

DOI: 10.35643/Info.27.1.9

Artículo original

Software libre, ciencia libre: explorando algunas enseñanzas que el software libre puede aportar a la ciencia abierta

Free Software, Free Science: exploring some lessons that Free Software can bring to Open Science

Software Livre, Ciência Livre: explorando algumas lições que o Software Livre pode trazer para a Ciência Aberta

Daniel Viñar Ulriksen¹ ORCID: [0000-0001-6896-3290](https://orcid.org/0000-0001-6896-3290)

¹ Coordinador de la Red de Unidades Informáticas de la Universidad de la República en el Interior.
Correo electrónico: dvinar@cci.edu.uy, daniel@uruguayos.fr

Resumen

En primer lugar, en este trabajo analizo las diferencias y matices en la filosofía que hay tras las palabras *libre* y *abierto*, tal como se cristalizaron en el campo del *software*. Procuró también, a través de su historia y el análisis de su ecosistema, esbozar la necesaria deconstrucción del paradigma de la propiedad intelectual.

En una segunda parte, un breve relato de la historia de la informática, el *software* libre y la Internet permite identificar puntos comunes en los procesos de la ciencia y del desarrollo de *software*, y algunas características o realizaciones del mundo en la era digital que pueden aportar a la ciencia abierta.

En la tercera parte especulo brevemente, a partir de la trayectoria que ha sido hasta ahora la del *software* libre, sobre los alcances posibles para una ciencia abierta, antes de concluir volviendo al punto central de la necesidad de una reforma radical de la regulación de la economía de lo inmaterial.

Palabras clave: CIENCIA ABIERTA; *SOFTWARE* LIBRE; *OPEN SOURCE*; *KNOWLEDGE COMMONS*; PATENTABILIDAD DEL *SOFTWARE*.

Abstract

I first analyse the differences and nuances in the philosophy behind the words *free* and *open*, as crystallised in the field of software. I also try, through its history and the analysis of its ecosystem, to outline the necessary deconstruction of the Intellectual Property paradigm.

In the second part, a brief account of the history of computer science, Free

Software and the Internet, allows us to identify commonalities in the processes of Science and Software development, and some characteristics or realisations of the world in the digital era that can contribute to Open Science.

In the third part, I briefly speculate, based on the trajectory of Free Software so far, on the possible scope for Open Science, before concluding by returning to the central point of the need for a radical reform of the regulation of the economy of the immaterial.

Keywords: OPEN SCIENCE; OPEN SOURCE; FREE SOFTWARE; KNOWLEDGE COMMONS; SOFTWARE PATENTABILITY

Resumo

Analiso primeiro as diferenças e nuances da filosofia por trás das palavras *livre* e *aberto*, tal como cristalizada no campo do software. Tento também, através de sua história e da análise de seu ecossistema, delinear a necessária desconstrução do paradigma da Propriedade Intelectual.

Na segunda parte, um breve relato da história da informática, do Software Livre e da Internet, nos permite identificar pontos comuns nos processos de Ciência e desenvolvimento de Software, e algumas características ou realizações do mundo na era digital que podem contribuir para a Ciência Aberta.

Na terceira parte, eu especulo brevemente, com base na trajetória do Software Livre até agora, sobre o possível alcance da Ciência Aberta, antes de concluir voltando ao ponto central da necessidade de uma reforma radical da regulação da economia do imaterial.

Palavras-chave: CIÊNCIA ABERTA; CÓDIGO ABERTO; SOFTWARE LIVRE; CONHECIMENTO COMUM; PATENTEABILIDADE DE SOFTWARE

Fecha de recibido: 11/10/2021

Fecha de aceptado: 21/01/2022

1. Elementos de metodología y planteamientos

Este artículo no busca ser una investigación metodológica, sino un ensayo que procura identificar en la madurez el *software* libre (SL) algunas enseñanzas para construir más ciencia abierta (CA). ¿Podemos, a partir de la filosofía y de las prácticas del SL, identificar factores claves para el buen desarrollo de la CA? ¿Son de orden técnico, institucional, regulatorio? ¿Qué vínculos se tejen entre CA

y SL?

Como académico he sido episódicamente docente; tengo esencialmente una formación y experiencia como consultor e ingeniero, generalista y en mis áreas de competencia, que son las ciencias básicas, la informática y las telecomunicaciones. Tanto en lo profesional como lo personal, mis prácticas se articulan fuertemente con una militancia en pro del *software*, el conocimiento y la cultura libres.

A partir de mi trayectoria y algo de bibliografía, exploro paralelos y diferencias entre SL y CA, en dos partes que corresponden a dos facetas esenciales del SL: por un lado, la concepción filosófica y el marco regulatorio de la economía de lo inmaterial y, por otro, el desarrollo técnico del mundo digital tal cual lo venimos descubriendo, que se basa en una tecnología que tiene características únicas y nuevas. Especulo en torno a algunos ejes posibles de desarrollo de la CA, a partir de elementos tecnológicos y proyectando a la ciencia las prácticas más recientes en el área del *software*.

Finalmente formulo elementos para una propuesta que considero clave para un buen desarrollo del SL y de la CA: un cuestionamiento y rediseño en profundidad de la propiedad intelectual, en particular del sistema de patentes, en pro de los comunes, los cuales asumo como una reflexión desde una posición involucrada y comprometida.

2. ¿Libre o abierto? ¿Propiedad intelectual?

Sorteemos primero dos puntos de terminología, que creo conviene no eludir.

¿Libre o abierto? Hablamos de *ciencia abierta*, de *datos abiertos*, de *acceso abierto*, de *recursos educativos abiertos*, de *gobierno abierto*..., pero para el software tenemos dos denominaciones: *software libre* u *open source*. Y también decimos *cultura libre*, no *abierta*. Y no son estos conceptos aproximativos, más o menos sinónimos; tienen, al contrario, definiciones muy precisas. SL es aquel que cumple cuatro libertades, las de poder: 0) correr el software para cualquier propósito; 1) estudiar como funciona, para lo cual se requiere su código fuente^[1];

2) modificarlo para adecuarlo a otras necesidades, y 3) distribuir versiones del software y sus derivados. Un SL puede además tener copyleft, exigencia para que cualquier software derivado cumpla con las mismas libertades que aquellos que modifica o integra, de manera de preservarlas en la evolución del software distribuido (Free Software Foundation, 1998). El movimiento de cultura libre proyecta al campo artístico, literario, académico, o a cualquier disciplina que da lugar a una creación inmaterial, las cuatro libertades del SL y el copyleft (también denominado compartir igual) (FreedomDefined, 2006). Creative Commons define un abanico de siete licencias, herramienta base del acceso abierto (open access), todas más permisivas que el tradicional «todos los derechos reservados» del copyright —autorizando al menos la circulación sin fines de lucro de copias de la obra—, pero de las cuales solo tres califican como cultura libre (Creative Commons, 2016).

Legítimos voceros de los movimientos del SL y del Open Source (OS) acuerdan que los dos trabajan sobre un mismo objeto (o casi), comparten su conocimiento y producen juntos un conjunto de software coherente e interconectado, se habla del FOSS o FLOSS (free/libre and open source software). Pero también coinciden en que persiguen objetivos y tienen valores diferentes. Richard Stallman, fundador del movimiento de SL y de la Free Software Foundation (FSF)[\[2\]](#), afirma que «ya sea tú dominas un programa informático, ya sea el programa te domina». Considera que el movimiento del SL defiende la libertad, mientras que el del OS busca la comodidad (Stallman, 2011, 2015b). Por su lado, en el OS, si bien se suele asumir que sí, importa la comodidad: se usa y se desarrolla FLOSS porque es más fácil, más barato y funciona bien; también (y por eso) se promueve una adopción de FLOSS menos rigurosa: no juzgar a priori sobre la malignidad de un software privativo, dejarle esa prerrogativa al usuario final.

John Maddog Hall, otra figura importante del movimiento de SL, propone que, mientras que el movimiento del OS se basa en un pacto entre desarrolladores, el del SL también involucra al usuario final, sea o no sea desarrolladorx[\[3\]](#).

Nos quedaremos con esta última caracterización para distinguir *libre* y *abierto*. Ciencia abierta sería, entonces, aquella que procura establecer un pacto entre investigadores y académicxs, que ponga a todxs de igual a igual en el acceso al

conocimiento, la información y los datos del acervo científico de la humanidad. Ciencia libre sería aquella que también considera esencial incluir al habitante de a pie, a todas las culturas de igual a igual (y no solo según su nivel de inversión en I+D), en todos los quehaceres de la ciencia, puesto que sus aportes son necesarios y pueden ser poderosos. Así sería, en todo caso, si los paralelos entre ciencia y *software* que pretendemos explorar acá resultan ser pertinentes.

