁶

No obstante, la legislación de propiedad intelectual que comenzaban a aplicarse fue interpretada como una amenaza para la consecución de los objetivos propuestos por el proyecto GNU, redactando como contraparte un manifiesto que respondía contundentemente al peligro de ser privatizado. El Manifiesto GNU fue tanto una declaración de principios ²⁷ como un pedido de apoyo a la comunidad de hackers internacional para la realización del proyecto, retomado más tarde en la redacción de la *General Public Licence* (Licencia Pública General, en inglés GPL), licencia creada con el objeto de generar un “paraguas legal” que protegiera el carácter libre del software, garantizando sus cuatro principios básicos o libertades inalienables:

- La libertad de ejecutar el programa como se desee, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que el usuario o usuaria quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a otros (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones.²⁸

En ese marco, la creación del concepto de copyleft constituyó el último escalón requerido para garantizar el acceso completamente libre al software y su código, el que sirviéndose de los derechos de autor estipulados en el copyright, invirtió su sentido mediante un ingenioso juego de palabras. En efecto, bajo la licencia GPL y en particular su inscripción dentro del copyright, quien desarrolla un software de modo individual o colectivo permite la ejecución, copia, re-escritura y redistribución de sus versiones modificadas pero, en tanto propietario de sus derechos, impide que se le añadan restricciones en la distribución. Es decir, el propietario, en lugar de otorgar el derecho de restringir las copias, concede a los usuarios la posibilidad de copiar y compartir programas, restringiendo únicamente la facultad de privatizar los cambios introducidos.

La relevancia del copyleft así, reside en la protección del uso del software y no su propiedad, puesto que bajo su tutela el autor se reserva los derechos de libre utilización de la obra, poniendo como única condición la imposibilidad de recortar o eliminar los derechos de libre uso. La imposición de nuevas condiciones que limiten su disponibilidad -por ejemplo, la distribución de código binario modificado sin posibilidad de acceder al fuente- quedan de ese modo vetadas, en cuyo caso se vulneraría la licencia y con ella, se dispararía el copyright original, quitando el derecho a servirse del software. Este punto resulta crucial, pues impide la utilización de cualquier componente de software libre por parte de software privativo, situación experimentada

²⁶ Vidal, *op. cit.*, p. 40

²⁷ Para observar el Manifiesto GNU en detalle, puede consultarse la página oficial de GNU: <https://www.gnu.org/gnu/manifiesto.es.html>

²⁸ Extraído de <http://www.gnu.org/licenses/gpl.html> [Consulta: 14/03/2019]

cuando por ejemplo, el software es reprogramado mediante la adición de nuevos componentes y luego se prohíbe su distribución, tal como ocurrió con Unix.

La constitución del proyecto GNU quedó de ese modo conformada por tres componentes que excedían a lo estrictamente técnico, mencionado en principio los desarrollos propios de la programación y creación colectiva de los elementos del software,²⁹ la autogestión financiera establecida con la FSF, y finalmente el constructo político-legal que supone la creación y aplicación de la GPL. Estos desarrollos técnicos y políticos sentaron las bases para la producción de un software completamente libre, entendiendo “libre” no como sinónimo de gratis (en inglés *free* significa tanto libre como gratis), sino como el que otorga la libertad para disponer el código a quien lo desee, utilizándolo del modo que lo desee. Como sostiene Stallman, “el ‘software libre’ es una cuestión de libertad, no de precio”,³⁰ y aunque su filosofía rechaza una práctica empresarial específica y ampliamente difundida con el software privativo, no se opone a su comercialización: sin ésta la FSF no hubiese recabado los fondos necesarios para el desarrollo, distribución o copia de los componentes de GNU.

Aunque a lo largo de media década de trabajo los avances producidos en el marco del proyecto GNU no bastaron para completar el diseño y ejecución de la totalidad de componentes requeridos para un sistema operativo completamente nuevo, sentaron las bases para que el proyecto finalmente se concrete durante los años ‘90.