Por eso la guiñada a RMS[4] con *ciencia libre* en el título de este artículo. Pero que sea tomado como un eje de reflexión, de ninguna manera como una guerra de nomenclaturas o capillas. El movimiento de CA ya tiene nombre y, aunque solo haya consenso o recursos para promoverlo entre académicxs, no deja de ser pertinente y urgente.

Otro concepto que Stallman deconstruye al elaborar el marco teórico y legal del SL es el de *propiedad intelectual* (PI). Cuando en el siglo XIX las naciones procuraron armonizar su normativa para los bienes inmateriales que permitieron la revolución industrial, vieron la luz dos acuerdos muy diferentes: el Convenio de París para la Protección de la Propiedad Industrial en 1883, que regula marcas y patentes, y el Convenio de Berna para la Protección de las Obras Literarias y Artísticas en 1886, que regula el derecho de autor. Ninguno de los dos documentos contiene la expresión *propiedad intelectual*, y las motivaciones económicas y morales de cada uno de los conceptos —derecho de autor, marcas y patentes— son muy diferentes. Medio siglo después, la Declaración Universal de Derechos Humanos solo consagra como tal el derecho de autor, estableciendo un justo equilibrio entre el «derecho a tomar parte libremente en la vida cultural de la comunidad, a gozar de las artes y a participar en el progreso científico y [sus] beneficios», por un lado, y, por otro, el de «la protección de los intereses morales y materiales que le correspondan por razón de [sus] producciones científicas, literarias o artísticas» (Naciones Unidas, 1948).

El derecho de autor tiene su origen esencialmente en la invención de la imprenta, para regular el mercado de la edición, en particular a través de la relación autor-editor y los derechos de reproducción de la obra. Consagra al respecto un monopolio del autor por un tiempo determinado. Pero a estos derechos patrimoniales se agrega —al menos en la tradición latina (de *civil law* más que de

common law) y los acuerdos internacionales— un derecho moral, que es perpetuo, inalienable e imprescriptible. Michel Foucault afirma:

los textos, los libros, los discursos empezaron a tener realmente autores (otros que personajes míticos, que grandes figuras sacralizadas y sacralizantes) en la medida en que el autor podía ser castigado, es decir, en la medida en que el discurso podía ser transgresor (Foucault, 1970, p. 8).

Pese a muchos debates, o a raíz de estos, el derecho de autor es de naturaleza muy diferente que la propiedad de bienes tangibles (Moyses, 1997) y el Convenio de Berna solo habla de *protección* de la obra. Se protege la forma, no el fondo. Se contemplan excepciones por uso justo (*fair use*) y se penaliza esencialmente el plagio, permitiendo siempre la cita e incluso el uso integral de la obra con fines de parodia.

Las marcas constituyen entidades en el espacio simbólico de nuestra sociedad, que identifican todo tipo de productos o personerías, físicas o jurídicas, materiales o incorpóreos. Regularlas es importante como lo es armonizar la ortografía o la correspondencia de las denominaciones geográficas y la señalización vial. Se penaliza esencialmente la falsificación.

Las patentes, cuyo papel se despliega esencialmente a raíz de la revolución industrial, atribuyen a su titular un monopolio por un tiempo limitado contra la publicación de los detalles de su invento. El sistema de patentes pretende así lograr, en beneficio mutuo y del interés general, el encuentro entre el inventor y la inversión necesaria para la explotación de la patente. No todo es patentable: originalmente las patentes requerían un «*flash* de genio» y se otorgaban, por ende, a individuos. Solo a mediados del siglo XX, con los antibióticos, pasó a patentarse un «paso inventivo no obvio» y pudo hacerlo una corporación. Según el Convenio sobre la Patente Europea y muchas legislaciones nacionales, las patentes se otorgan en todos los campos de la tecnología susceptibles de aplicación industrial, pero están explícitamente excluidos los descubrimientos, la ciencia, la matemática, los juegos, las prácticas mentales y de negocios y los programas informáticos.

Por más que compartan tener por objeto bienes intangibles de la humanidad, agrupar estos tres conceptos[5] bajo el de PI plantea varios problemas. El

concepto de propiedad está fuertemente impregnado de la economía de los bienes tangibles. Al compartir la comida o el uso de la tierra, disminuimos el beneficio propio de esa propiedad, que en general solo cedemos a cambio de una contraparte de mismo valor monetario.

Los bienes intangibles tienen la particularidad de ser reproductibles al infinito y de no sufrir desgaste, a costo marginalmente nulo. Son abundantes, no escasos. Si compartes una idea, los demás también tendremos la totalidad de la idea, sin que esta disminuya en nada para ti, y juntos podremos mejorar ese bien común. Sin prejuicio económico o moral, es claro, entonces, que los términos del intercambio en la economía de lo inmaterial son muy diferentes de los de la economía de lo material. Y este universo se presta particularmente a la colaboración y la constitución de comunes.

A la inversa, la escuela de la PI asume que, en razón de esa abundancia, es conveniente regular para generar una escasez artificial y así crear valor y mecanismos de mercado capaces de atraer la inversión necesaria a esta economía del mundo virtual, y asegurar así la economía material que sustenta.

Si bien originalmente es la difusión del conocimiento y de la cultura lo que motiva una legislación en PI, a lo largo de este medio siglo de inicio de la era de la información, es esta última visión privatizadora, de cercamiento de los comunes (Boyle, 2003), la que ha dominado la acción del legislador, sin que exista evidencia de sus postulados, más bien lo contrario.

En la cultura y el entretenimiento, desde la fotocopia y el minicasete hasta la compartición de archivos en Internet, la rigidez de la industria promovió un prohibicionismo que hace, por ejemplo, que hoy en Uruguay un bibliotecario cometa un delito penal al prestar un libro o hacer una copia de un ejemplar único en mal estado (Díaz Charquero & Goñi Mazzitelli, 2021).

Incluso en el campo de las marcas, cuya legislación no plantea tales conflictos sobre un bien común y que es bastante funcional a la cognición humana contemporánea, no han faltado excesos de la *cortina de humo* de la PI, por ejemplo, cuando se pretende que usar una marca para criticarla portaría perjuicio a la PI de sus titulares.

Los sistemas de patentes han sido reformados a iniciativa de quienes quieren patentar y, por ende, más a su favor que considerando el interés general. Las oficinas de patentes funcionan sin verdadero contrapoder y obtienen sus recursos de onerosos procesos de registro de patentes, siendo así poco neutrales al evaluar la patentabilidad, generando intereses encontrados con los titulares de patentes. En Estados Unidos el campo de lo patentable puede extenderse por mecanismos de jurisprudencia[6] y la Oficina Europea de Patentes (EPO) homologa patentes con el solo argumento de haber sido otorgadas en Estados Unidos. Y así cualquier cosa se patenta. En disciplinas como el desarrollo de *software* —sobre todo FLOSS—, en las que la innovación procede por acumulación de pequeños pasos registrados en alguna línea de código más que por grandes inventos conceptuales, una patente trivial o vaga es una espada de Damocles que, amenazando un punto particular, puede derrumbar todo el edificio (Foundation for a Free Information Infrastructure [FFII], 2004). Tanto la oposición a una patente como defenderse frente a una demanda por infracción son procesos sumamente onerosos. Más que un instrumento de promoción de la invención, las patentes se han vuelto un arma disuasiva de defensa de los intereses, no de quienes inventan, sino de quienes capitalizan en PI. Guadamuz analiza:

La situación de las patentes en biotecnología (...) genera un entorno que vive en constante temor a los litigios. (...) Los investigadores que trabajan en el área de la genética han reducido considerablemente las prácticas normales de colaboración académica debido al temor a la violación de patentes.

Del mismo modo, las patentes genéticas demasiado amplias podrían (...) sofocar la competencia en la naciente industria biotecnológica. Los pequeños centros de investigación, las instituciones educativas y los investigadores individuales pueden tener dificultades para llevar a cabo investigaciones por miedo a verse envueltos en costosos litigios sobre patentes. Además, incluso si una patente biotecnológica se ha concedido erróneamente, las partes interesadas y los investigadores tendrían que implicarse en un largo procedimiento para anular la patente inválida, lo que ahogaría aún más la investigación (Guadamuz, 2005, p. 335).

Tal es la situación en las disciplinas que, como el *software*, tienen una fuerte interdependencia comunitaria e involucran procesos cumulativos en su innovación y progreso del conocimiento. Por donde sea que se lo mire, el paradigma de la PI, en primer lugar el de las patentes, está roto:

A diferencia de las tecnologías simples[7], como las químicas, para las que son indispensables, las patentes son intrínsecamente *desventajosas* para las tecnologías complejas. (...)

Es de dudar que el nivel de inversión en I+D en tecnologías complejas se reduzca significativamente si se suprime por completo el sistema de patentes (Kingston, 2001, p. 407 y 409).

E incluso en el caso de la química y la farmacéutica, Kapczynski (2016) hacía el siguiente anuncio premonitorio, que podemos meditar a la luz de la covid-19, y analizaba el interés y la voluntad de respuesta de los actores económicos en el marco actual de la PI:

Se considera inevitable una nueva pandemia de gripe, pero al igual que un gran terremoto, su momento es muy incierto. Un estudio (...) [preveía] un 100% de posibilidades de que se produjera una nueva pandemia de gripe en los próximos treinta años. Pero las patentes sólo duran veinte años, y el valor actual devaluado de ganancias a largo plazo descontado para las empresas disminuye rápidamente.

Más aún, en un escenario de verdadera pandemia, con millones de enfermos graves, los titulares de las patentes estarían bajo una presión política impensable para permitir voluntariamente el uso de la patente. Y en la alternativa, de acuerdo con el derecho internacional, los países podrían simplemente anular las patentes (Kapczynski, 2016, p. 1559).

Por ciertas características del territorio de su despliegue, el FLOSS ha podido sortear muchos escollos de la PI, en particular construyendo un marco de licencias permisivas que se apoyan en el derecho de autor vigente —las cuales, además, hacen todo lo posible por excluir las patentes—, generando un ecosistema «técnicamente viable, económicamente sustentable y socialmente justo»[8]. Pero su éxito no deja de ser parcial, por haber conquistado parte importante del mercado solo bajo forma de *software* cerrado o *software* como servicio, es decir, participando de engendrar los GAFAMN[9], gigantes de la economía virtual, que nos llegan asociados a la recolección y monetización de datos personales y a la vigilancia permanente, sector cuya capitalización ha sobrepasado la del petróleo y cuyos efectos colaterales son la posverdad de las *fake news* y el fin de la privacidad (Viñar Ulriksen, 2020).

Sin duda, del SL podemos sacar muchas enseñanzas para una ciencia libre o abierta —intentamos identificar algunas aquí—. Pero mientras no emprendamos una deconstrucción integral y en profundidad del paradigma de la PI, el

movimiento de CA quedará acotado a la precariedad y a una semiclandestinidad, a la sombra de una ciencia dominante sometida a los grandes intereses. No dejará de tener grandes logros y aciertos, como los ha tenido el SL. Los grandes intereses en PI no pretenden impedir que cualquiera investigue e innove (al contrario), pero buscan tener los cerrojos necesarios para controlar los réditos económicos y la competencia.

La Quadrature du Net (2012) formuló una propuesta de reformas del derecho de autor, a la que ha adherido a nivel global el movimiento de cultura libre y acceso abierto (Fundación Vía Libre, 2013), que apunta a equilibrar las prácticas y adecuarlas a la nueva realidad digital, en particular despenalizando la copia y circulación de obras sin fines comerciales y estableciendo una gestión por registro (no por omisión) de los derechos patrimoniales de autor.

Respecto a las patentes, Kingston y otrxs autorxs, si bien constatan la inadecuación profunda del sistema, más que cuestionarlo, proponen reformas para atacar sus disfunciones, como una revisión de patentes en modalidad *open source* mediante herramientas tecnológicas adecuadas o el establecimiento de un arbitraje técnico sistemático, en vez de un proceso judicial para la oposición o la infracción de patentes, y que luego la eventual apelación ante la Justicia esté a cargo de quien cuestione el arbitraje (Arup & Van Caenegem, 2009; Kingston, 2001, 2009).

Sin duda, tales propuestas pueden mejorar el funcionamiento del sistema, que hoy es particularmente desequilibrado, injusto, oneroso e ineficaz. Pero por estas mismas razones, quizás sea adecuada una propuesta más radical, que procuro esbozar en la conclusión.

3. Breve historia de la informática, el FLOSS y la Internet

Como muchas otras tecnologías, las de la información nacieron entre la industria, la academia e incluso los garajes de inventores *amateurs*. Pero la materia de las tecnologías de la información y la comunicación, si bien comprende el *hardware*, es decir las computadoras y otros artefactos electrónicos, que son objetos tecnológicos bastante tradicionales, también incluye, y en proporción creciente, el *software*, substrato puramente simbólico, informacional, muy similar al que trabaja la matemática y toda la parte teórica, no experimental, de la ciencia.

Miremos, no obstante, primero el ordenador en sí, que no deja de ser una tecnología muy reciente y particular, cuyo desarrollo conoció un ritmo de progreso nunca visto antes en la industria. Habitualmente, un proyecto tecno-industrial suele responder a un compromiso entre varios factores. Por ejemplo, construir un avión de pasajeros supersónico será muy caro y consumirá mucho combustible. Sin anticipar una crisis petrolera, como fue el caso del Concorde, puede ser una hazaña tecnológica, pero será un fracaso comercial. Con las tecnologías de circuitos integrados en semiconductores, todos los parámetros mejoran al mismo tiempo. Cuanta más densidad de transistores contienen los microchips, más rápido van y más barato cuestan relativamente. Durante más de tres décadas, el progreso exponencial de las computadoras resultó ajustarse a la ley de Moore[10], duplicando su capacidad cada dos años.

Esto plantea capacidades de cálculo y almacenamiento —recursos primarios del *software*— hasta hace poco inimaginables, lo que también abre el paso a la emergente ciencia de datos y una aceleración de la innovación de la misma índole en disciplinas como las nanotecnologías, las biotecnologías o la genómica.

Por su lado, el *software*, componente esencial totalmente inmaterial de la informática, nos acerca en su esencia a las prácticas de la ciencia. Sin pretender situarnos en una formalización rigurosa, es legítimo hacer una analogía entre un teorema matemático y su demostración, por un lado, con un programa de computación y su código fuente, por otro[11].

¿Cómo sería entonces la *matemática privativa*, si existiera? Las *empresas de matemáticas* serían dueñas de sus teorías, las demostraciones serían secretas, podrían legítimamente cobrar a cualquiera que las utilice y cada vez que las utilice. ¿Con qué sustento los físicos o la industria podrían utilizar tales teoremas para su investigación o sus desarrollos? ¿Qué perito podría atribuir responsabilidades frente a una falla que pueda ser debida a un error del teorema? En resumen: sin revisión entre pares, independientes y de igual a igual, ¿como podemos afirmar la exactitud de la demostración del teorema?

Desde siempre, la metodología académica, de la ciencia positiva, ha sido proceder por hipótesis y duda, cuestionando y verificando lo afirmado por los hechos, permitiendo a otrxs reproducir la investigación, para confirmar o refutar los

resultados. La prueba es un mero consenso temporal en este proceso iterativo de verificación. Y, desde los dos teoremas de incompletitud de Gödel (Girard et al., 1997), sabemos que, en un universo que contiene al menos la aritmética, ninguna verdad es absoluta, sino solo relativa a cierto marco axiomático, el cual nunca podrá ser exhaustivo.

Si seguimos la analogía, parece natural, entonces, entre desarrolladores de *software* someter su código fuente a la verificación por pares, para asegurar su exactitud, seguridad y calidad.

Pero los fundamentos del SL —que, por un lado, privan al desarrollador del rédito que pueda obtener de un código fuente secreto, al mismo tiempo que aseguran este proceso de revisión independiente entre pares y abren la vía a la constitución de un acervo de *software* incremental y colaborativo— no nacieron de una larga tradición como la académica, sino de prácticas innovadoras y conflictos entre diferentes intereses.

En los años 60, cuando surgieron los primeros ordenadores con *software* modificable, IBM dominaba ampliamente el mercado. Se opuso por más de una década a la patentabilidad del *software*, a fin de controlar el mercado del servicio de desarrollo de *software* mediante su capital de patentes de *hardware*. El *software* se volvió patentable en Estados Unidos en 1981, año también de lanzamiento del *personal computer* (PC) por IBM, y Microsoft fue la primera empresa que invirtió masivamente en patentes para proteger sus productos (Allison et al., 2006).