Linus Torvalds y GNU/Linux

El lento pero constante avance de la extensión del uso de microcomputadoras más allá de ámbitos académicos a principios y mediados de la década de 1990, junto a una mayor conexión a las distintas redes tendidas a lo largo del territorio norteamericano y europeo, habilitaron la construcción de una red de usuarios y hackers que oxigenaría un contexto en el que “... todos creían que la era del tecno-heroísmo se había acabado, que la industria del software y la naciente Internet serían dominadas por colosos como Microsoft...”.³¹ (Raymond, 2004: 38). Producto de una tendencia que venía expresándose en la década anterior, con la comercialización de microcomputadoras de escritorio, la extensión de redes abiertas al público general, así como el tendido de “pasarelas” entre redes como ARPA y Usenet (la red establecida entre computadoras con sistema Unix), sentaron las bases para el desarrollo de la World Wide Web y permitieron la emergencia de nuevas generaciones de usuarias y usuarios durante esos años.³²

²⁹ El sistema GNU incluye programas que no son software de GNU, desarrollados por otras personas y otros proyectos para sus propios fines, pero que al ser software libre pueden utilizarse: el editor de textos (TeX) escrito por Donald Knuth, el sistema de ventanas (X Window System) escrito por Bob Scheifler, un compilador (GCC) y un editor (EMACS) desarrollados por Stallman.

³⁰ Richard Stallman, “¿Qué es el software libre?”, en <http://www.gnu.org/licenses/gpl.html> [Consulta: 14/03/2019].

³¹ Raymond, *op. cit.*, p. 38.

³² Flichy, *op. cit.*

Aunque las motivaciones de los jóvenes hackers ahora se asociaban por lo general a desarrollos estrictamente técnicos y no tanto a cuestiones políticas o éticas como en el proyecto GNU, el programar “por placer” o “*just for fun*” (“simplemente por diversión”) continuaba siendo uno de los incentivos principales de su actividad. En ese sentido, los desarrollos iniciados por un joven estudiante finlandés a partir de 1991 marcarían la tónica de un nuevo método de trabajo dentro de los círculos de programadores y hackers, en apariencia, carentes de los atributos que definían a aquella cultura de “hackers auténticos”, pero con una forma de organización hasta entonces inédita.³³

Por ese entonces, Linus Torvalds era un joven estudiante de programación que, ante la imposibilidad económica de acceder a las caras versiones de Unix comerciales, decidió construir una versión propia mediante las herramientas desarrolladas por la FSF, pero independientemente de ella. Con ese propósito y sirviéndose del sistema operativo Mimix, compartió su código fuente en un *newsgroup*³⁴ y, solicitando la colaboración de otros programadores, obtuvo rápidamente la repuesta de miles de usuarios y usuarias dispuestas a contribuir con su iniciativa: el desarrollo un Unix totalmente equipado con fuentes gratuitas y de distribución libre. La contribución de Torvalds residió concretamente en el diseño y desarrollo del denominado kernel o núcleo del sistema operativo, el elemento que contiene las instrucciones más importantes del sistema y que garantiza la comunicación segura entre el software y el hardware, el único componente faltante dentro del proyecto GNU, y por ende, el más complejo de realizar.

Aunque dicho proyecto fue percibido por Torvalds como un hobby, su decisión de inscribirlo dentro de la GPL resultó clave para la historia de la programación hacia fines del siglo XX, posibilitando la emergencia de una nueva forma de organización entre las comunidades de hackers. La idea de “colgar” el código y solicitar colaboración en los canales de comunicación mediada por computadora utilizados en ese momento, permitió que la construcción del kernel se realizara entre miles de desconocidos conectados únicamente por la red, sin estándares rígidos y sin verticalismo. La estrategia seguida luego de aquel “llamamiento” consistió en “(...) hacer públicas las nuevas versiones todas las semanas y a los pocos días recoger las respuestas de cientos de usuarios, creando una suerte de selección darwiniana acelerada sobre las mutaciones introducidas por programadores.”³⁵ El proyecto tendido alrededor de Linux –nombre que se le dio al núcleo- mostró así que el trabajo hacker podía organizarse sin una única filiación institucional y sin ceñirse a las presiones y condiciones del mercado, realizando desde el ámbito doméstico las modificaciones o arreglos pertinentes, aportando documentación, o simplemente expresando su opinión sobre la calidad del kernel.³⁶

³³ Raymond, *op. cit.*

³⁴ Los *Newsgroups* o grupos de noticias posibilitaban, en el marco de la red Usenet, la lectura y el envío de mensajes a distintas grupales organizadas jerárquicamente según los tópicos seguidos por los usuarios.