Frente a los sistemas operativos de los constructores, específicos a su *hardware*, en el ámbito académico se crearon el lenguaje C y el sistema Unix (Ritchie et al., 1988), capaces de ser compilados y correr en múltiples *hardware*, ofreciendo una interfaz unificada al usuario. Unix fue adoptado por la industria y se difundió, pero también se balcanizó en múltiples versiones, y no faltaron pleitos por PI. De los arreglos de estos [\[12\]](#), surgieron versiones libres de Unix, como NetBSD o FreeBSD, pero es curioso constatar que no lograron un ecosistema como luego el de GNU/Linux.

Otro conflicto sucedió en el MIT, cuando Richard Stallman no pudo acceder al código fuente del piloto de una impresora para adaptarlo a otra. Entiende entonces

el poder que confiere mantener secreto el código fuente del *software* y el riesgo que significa para su comunidad, y decide en 1984 fundar la FSF y el proyecto GNU, que busca volver a escribir un sistema enteramente libre. A fines de los años 80, muchas herramientas de GNU, como su compilador GCC, eran de uso muy difundido. Pero faltaba el núcleo, el *kernel* (Stallman, 1999).

En 1991, Linus Torvald completa el sistema operativo libre con su *kernel* Linux. Por lealtad al proyecto GNU, cuyas herramientas utilizaba, Torvald adoptó la licencia GPL, propia del SL con *copyleft*. Pero, al diseñar una interfaz de módulos en la versión 2 del núcleo Linux, quiso permitir —pese a la interpretación estricta que pretendía la FSF, o al menos Stallman— el desarrollo de módulos bajo cualquier licencia, incluso privativas, entregando solo el ejecutable:

Acabamos decidiendo (o tal vez yo acabé decretando) que las llamadas al sistema no se considerarían como enlace contra el núcleo. Es decir, cualquier programa que se ejecutara sobre Linux no se consideraría cubierto por la GPL (Torvalds, 1999, p. 10).

Probablemente, esta elección salomónica (adoptar la GPL y su carácter viral para obras derivadas, al mismo tiempo que daba una interpretación limitada del *copyleft* —considerando que no aplica para un programa que interactúa con el núcleo, sino solo para un *software* derivado del código—) jugó un papel importante en la oportunidad que logró GNU/Linux de construir un ecosistema viable de magnitud. A fines de la década de los 90, este fue acuñado como el *open source* (Perens, 1999).

Los proyectos empresariales de *software* solían adoptar metodologías clásicas de la industria, que detallan primero la concepción y luego proceden al desarrollo unitario de las partes a partir de una especificación exhaustiva, metodologías que fueron cuestionadas desde los inicios (Royce, 1987) y no dieron buenos resultados. El *software* privativo de fines del siglo XX generó una economía injusta, de dependencia y de cuasimonopolios, fracasando en calidad, mantenimiento y evolución, así como en generar estándares de almacenamiento e intercambio. Entendiendo que el desarrollo de *software* es una actividad creativa y no industrial, la academia y las comunidades del FLOSS concibieron metodologías ágiles (Wilkinson, 2018) y lograron un acervo de *software* colaborativo mucho más versátil y de mayor utilidad, desarrollo que está

íntimamente vinculado al despliegue de la Internet:

Desde hace casi veinte años, los grupos que desarrollaron Internet —científicos e ingenieros informáticos— también han desarrollado nuevos enfoques de colaboración utilizando Internet como plataforma de comunicación y mediante licencias innovadoras construidas en torno al principio del Copyleft. El resultado es la creación de proyectos de SL, regímenes de propiedad común o comunes (Schweik, 2007, p. 302).

Algunos ejemplos de éxito del SL, como el del navegador Firefox, nacido de la liberación de Netscape (Hamerly et al., 1999), no fueron solamente cuantitativos. Cuando alcanzó estabilidad, no solo dominó el mercado, sino que logró plasmar una nueva generación de estándares en el acceso a la Internet: la web 2.0.

Progresivamente, la industria del *software* privativo adoptó los productos, metodologías y herramientas del FLOSS, mejorando en proporción sus propios sistemas. A partir de los años 2000 emergió la economía en línea, con los nuevos gigantes de la Internet: los GAFAMN. El concepto de comunidad que se manejaba en el FLOSS, funcional para construir audiencias en línea, fue central en la mutación del paradigma.

Como objeto técnico, desde el punto de vista concreto de partes de mercado, el FLOSS ha ganado por una victoria abrumadora sobre el *software* propietario[13]. Ya en el año 2000 Apple, al borde de la quiebra, logró renacer entre otras cosas por tomar la decisión estratégica de basar la nueva versión de su sistema operativo MacOS X en FreeBSD[14]. Gran parte de las contribuciones al núcleo Linux proviene hoy de grandes empresas, incluida Microsoft, que en 2012 llegó a ser el primer contribuidor[15].

El pacto entre desarrolladores de *software* del OS funcionó bastante bien. Hoy tenemos herramientas de comunicación, colaboración y constitución de acervo de código como Git, con sus frontales web GitLab (como *software*) o Github (como servicio, que fue adquirido por Microsoft). Preservan y vinculan cada una de las contribuciones de *software*, atribuyen (por firma electrónica basada en encriptado fuerte) la autoría de cada aporte, preservan huella de las versiones distribuidas, acumulan y automatizan las pruebas de nuevas entregas de código, así como los procesos de despliegue, los cuales se realizan *en la nube*, abstracción funcional de nuestros sistemas operativos tradicionales.

Cualquier departamento de tecnología de la información, por modesto que sea, puede ser partícipe de ese gran proyecto global. En la Red de Unidades Informáticas de la Udelar en el Interior, por ejemplo, mantenemos una plataforma de servicios que integramos en un repositorio reservado, pero que desarrollamos utilizando y compartiendo *roles* en la Ansible Galaxy, módulos de código reutilizables por otros para fines similares[16]. Con base en estas herramientas y estas prácticas, se acuñó el término *DevOps* (por *desarrollo* y *operaciones*), que designa una cultura profesional en torno a un conjunto de herramientas y prácticas que asegura diseño, desarrollo, evolución y operaciones del *software* integradas en forma continua (Ebert et al., 2016).

Pero, desde el punto de vista de la libertad, el horizonte es mucho más sombrío. Hoy se habla de *SaaS: software as a service*, ‘*software* como servicio’. Es en tu teclado que escribes y en tu pantalla que lees, pero la información no está en tu dispositivo local, su procesamiento y almacenamiento se realizan en la *nube* de tu proveedor en línea. Y ya no eres tú quien administra tus dispositivos: tu proveedor de Internet administra tu enrutador de acceso, Google o Apple administran tu teléfono móvil, tablet, o *smart TV*, Apple o Microsoft tu ordenador personal. No importa que sus aplicaciones se basen esencialmente en FLOSS, ya que ni siquiera tienes acceso completo a tu instalación del sistema. Es *software* cerrado. «Si es gratis, el producto eres tú», mediante recolección de datos que son monetizados y pueden ser usados para influenciarte a ti y a los demás, y que implican vigilancia permanente.

Si bien logró conquistar todos los otros sectores, GNU/Linux siempre fue minoritario en el *software* de escritorio, aunque hoy tenga una oferta pertinente. Quizás fue solo un asunto de *time to market*: Windows y su *challenger* Mac OS ya dominaban el mercado de la computadora personal cuando GNU/Linux tuvo una oferta completa para reemplazarlo.

Lo que le faltó al SL para conquistar la PC, en particular la adhesión de los constructores de *hardware* a la distribución y que el usuario pueda dispensarse de un servicio de soporte, llegó con el *software* cerrado (tal o cual distribución de Android) y con el *software* como servicio (Gmail, GoogleDocs, Facebook, Whatsapp...).

Algo parecido sucedió con las redes de comunicación de datos y sus protocolos. Se recuerda a menudo los orígenes militares de Internet: el ejército de los Estados Unidos habría querido construir una red descentralizada, cuya parte restante seguiría funcionando, aunque se destruyera cualquier subconjunto de su infraestructura. Pero también puede ser todo lo contrario: algunos investigadores libertarios habrían logrado que el ejército financiara el desarrollo de una red de comunicaciones revolucionaria, acorde a sus ideales. Sea como sea, el proceso de diseño de redes informáticas a partir de los años 60 vio dos grandes modelos confrontarse: por un lado, el modelo OSI, propuesto por los Estados, las empresas de telecomunicaciones y los principales fabricantes de computadoras de entonces[17], reunidos en torno a lo que fue el Comité Consultivo Internacional de Telégrafos y Teléfonos (CCITT, hoy la Unión Internacional de Telecomunicaciones [UIT])[18] —un modelo complejo, jerárquico y centralizado, similar al de las redes telefónicas y telegráficas— y, por otro lado, el modelo propuesto por la comunidad de Internet, esencialmente desde los laboratorios de investigación en computadoras y redes. Fue este último, simple, descentralizado, red colaborativa de redes, que demostró con la experiencia ser la mejor solución. En Internet, las propuestas de normas nacen como RFC: *request for comments*, ‘llamado a comentarios’. Cuando no hay más comentarios, la RFC puede convertirse en un estándar. Las normas no son, como en el OSI, un compromiso entre un ideal teórico complejo e intereses corporativos o nacionales, sino un mínimo necesario, consensuado y concreto de lo que los sistemas deben respetar para interactuar.

El carácter libre de la publicación de las RFC, ampliamente distribuidas, jugó un papel importante. Antes de ser *la red*, Internet ayudó a interconectar una cantidad de redes tempranas, en particular académicas y de investigación (Quarterman, 1990). Una RFC suele estar acompañada de un prototipo que implementa los protocolos que describe, en general como SL. A su vez, al desplegarse, Internet ofreció a las comunidades de SL el territorio para sus herramientas de comunicación y de trabajo colaborativo. La Internet se construye con SL y es el crisol de sus comunidades.

La red en sí es tan simple como posible. Asegura sus funciones esenciales de

enrutar y transportar información, y solo estas. Toda la inteligencia es desplazada a los márgenes de la red (Carpenter, 1996; Meyer & Bush, 2002).

Aunque sea contraintuitivo, estas parecen ser las leyes de la complejidad y el caos. Un conjunto de reglas simples, consensuadas y evolutivas entre múltiples actores independientes es más exitoso que una gran planificación de ingeniería para asumir el crecimiento y la innovación vertiginosos de un sistema global como Internet. Si los militares hubieran tenido que diseñar la red de comunicaciones global de sus sueños por su cuenta, es de dudar que hubieran dado a luz a la Internet tal como la conocemos.

Los nodos de la red no hacen diferencia entre diferentes tráfico, ni por origen, ni por destino, ni por contenido, en una estrategia de *mejor esfuerzo posible sin garantía de servicio*, que posteriormente fue acuñada como *neutralidad de la red* (Wu, 2003). Si bien es una herencia de cierta construcción técnica más que una intención y que, como diseño, es más arquitectural que funcional (Meyer & Bush, 2002), la *neutralidad de la red* es considerada hoy central para la libertad, la innovación y la competencia en la economía de la Internet abierta (La Quadrature du Net, s. f.; Lee & Wu, 2009) y ha sido el objeto de fuertes debates a nivel global^[19].

Finalmente, no podemos cerrar este recorrido por el SL e Internet sin mencionar el encriptado y los retos respecto a privacidad y vigilancia, que merecen un artículo en sí (Viñar Ulriksen, 2020). El progreso exponencial de la capacidad de cálculo confirmó los hallazgos de Turing al descifrar el código de la máquina Enigma de los servicios de inteligencia nazis: entre quien procura ofuscar un mensaje por encriptación y quien intenta descifrarlo, tiene ventaja quien encripta. Vivimos en un universo cuyas leyes amparan la encriptación, y con ella el derecho a la privacidad, la desmaterialización de una firma y los sistemas transaccionales distribuidos en base a *blockchain*. Pero esto requiere también el uso de SL, o confianza ciega en un *software* privativo que siempre podrá, silenciosamente, hacer uso indebido de la información una vez descifrada. Los servicios de inteligencia globales no se resignaron a la liberación de la encriptación a principios del milenio, sino que, sometiendo a los constructores de *hardware* y a los proveedores de *software* cerrado y *software* como servicio, establecieron una

escalofriante red global de vigilancia permanente (Moglen, 2013; Snowden, 2019).

4. Software libre y ciencia abierta

La primera encrucijada que plantea la filosofía del SL respecto a la ciencia reside en cualquier uso que se haga de un *software* en una actividad científica. La metodología científica exige que se provean todos los elementos necesarios para reproducir o refutar la investigación llevada a cabo, por ende, ¿cómo podemos legitimar la utilización de un *software* privativo, por ejemplo, para recolectar o procesar datos en una actividad científica? Sin el código fuente, receta de cocina de lo que realiza el *software*, no hay reproductibilidad o refutabilidad más que volver a pasar por la misma caja negra. La CA debe, entonces, adoptar el principio de usar solo SL. Varias iniciativas existen al respecto, como el *Code is Science Manifesto* (Yehudi et al., 2019) o el *Reproducibility PI Manifesto* (Barba, 2012), pero es necesario consensuarlo como un pilar de la metodología científica en la era digital.

Este principio debe extenderse a todas las formas de *software*, especializadas o no, y en todas las actividades vinculadas a la actividad científica. Haciendo ciencia, es oportuno no utilizar GoogleDocs, WhatsApp, Zoom, YouTube..., o incluso un buscador privativo como Google o Bing. Estos, para darte *mejores resultados*, toman en cuenta en cada consulta todas tus búsquedas anteriores. El buscador *te burbujea*, te encierra en la esfera de lo que se te parece. La respuesta no es la misma para cada persona, para cada consulta[20]. En la respuesta se introduce un sesgo, que eventualmente no es otro que el *sesgo conformista* de la especie humana y sus parientes más cercanas (De Waal, 2016).

Un reto actual, al que debe participar la comunidad científica y académica, es la construcción de una infraestructura federada de instancias de SL para ofrecer una alternativa a los proveedores comerciales mencionados, con un conjunto de servicios respetuosos de la privacidad, que puedan ser soberanos en cualquier nivel, en base a soluciones[21] como Element/Matrix para mensajería instantánea y videollamadas, Mastodon para redes sociales, Meet-Jitsi para videoconferencias, PeerTube para la publicación de videos a demanda, NextCloud para

almacenamiento personal y compartido, por ejemplo.

Pero el SL puede ser no solo instrumento, sino también ejemplo inspirador para otras disciplinas.

Los ideales colaborativos y los principios aplicados en los proyectos de FLOSS podrían ser aplicados para cualquier colaboración construida entorno a PI (no solo software) y podría potencialmente aumentar la velocidad a la que innovaciones y descubrimientos se hacen (Schweik, 2007, p. 277).

Analizando el SL y la ciencia en el marco propuesto por Hess y Ostrom para los comunes del conocimiento, Schweik (2007) identifica y estudia los factores que generan valor en un sistema de licenciamiento libre o abierto y que motivan a lxs integrantes de la comunidad de SL a participar de la constitución de comunes y lo compara al caso de la ciencia, en el que esencialmente se establecen a través del sistema de publicaciones arbitradas:

El entorno [de la Internet], en que archivos digitales pueden ser copiados y transferidos globalmente en un instante y a muy bajo costo, hace mucho más fácil tratar la información y el conocimiento como un bien común. Sin embargo [...], estos avances tecnológicos entran en conflicto con otras tendencias sociales y con la evolución de la legislación sobre derechos de autor, que empuja a tratar la información y otros productos digitales como bienes privados con fines lucrativos (Schweik, 2007, p. 278).

Las motivaciones de lxs contribuyentes al FLOSS y de lxs científicxs son de naturaleza muy similar, pero las modalidades y herramientas con las que elaboran su producción intelectual y establecen y ejercen sus derechos tienen varias diferencias. En múltiples oportunidades, Schweik (2007) subraya los límites del sistema de publicaciones científicas que acota lo que se comparte a un artículo, mero relato o informe de todo un proceso de investigación. A la inversa, las herramientas de control de versión de *software* y de trabajo colaborativo del FLOSS registran cada una de las versiones del código, su proceso de elaboración, de pruebas, las actas de revisión, su documentación, la gestión de proyecto, etcétera. En un GitLab o en Github, por ejemplo, existe un botón denominado *Blame* que permite atribuir individualmente a cada contribuyente cada una de las líneas de código de cierta versión de un archivo.