³⁵ Raymond, *op. cit.*, p. 40.

³⁶ Moineau y Papathéodorou, *op. cit.*

Esa organización novedosa fue caracterizada rápidamente como modelo “bazar” en contraposición al “catedral”, el modelo regente en las grandes compañías de software donde la diagramación de tareas se establece vertical y centralizadamente, sin aceptar contribuciones externas al equipo que lleva adelante el trabajo. El modelo bazar por el contrario, se distingue por estar abierto a la cooperación de los y las interesadas, atrayendo por lo general a grandes cantidades de participantes que le otorgan dinamismo, basándose en la continua retroalimentación entre las y los programadores/usuarios y las características del sistema operativo. La velocidad con que se realizan las actualizaciones es otro de los aspectos que lo diferencia del modelo catedral, derivado de la restitución de la unidad establecida entre programadores/usuarios, y cuya escisión en los modelos de producción de software privativo redundaba en actualizaciones sometidas a largos períodos de pruebas que aseguren la eliminación de errores antes de enviarlo al mercado.³⁷

Dicha restitución de la unidad entre productores/usuarios, junto a la lógica cooperativa del modelo “bazar” fueron también interpretadas desde un ángulo político, pues a pesar de guiarse por motivaciones técnicas y no tanto por preceptos éticos como los y las hackers de la década anterior “(...) the passionate commitment to hacking and especially the ethics of access enshrined in free software licensing, express as well as celebrate unalienated, autonomous labor, which also broadcasts a powerful political message. [...el compromiso apasionado con el hacking y, especialmente, la ética del acceso consagrado en las licencias de software libre, expresan y celebran la mano de obra autónoma e independiente, que también transmite un poderoso mensaje político.]”³⁸ En ese sentido, la organización en red tendida en torno a la producción de software libre es interpretada como alternativa fuente al trabajo alienado, pues dispone tanto de los medios de producción (los editores, compiladores y demás herramientas utilizadas para programar), como del producto del trabajo hacker,³⁹ desarrollándose en un marco donde “...las empresas y las instituciones de gobierno han perdido su monopolio sobre la investigación y el desarrollo.”⁴⁰

Finalmente, el establecimiento de una organización del trabajo novedosa y dinámica, permitió que en 1992 el kernel Linux se completara y, en combinación con los componentes del sistema GNU, diera lugar a un software completamente libre denominado GNU/Linux. Aunque inadvertido en ese momento por fuera de comunidades de hackers o de entornos de aficionados a la programación, hacia finales de la década la explosión en torno a Silicon Valley provocaría que las miradas de las ascendentes corporaciones informáticas se dirijan hacia él, primero para combatirlo, y luego para incorporarlo al funcionamiento de sus empresas, invirtiendo en su desarrollo.

³⁷ Raymond, *op. cit.*; Söderberg, *op. cit.*; Vidal, *op. cit.*

³⁸ Coleman, *op. cit.*, 28.

³⁹ Ídem.

⁴⁰ Soderberg, *op. cit.*, 11.

Sin embargo, y a pesar de que iniciativas como el proyecto GNU o GNU/Linux no se opusieran a la comercialización del software o sus componentes, los fundamentos éticos que los sustentaban fueron juzgados como inapropiados desde el punto de vista comercial por las compañías dispuestas a incursionar en ese mercado. Reunidas en 1998 en la cumbre conocida como *Freeware Summit*, la intencionada omisión del mentor y referente del movimiento respondió a la necesidad empresarial de “despegarse” de la imagen izquierdista de Stallman, y de presentar a las iniciativas de software libre como comercializables y amigables hacia los inversionistas.⁴¹ La estrategia de marketing desplegada entonces fue la permutación del término “libre” (*free*) por “abierto” (*open*), realizando un movimiento retórico que también modificaba un aspecto capital de la GPL: aunque la licencia open source requería que el software pueda distribuirse libremente, garantizando la transparencia de su código fuente y el derecho a crear versiones modificadas, no exigía que se incluyan en ella los derivados del código original. Es decir, el software bajo licencia open source habilitaba a que las transformaciones realizadas por usuarios puedan publicarse posteriormente bajo licencia copyright, ofreciendo una puerta trasera para que las empresas se apropiaran del código y de las contribuciones efectuadas por los y las usuarias, licencias que “...pueden ser descritas como un principio de organización para la sistematización de la ‘acumulación originaria’, es decir el robo de trabajo social que tiene lugar en las comunidades en desarrollo y en los bienes comunes”.⁴²