CA abarca mucho más que acceso abierto a publicaciones. Su materia de trabajo debe comprender no solo la literatura científica, sino también sus modelos, sus

experimentos, sus datos y sus hipótesis aún en formulación, tanto como el mismo proceso de verificación entre pares. Es importante ver esto reflejado en documentos marco como el Open Science Framework (Foster & Deardorff, 2017) y en la práctica de las revistas de acceso abierto que se han creado. *Image Processing On Line (IPOL Journal)*, por ejemplo, exige, además de la publicación de un *paper* tradicional, el código fuente del *software* que ejecuta los procesamientos de imágenes presentados, así como una instancia en línea de este, de manera que cualquier lectorx pueda probar los algoritmos con su propia base de imágenes, para sus propios objetivos (*IPOL*, s. f.; Limare et al., 2012). Un editor de esta *revista* notaba que la preparación de tal *publicación*, incluidos *software* y documentación, puede requerir hasta cinco veces más trabajo que una publicación científica tradicional. La reputación y otros valores que derivan de este proceso deben ser proporcionales al esfuerzo.

Por cierto, como señala Schweik (2007, p. 294) en su propuesta de *next-generation e-journal*, para hacer funcionar la idea en el contexto de la actual cultura científica y académica es central vincular el OS y las contribuciones de contenido abierto al proceso de publicación arbitrada, pero también es necesario un esfuerzo comunitario para que dicha cultura evolucione dando valor a estos nuevos parámetros de desempeño. El cambio es profundo, el concepto de autoría se transforma: ya no es individual respecto a unidades discretas y homogéneas, sino que circulan múltiples versiones de diversos objetos intangibles y son las herramientas de gestión de ese patrimonio las que atribuyen de forma granular las partes de autoría. El valor está en el conjunto, en la dinámica de múltiples producciones intelectuales que interactúan y producen, más que en la autoría individual.

A imagen de las actuales *forjas* y repositorios de *software*, como un GitLab o Github, las plataformas de CA contendrían no solo una biblioteca de contenidos textuales en acceso abierto bajo licencias Creative Commons, sino cualquier tipo de objeto intangible, sean modelos científicos, experimentos y resultados, bases de datos abiertas, protocolos y metodologías, en muchos casos expresados bajo forma de *software*, que se puedan intervincular dinámicamente y adaptarse y aplicar en múltiples contextos concretos.

¿Qué sería entonces el equivalente a una cultura y una práctica DevOps[22] en esa CA que se pretende construir? Una plataforma DevOps comprende no solo la instrumentación para la gestión del código, sino todo un conjunto de automatismos que vinculan, en procesos de integración y despliegue continuos, los repositorios de código con las capacidades de cómputo de la producción, permitiendo la emergencia de nuevas modalidades de *software*. De la misma manera, la CA puede así interconectar investigación teórica y experimental y distribuir las; diseñar, por ejemplo, modelos o experimentos que se difundan bajo licencias libres y que múltiples investigadores o grupos independientes puedan aplicar a su contexto, que luego consolide el conjunto de datos, el cual a su vez pueda retroalimentar la investigación teórica o la elaboración del modelo. Los mismos procesos de revisión a ciegas entre pares pueden ser instrumentados en los repositorios de CA, el arbitraje podría evolucionar hacia un actividad distribuida y multidisciplinaria, dada la diversidad de productos que son objeto de revisión. Se pueden imaginar interacciones con otros sectores más allá de la investigación científica, como herramientas comunes a la investigación y la enseñanza, con diversas modalidades de intervención del estudiantado, o instrumentos similares con la sociedad civil. Al igual que lo que sucede para el *software*, la CA puede derrumbar las barreras entre la investigación, el diseño y la industrialización de un producto, estableciendo procesos iterativos de optimización y mejora. Involucrando así el conjunto de la economía, se pueden constituir redes para abordar la complejidad de los problemas globales (Rischard, 2001), como ser establecer concretamente los circuitos de la economía circular necesaria a un futuro sostenible.

Un campo científico-tecnológico que naturalmente puede ser objeto de una re-ingeniería OS es la industria del *hardware*, para construir *hardware* libre. El movimiento de SL se interesó menos por el *hardware* libre que por el *software* en razón de su idealismo pragmático: con la tecnología del siglo XX, era razonable considerar que una electrónica abierta —en el sentido de tener interfaces externas documentadas— obedecía por completo al *software* desde el cual se la usaba. Lo que se necesitaba en ese entonces eran voluntades para estudiar y documentar por ingeniería inversa las interfaces de *hardware* cerrado. Pero hoy ya no es así: los progresos de la tecnología desdibujan los límites entre *hardware* y *software*, hacen

que interfaces abiertas no impidan la posibilidad de existencia de funcionalidades ocultas maliciosas en el *hardware*, y el *hardware* específico y cerrado se usa para forzar el uso de *software* cerrado. Además de la infraestructura federada de servicios mencionada, es esencial construir una industria del *hardware* libre, con un programa específico o regulando la industria actual.

Finalmente, ya mencionamos la encriptación y la importancia de la privacidad, pero conviene recordar que esta también es necesaria para la ciencia. «Si a primera vista una idea no es absurda, no hay esperanza para ella», frase que se atribuye a Albert Einstein[23] (Calaprice, 2011, p. 480). Es decir que, al descubrir algo nuevo, forzosamente será totalmente descabellado, porque si fuera razonable e intuitivo sin pensar diferente, ya se le hubiera ocurrido a alguien. Sabiendo que investigar, incluso en la era del *big data* y la ciencia de datos no es tanto llegar a cierta revelación luego de una acumulación de observaciones, experimentos y recolección de datos[24], sino emitir hipótesis, elaborar modelos y diseñar observaciones o experimentos que las confirmen o refuten, concluimos que investigar e innovar requiere poder emitir las hipótesis más diversas, incluso políticamente incorrectas, sean cuales sean los intereses que cuestionan. La vigilancia es contraria a la creatividad y a la innovación, y, por ende, a la inteligencia.

5. Elementos de conclusión

La CA no solo hereda de y se inspira en el movimiento del SL, sino que comparte con este el construir un acervo de bien común inmaterial, en el mismo territorio virtual que constituye la Internet, con principios similares de colaboración entre pares. Deberá transitar su plena mutación a la era digital y en ella la transformación de su sistema de evaluación, hoy centrado en el proceso de publicación arbitrada, pudiendo en este proceso inspirarse en las metodologías, prácticas y herramientas del FLOSS y otras comunidades. Hasta acá, todo va *en el sentido de la historia*.

Al mismo tiempo, tanto o más que para el SL, un desarrollo pleno de la CA requiere una Internet neutra, un derecho pleno a la privacidad —para lo cual necesitamos infraestructuras soberanas, con *hardware* y *software* libres— y un

marco regulatorio que contemple y proteja los comunes.

Debemos reformar la PI hasta volver a sus fundamentales. Para el derecho de autor tenemos una propuesta consensuada en el campo pro acceso abierto (Fundación Vía Libre, 2013; La Quadrature du Net, 2012). Debemos mantenerla en la agenda.

Para las patentes, que afectan varias disciplinas científicas más que el propio *software*, aún debemos diseñar y consensuar una propuesta, para la cual planteo aquí algunos ejes de trabajo[25].

Existe un mundo de producción intelectual que funciona muy bien sin patentes, que ni siquiera lee la parte pública de las patentes registradas[26], que rechaza por completo ese sistema por ser nefasto a sus formas de innovación y creación de riqueza intelectual. Este merece la misma atención y protección de la ley y del Estado que el sistema vigente de patentes, derivado de una evolución muy cuestionable de un sistema de otros tiempos. La realidad actual es una injusticia mayor para la seguridad de los comunes de la humanidad y un fuerte freno a la innovación, y eso es lo primero que conviene corregir.

Las oficinas de patentes no son actores económicos naturales, existen y funcionan porque se les delega una prerrogativa soberana: regular la innovación en pro del bien común. Si bien es legítimo que se registren ciertas patentes, también lo es que se considere el campo donde no aplican, no solo en función de reglas conceptuales como el «*flash* de genio» o el «paso inventivo no obvio», sino también del respeto a los derechos de las comunidades que prefieren innovar y construir comunes, en particular comunes inmateriales, fuera del reino de las patentes. Es necesario y urgente establecer un marco regulatorio para esta economía colaborativa, con instrumentos como el registro de obras libres y abiertas, y mecanismos de protección y resolución de litigios.

Se puede pensar en un *copyleft* extendido a las patentes: las licencias libres y abiertas podrán permitir o no que el bien inmaterial sea usado en el universo de las patentes (a la inversa no debe ser posible: como en la GPL, las patentes —excepto autorización explícita y sin restricciones— deben ser excluidas del mundo abierto, para no poner espadas de Damocles sobre los comunes). Esos

litigios deberán ser atendidos por las autoridades y no solo por el mercado, velando por no caer en los escollos del pasado.

Con Aaron Swartz, Edward Snowden, Julian Assange y otrxs, las causas del conocimiento abierto, así como de la privacidad y contra la vigilancia permanente, ya tienen sus mártires.

Como el del FLOSS, el movimiento de CA puede tener —y ya está teniendo— grandes logros aprendiendo de las comunidades tempranas de la era de la información, como los *hackers* del SL. Pero, al explorar lo que vincula estos movimientos, no dejan de ser centrales las encrucijadas filosóficas, éticas, morales y políticas que también comparten.

Referencias bibliográficas

Allison, J. R., Dunn, A., & Mann, R. J. (2006). software patents, incumbents, and entry. *Tex. L. Rev.*, 85, 1579.

Alvarellos Maceira, F., Bañobre Dopico, D., & - nbek .-. (2010). *software libre*.
<https://www.youtube.com/watch?v=FvLJ2JotttM>

Arup, C., & Van Caenegem, W. (2009). *Intellectual property policy reform: Fostering innovation and development*. Edward Elgar Publishing.
<https://books.google.com.uy/books?id=9BFdihBYkTOC>

Barba, L. A. (2012). *Reproducibility PI Manifesto*. 956610 Bytes.
<https://doi.org/10.6084/M9.FIGSHARE.104539>

Boyle, J. (2003). The second enclosure movement and the construction of the public domain. *Law & Contemp. Probs.*, 66, 33.

Calaprice, A. (2011). *The Ultimate Quotable Einstein*. Princeton University Press &The Hebrew University of Jerusalem.

Carpenter, B. E. (1996). *Architectural Principles of the Internet* (RFC 1958).
<https://datatracker.ietf.org/doc/html/rfc1958>

Creative Commons. (2016). Understanding Free Cultural Works. *Creative Commons*. <https://creativecommons.org/share-your-work/public-domain/freeworks/>

De Waal, F. (2016). *Are we smart enough to know how smart animals are?* WW Norton & Company.

Díaz Charquero, P., & Goñi Mazzitelli, M. (2021). *INFORME DE RESULTADOS DE LA ENCUESTA: ¿Cuánto sabes sobre derecho de autor?* (p. 24).

DatySoc. <https://datysoc.org/wp-content/uploads/2021/09/Resultados-de-la-encuesta-Cuanto-sabes-sobre-derecho-de-autor.pdf>

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *Ieee software*, 33(3), 94-100.

Editorial Policy. (s. f.). IPOL Journal · Image Processing On Line.

<http://www.ipol.im/meta/policy/>

FFII. (2004). *Patented Webshop*. <http://webshop.ffii.org/>

FFII. (2007, marzo 6). *Patentability and Patent Governance in Europe*.

<https://web.archive.org/web/20070306012134/http://eupat.ffii.org/>

Foster, E. D., & Deardorff, A. (2017). Open science framework (OSF). *Journal of the Medical Library Association: JMLA*, 105(2), 203.

Foucault, M. (1970). Que'est-ce qu'un auteur? En *Dits et écrits, 1954-1988, Tome III: 1976-1979: Vol. Texte n° 258*. Gallimard.

[https://www.psychanalyse.com/pdf/QU%20EST-CE%20QU%20UN%20AUTEUR%20-%20MICHEL%20FOUCAULT%20-%20CONFERENCE%201970%20\(25%20pages%20-%20246%20Ko\).pdf](https://www.psychanalyse.com/pdf/QU%20EST-CE%20QU%20UN%20AUTEUR%20-%20MICHEL%20FOUCAULT%20-%20CONFERENCE%201970%20(25%20pages%20-%20246%20Ko).pdf)

Free software Foundation, P. G. (1998). *¿Qué es el software libre?*

<https://www.gnu.org/philosophy/free-sw.html>

Free software Foundation, P. G. (2004, 2021). *Lista de licencias con comentarios*.

<https://www.gnu.org/licenses/license-list.es.html>

FreedomDefined. (2006). *Definition of Free Cultural Works*.

<https://freedomdefined.org/Definition>

Fundación Vía libre. (2013, octubre 10). Elementos para una reforma del derecho de autor. *FUNDACIÓN VÍA LIBRE*.

<https://www.vialibre.org.ar/elementos-para-una-reforma-del-derecho-de-autor/>

Girard, J.-Y., Gödel, K., Nagel, E., & Newman, J. R. (1997). *Le Théorème de Gödel*. Seuil.

Guadamuz González, A. (2005). Open science: Open source licenses in scientific research. *NCJL & Tech.*, 7, 321.

- Hamerly, J., Paquin, T., & Walton, S. (1999). Freeing the Source—The Story of Mozilla. En C. DiBona & S. Ockman, *Open sources: Voices from the open source revolution*. O'Reilly Media, Inc.
[https://walidumar.my.id/buku.elektronik/networking.umum/O'Reilly -
Open Sources, Voices from the Open Source Revolution.pdf](https://walidumar.my.id/buku.elektronik/networking.umum/O'Reilly_-_Open_Sources,_Voices_from_the_Open_Source_Revolution.pdf)
- Kapczynski, A. (2016). Order without intellectual property law: Open science in influenza. *Cornell L. Rev.*, 102, 1539.
- Kingston, W. (2001). Innovation needs patents reform. *Research Policy*, 30(3), 403-423.
- Kingston, W. (2009). Why patents need reform, and some suggestions for it. *Intellectual Property Policy Reform: Fostering Innovation and Development*, 11.
- La Quadrature du Net. (s. f.). *Redes de telecomunicaciones*. La Quadrature du Net. Recuperado 8 de octubre de 2021, de
<https://www.laquadrature.net/es/telecomunicacion/>
- La Quadrature du Net. (2012). *Éléments pour la réforme du droit d'auteur et des politiques culturelles liées*. La Quadrature du Net.
<https://www.laquadrature.net/elements-pour-la-reforme-du-droit-dauteur-et-des-politiques-culturelles-liees/>
- Lee, R. S., & Wu, T. (2009). Subsidizing creativity through network design: Zero-pricing and net neutrality. *Journal of Economic Perspectives*, 23(3), 61-76.
- Limare, N., Oudre, L., & Getreuer, P. (2012). IPOL: Reviewed publication and public testing of research software. *2012 IEEE 8th International Conference on E-Science*, 1-8.
- Meyer, D., & Bush, R. (2002). *Some Internet Architectural Guidelines and Philosophy* (Request for Comments RFC 3439). Internet Engineering Task Force. <https://doi.org/10.17487/RFC3439>
- Moglen, E. (2013). *Snowden and the Future*. <http://snowdenandthefuture.info/>
- Moyse, P.-E. (1997). La Nature du droit D'auteur: Droit de Propriete ou Monopole. *McGill, LJ*, 43, 507.
- Naciones Unidas. (1948). *La Declaración Universal de Derechos Humanos*. Naciones Unidas; Naciones Unidas. <https://www.un.org/es/about-us/universal-declaration-of-human-rights>

- Perens, B. (1999). The Open Source Definition. En C. DiBona & S. Ockman, *Open sources: Voices from the open source revolution*. O'Reilly Media, Inc. https://walidumar.my.id/buku.elektronik/networking.umum/O'Reilly_-_Open_Sources,_Voices_from_the_Open_Source_Revolution.pdf#%5B%7B%22num%22%3A2108%2C%22gen%22%3A0%7D%2C%7B%22name%22%3A%22XYZ%22%7D%2C0%2C792%2Cnull%5D
- Quarterman, J. S. (1990). *The matrix: Computer networks and conferencing systems worldwide*. Digital Press.
- Rischar, J.-F. (2001). High Noon: We Need New Approaches to Global Problem-Solving, Fast. *Journal of International Economic Law*, 4(3), 507-525.
- Ritchie, D. M., Kernighan, B. W., & Lesk, M. E. (1988). *The C programming language*. Prentice Hall Englewood Cliffs.
- Royce, W. W. (1987). Managing the development of large software systems: Concepts and techniques. *Proceedings of the 9th international conference on software Engineering*, 328-338.
- Schweik, C. M. (2007). Free/open-source software as a framework for establishing commons in science. En C. Hess & E. Ostrom (Eds.), *Understanding knowledge as a commons: From theory to practice* (pp. 277-309). MIT Press.
https://www.wtf.tw/ref/hess_ostrom_2007.pdf#page=292
- Snowden, E. (2019). *Permanent record*. Pan Macmillan.
- Stallman, R. (1999). The GNU Operating System and the Free software Movement. En C. DiBona & S. Ockman, *Open sources: Voices from the open source revolution*. O'Reilly Media, Inc.
https://walidumar.my.id/buku.elektronik/networking.umum/O'Reilly_-_Open_Sources,_Voices_from_the_Open_Source_Revolution.pdf
- Stallman, R. (2011). *Keep Control of Your Computing, So It Doesn't Control You!*
<https://www.gnu.org/philosophy/keep-control-of-your-computing.en.html#top>
- Stallman, R. (2015a). *Free software, free society: Selected essays of Richard M. Stallman*. Free software Foundation. <http://www.gnu.org/doc/fsfs3-hardcover.pdf>