El cambio de “libre” a “abierto” tuvo éxito, y grandes compañías informáticas aplicaron la licencia open source a sus productos, o bien adoptaron modelos de software derivados de GNU/Linux en sus infraestructuras y sistemas, una vez iniciada la década del 2000.⁴³ Este acercamiento significó que los sistemas libres o abiertos adquirieran credibilidad más allá de ámbitos del *underground* informático o de la denominada cultura *geek*, respondiendo a las necesidades de un mercado cada vez más orientado a los servicios y que exigía una reestructuración en su forma de organizarse. Sin embargo, y a pesar del avance privatizador sobre las iniciativas de software libre, compañías como Linux lograron asentarse ganado terreno frente a los modelos de empresa de software privativo, emergiendo una multiplicidad de iniciativas que, inscritas bajo sus preceptos, recorren proyectos comerciales enormes como Debian, pasando por Wikipedia, hasta su utilización exclusiva desde los estados de América Latina, por ejemplo, mediante su adopción en programas educativos o en los servicios informáticos utilizados por sus administraciones públicas, aspectos señalados como centrales en la denominada soberanía informática de dichos países.⁴⁴

⁴¹ Coleman, *op. cit.*

⁴² Söderberg, *op. cit.*, p. 53.

⁴³ Un caso paradigmático fue la liberación on-line del código del navegador Netscape hacia finales de los años '90, motivada por los problemas económicos que atravesaba dicha compañía, llamando la atención de diversas empresas informáticas que rápidamente vieron en esa estrategia una oportunidad de negocios, reuniéndose más tarde en la *Freeware Summit*. IBM, Dell, Oracle o Sun, comenzaron desde entonces a lanzar productos bajo licencias open source, así como a adoptar a GNU/Linux para el desarrollo de sus servicios informáticos, invirtiendo en el desarrollo y distribución de ese sistema operativo.

⁴⁴ Stallman, Richard “El software libre es ahora aún más importante” Extraído de <https://www.gnu.org/philosophy/free-software-even-more-important.es.html> [Consulta: 10/07/9]

III. Conclusiones

La recapitulación realizada en torno a los avances y desarrollos en las tecnologías de información y comunicación con sus correspondientes formas de organización del trabajo en la programación y el hacking a partir de la década del '50, la paulatina privatización y cerramiento del código fuente hacia finales de los '70 y principios de los '80, así como la emergencia del software libre como movimiento capaz de afrontar la falta de acceso a la información vertida en el mencionado código, pretendieron exponer el carácter histórico del desarrollo de dichas tecnologías y servicios informáticos, en consonancia con las transformaciones políticas y económicas correspondientes al periodo analizado en Occidente.

Los cambios experimentados en las formas de producción científica, puntualmente su financiamiento y organización, establecidos en el marco del avance neoliberal y de la reestructuración de la organización del trabajo, como corolario del fin de la producción en masa, el pleno empleo y la cadena de montaje fordista, permiten interpretar la creciente informatización del proceso productivo como una estrategia del capital para afrontar la crisis económica de los años '70 y desarticular la conflictividad social ascendente experimentada en ese contexto. El viraje desde una producción mayoritariamente industrial hacia una orientada a los servicios, implicó también una mayor flexibilidad en los empleos y en la organización de la producción, presentada bajo el postfordismo como un servicio supeditado a la demanda, crecientemente informatizado y tendiente a convertir los avances tecnológicos en medios de producción y en capital fijo.⁴⁵

En este marco, la constitución de un movimiento en oposición a las transformaciones impuestas en los modos de gestionar y organizar el trabajo científico, procurando mantener la libertad de intervenir y decidir sobre qué usos y prácticas desplegar en la interacción con las computadoras, se presentó como una alternativa que por primera vez puso en discusión el carácter político de los servicios y productos informáticos, en tanto clave de acceso a la libertad de información. Aunque el movimiento por el software libre haya mutado a lo largo de su trayectoria, no sin presentar cambios de rumbo y heterogeneidades en su interior, consideramos que su relevancia estriba en constituir un modelo ético de vinculación alternativa con la tecnología, por fuera de los intereses comerciales de las grandes compañías monopólicas de media y comunicación, destacándose por una forma de trabajo original, anclado en la inventiva, la cooperación y el programar “solo por diversión”.