- Stallman, R. (2015b). Why Open Source Misses the Point of Free *software*. En *Free software, free society: Selected essays of Richard M. Stallman* (pp. 75-82). Free *software* Foundation (Cambridge, Mass.).
<http://www.gnu.org/doc/fsfs3-hardcover.pdf>
- Torvalds, L. (1999). The Linux Edge. En C. DiBona & S. Ockman, *Open sources: Voices from the open source revolution*. O'Reilly Media, Inc.
https://walidumar.my.id/buku.elektronik/networking.umum/O'Reilly_-_Open_Sources,_Voices_from_the_Open_Source_Revolution.pdf#%5B%7B%22num%22%3A753%2C%22gen%22%3A0%7D%2C%7B%22name%22%3A%22XYZ%22%7D%2C%2C792%2Cnull%5D
- Viñar Ulriksen, D. (2020). Panoptique global contre réseaux communautaires. Quelques notes à propos de Permanent Record, d'Edward Snowden. *Desexil, praxis - mémoire -archives*. <https://desexil.com/panoptique-global-contre-reseaux-communautaires-quelques-notes-a-propos-de-permanent-record-dedward-snowden/>
- Wilkinson, H. (2018, noviembre 26). *El desarrollo de software como debería ser*. <https://www.youtube.com/watch?v=x4CXWBW5fMo>
- Wu, T. (2003). Network neutrality, broadband discrimination. *J. on Telecomm. & High Tech. L.*, 2, 141.
- Yehudi, Y., Aragon, S., Mun, Y., Konovalov, A., Knight, V., Rivière, P., Willighagen, E., Ilozulu, C., & Morley, A. (2019). *Code is Science Manifesto: First Release*. <https://doi.org/10.5281/zenodo.3484107>

Notas

[1] El código fuente de un *software* es su versión tal cual la elabora el desarrolladorx. Puede no estar incluido en la distribución del *software*, entregando solo el archivo binario, o ejecutable, que es suficiente para la computadora, pero insuficiente para un entendimiento humano.

[2] La Free Software Foundation (www.fsf.org), creada en 1984, elaboró y mantiene un juego de licencias para software libre, en particular la General Public Licence (GPL), licencia de SL con copyleft. También analiza la compatibilidad entre licencias de software (Free Software Foundation, 2004) y promueve el software libre a nivel global.

[3] Dicho por Maddog en conversación, confirmado por correo personal el 24/09/21 a las 08:54, traducción de quien escribe.

[4] Software libre, sociedad libre, de Stallman (2015a), also known as RMS.

[5] Derecho de autor, marcas y patentes, a los cuales se agregan más recientemente las

indicaciones geográficas de origen (IGP), cuyas especificidades también merecerían análisis más que amalgama.

[6] Puede verse un relato y análisis de litigios sobre patentabilidad del software en Allison et al., 2006.

[7] «Un producto o proceso simple es aquel que puede ser entendido o comunicado por un solo individuo», define Kingston (2001, p. 407).

[8] Eslogan de varias comunidades de software libre hispanófonas.

[9] GAFAMN, acrónimo para designar a los gigantes de la economía digital: Google, Amazon, Facebook, Apple, Microsoft y Netflix.

[10] Ver: https://es.wikipedia.org/wiki/Ley_de_Moore.

[11] Dado el tema del artículo, desarrollo aquí la analogía entre teorema y software, pero —que el rigor de la ciencia sepa disculpar— el arte culinario es probablemente una mejor analogía del software: la receta de cocina es una codificación precisa muy similar a un programa informático. A través de este paralelo clásico, Bañobre y Alvarellos (2010) explican muy claramente a los/as niños/as el ecosistema y las ventajas del SL respecto al privativo a través de un video de 2 min 30 seg.

[12] En la justicia anglosajona, los querellantes siempre tienen la posibilidad de hacer un deal en privado, dando fin al pleito. Esto se ha utilizado mucho en pleitos sobre propiedad intelectual, accediendo in fine las dos partes a cierta versión del bien inmaterial en disputa.

[13] En los sistemas operativos, la totalidad de las 500 primeras supercomputadoras del mundo están equipadas con GNU/Linux, y también lo están más del 98 % de los servidores en Internet, la computación embarcada se realiza en su mayoría con software de código abierto, y Android, el sistema operativo móvil de Google, también basado en el núcleo Linux, equipa más del 80 % de los teléfonos inteligentes vendidos en el mundo. Ver: https://en.wikipedia.org/wiki/Usage_share_of_operating_systems.

[14] Sistema Unix libre que ya mencionamos (páginas 341 y 342).

[15] Ver el informe anual de la Fundación Linux, página 13: https://www.linuxfoundation.org/wp-content/uploads/2020/08/2020_kernel_history_report_082720.pdf, y respecto al lugar de Microsoft: https://www.theregister.com/2013/09/16/linux_foundation_kernel_report_2013/.

[16] Los servicios están presentados aquí: <https://www.interior.edu.uy/>, el espacio de trabajo del código de infraestructura es <https://git.interior.edu.uy/cielito/adminsys/-/wikis/> y el espacio comunitario de roles Ansible compartidos está en <https://galaxy.ansible.com/udelarinterior>.

[17] Ver: https://es.wikipedia.org/wiki/Modelo_OSI.

[18] El CCITT se transformó en la UIT en 1992. Ver: https://es.wikipedia.org/wiki/Unión_Internacional_de_Telecomunicaciones.

[19] Los primeros escollos de la neutralidad se dieron en las conexiones por redes móviles en países periféricos, con ofertas como Internet.org de Facebook, que solo

permiten acceso a ciertos servicios, mediante una subvención de estos. Presentados como una acción humanitaria, la realidad es que el producto es el/la usuario/a. Esto ha permeado el mercado de las telecomunicaciones en la mayoría de países que no han consagrado legalmente la neutralidad de la red, con ofertas como «Whatsapp gratis», vulnerando la competencia entre servicios de mensajería instantánea.

[20] Existen buscadores como <https://DuckDuckGo.com/>, que reivindica ser neutro y respetuoso de la privacidad. Ofrecen una manera fácil de saltar a otro buscador: agregando «w!» en los términos busca en Wikipedia, «g!», en Google, etcétera. Hoy podemos obtener interesantes resultados combinando estas diferentes búsquedas, pero el objetivo debería ser un buscador basado en CA, cuyo algoritmo de ranking y relevancia pueda ser estudiado, y cuyos parámetros puedan ser ajustados por el/la usuario/a.

[21] Todas soluciones que integramos y desplegamos en la plataforma de servicios de la Red de Unidades Informáticas de la Udela en el Interior: <https://www.interior.edu.uy>.

[22] Ver en página Error: no se encontró el origen de la referencia algunas líneas sobre la cultura DevOps.

[23] Gracias a Rafael Grompone por esta frase y por haber investigado su atribución.

[24] Aunque escribiendo este paper supe que hay excepciones: al menos las «tecnologías simples» de la química de la industria farmacéutica (Kingston, 2001).

[25] Parto aquí de ideas e intercambios planteados en el contexto de la campaña contra la directiva europea de patentes de software (la cual no prosperó) (FFII, 2007), en particular, en el marco de las Rencontres Mondiales du Logiciel Libre en Dijon, Francia, en 2005 (https://fr.wikipedia.org/wiki/Rencontres_mondiales_du_logiciel_libre).

[26] En las comunidades de SL, es una recomendación no leer patentes. En efecto, si bien estas son documentos con cierto valor regulatorio, son de orden muy inferior. No aplica «la ignorancia de la ley no exime de su cumplimiento». Al contrario, no saber es un argumento legítimo de innovación por cuenta propia —aunque no pague el abogado—.

Nota del editor

El presente manuscrito fue aprobado para su publicación por el editor Gregory Randall.

Nota de contribución autoral

El texto es 100 % original del autor.