Por último, recuperar su ejemplo resulta útil para iluminar fenómenos actuales como la creciente extracción de datos utilizados como fuente de ganancia, cuyos modelos de “caja negra” aplicados a sistemas operativos, plataformas y redes sociales, utilizan algoritmos para monitorear las acciones y movimientos de los usuarios y usuarias, con su concomitante penetración en ámbitos cotidianos y en las formas de sociabilidad. La recuperación de experiencias como la del movimiento software libre, nos invita a reflexionar sobre la

⁴⁵ Terranova, Tiziana “*Red Stack Attack! Algoritmos, capital y la automatización de lo común*”, en Armen Avanesian y Mauro Reis (comps.) *Aceleracionismo. Estrategias para una transición hacia el postcapitalismo*, Buenos Aires, Caja Negra, 2017.

posibilidad de construir alternativas frente a los modos en que se presentan esas plataformas, aplicaciones y sistemas operativos cerrados, habilitándonos a cuestionar críticamente la naturalización y la aparente neutralidad de su funcionamiento opaco.

Bibliografía

Coleman, Gabriella. *Coding Freedom. The ethics and aesthetics of hacking*, Princeton, Princeton University Press, 2013.

Flichy, Patrice. *El imaginario de Internet*, Madrid, Tecnos, 2003.

Hardt, Michael y Negri, Antonio. *Imperio*. Cambridge, Harvard University Press, 2000.

Hurtado, Diego y Zubeldía, Lautaro. “El rol de las Universidades en el desarrollo económico”, *Política universitaria, Instituto de estudios y capacitación*. Federación nacional de docentes universitarios, número 4, octubre de 2017, pp. 24-33.

Míguez, Pablo. “Del *General Intellect* a las tesis del ‘capitalismo cognitivo’: aportes para el estudio del capitalismo del siglo XXI”. *Bajo el Volcán*, año 13, número 21, septiembre 2013-febrero 2014, pp. 27-57.

Moineau, Laurent y Papatheodorou, Aris. “Cooperación y producción inmaterial en el software libre. Elementos para una lectura política del fenómeno GNU/Linux”, extraído de <https://sindominio.net/biblioweb/telematica/cooperacion.pdf> [consulta: 27/06/2019]

Raymond, Eric S. “Breve historia de la cultura hacker”, en Carlos Gradin (compilador) *Internet, hackers y software libre*, Buenos Aires, Editora Fantasma, 2004, pp. 27-44.

Söderberg, Johan (2019) *Hackeando al capitalismo: el movimiento de software libre y de código abierto*. Traducción y edición por el Partido pirata. Extraído de https://utopia.partidopirata.com.ar/hackeando_al_capitalismo.html [Consulta: 27/06/2019]

Srnicek, Nick. *Capitalismo de plataformas*, Buenos Aires, Caja Negra, 2018.

Stallman, Richard. “El proyecto GNU”, extraído de <https://www.gnu.org/gnu/thegnuproject.html> [Consulta: 24/5/2019]

Stallman, Richard. “¿Qué es el software libre?”, extraído de <http://www.gnu.org/licenses/gpl.html> [Consulta: 14/03/2019].

Terranova, Tiziana. “*Red Stack Attack!* Algoritmos, capital y la automatización de lo común”, en Armen Avanessian y Mauro Reis (comps.) *Aceleracionismo. Estrategias para una transición hacia el postcapitalismo*, Buenos Aires, Caja Negra, 2017.

Vidal, Miquel. “Cooperación sin mando: una introducción al software libre”, en Carlos Gradin (compilador) *Internet, hackers y software libre*, Buenos Aires, Editora Fantasma, 2004, pp. 45-70.