

En *Diálogos en Ciencia, Tecnología y Sociedad: conocimiento, producción colaborativa, innovación*. Córdoba (Argentina): Brujas.

FLOSS e innovación: exploraciones teóricas convergentes.

Zanotti Agustín y Hernán Morero.

Cita:

Zanotti Agustín y Hernán Morero (2018). *FLOSS e innovación: exploraciones teóricas convergentes*. En *Diálogos en Ciencia, Tecnología y Sociedad: conocimiento, producción colaborativa, innovación*. Córdoba (Argentina): Brujas.

Dirección estable: <https://www.aacademica.org/agustin.zanotti/42>

ARK: <https://n2t.net/ark:/13683/p6uq/bZ7>



Esta obra está bajo una licencia de Creative Commons.
Para ver una copia de esta licencia, visite
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>.

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>.

SEGUNDA PARTE

Discusiones sobre innovación e inclusión

FLOSS e innovación: exploraciones teóricas convergentes

Hernán Alejandro MORERO

Agustín ZANOTTI

Introducción

El término innovación es utilizado en distintos ámbitos. En su acepción más habitual, se refiere a la introducción de nuevas prácticas, técnicas, conocimientos o recursos en la actividad económica, traducidos en mejoras de productividad o rentabilidad. En los últimos treinta años, el sector informacional¹⁴ tuvo un rol central en el crecimiento y la innovación de las principales economías. En particular, la industria del software y servicios informáticos (SSI) ha sido extremadamente innovativa, y los trabajadores del sector han tenido que adaptarse a un entorno de cambio permanente (Lerner y Schankerman, 2010, p. 24).

El presente capítulo reúne elementos para la construcción de un marco analítico que permita comprender la articulación entre Software Libre y de Código Abierto (FLOSS¹⁵) e innovación. Brevemente, un programa es considerado de código abierto

¹⁴Nos referimos al sector dedicado a la producción de bienes y servicios basados en Tecnologías de la Información y la Comunicación (TIC), incluyendo el software, el hardware (industria electrónica) y las telecomunicaciones.

¹⁵En inglés *FLOSS* (*Free/Libre and open Source Software*). El término abarca los conceptos de software libre y software de código abierto, que, si bien comparten modelos de desarrollo similares, tienen diferencias en sus implicancias filosóficas. Sus referentes son la *Free Software Foundation*, en el primer caso, y la *Open Source Initiative*, en el segundo. En esta ponencia, utilizamos de manera

si los usuarios tienen la libertad de ejecutar, copiar, distribuir, modificar y mejorar el producto. La libertad no se refiere a la gratuidad de los programas, sino a la producción colectiva de los conocimientos y herramientas que constituyen las aplicaciones informáticas.

Tal como señalan Boldrin y Levine (2008), el FLOSS es un caso donde una parte del sector abandona voluntariamente las restricciones de la propiedad intelectual en favor de la creación en competencia y una mayor difusión/apropiación de innovaciones. El proceso acumulativo y multiautoral de ingeniería informática en red (Pereira et.al., 2007) permite la construcción y regulación del software como un tipo de bien común (Vercelli y Thomas, 2008; Bollier y Helfrich, 2014). Definidos por su no-rivalidad y no-exclusividad, podemos ubicarlo dentro de un conjunto de nuevos bienes comunes informacionales surgidos o repotenciados de la mano de las TIC (Hess, 2008).

El FLOSS ha crecido tanto en servidores como dispositivos inteligentes. También, se evidencia una extensión de lenguajes como Perl, PHP, JAVA, Python, entre otros, los cuales son sostenidos por fundaciones sin fines de lucro. Sobre ellos existe una cantidad de iniciativas en desarrollo¹⁶. El *open source* parece volverse así un modelo de negocios dominante, al tiempo que inclusive corporaciones tradicionalmente asociadas al software propietario¹⁷ han liberado, en los últimos años, código fuente de proyectos y contribuido al desarrollo GNU/Linux¹⁸.

unificada ambos términos, dado que refieren al mismo modelo de producción de software en términos económicos y productivos.

¹⁶Las estadísticas de Open Hub (mayo de 2016) indican que el desarrollo *open source* alcanza los 34 billones de líneas de código, con 3,7 millones de desarrolladores y 670.000 proyectos, de los cuales, sólo un 20% se encuentran actualmente activos (Open Hub sitio web, 2016). El FLOSS ha reconfigurado los modelos de negocios y las estrategias, tanto de PyMes como de Grandes Empresas Multinacionales, incluyendo a empresas como IBM, Oracle, Phillips, Nokia, Intel o SAP, que han empezado a optar, en las últimas décadas, por integrar aplicaciones FLOSS en sus actividades de I+D, servicios y productos *core* (Harison y Koski, 2010; Dahlander y Magnusson, 2005). Así, por ejemplo, en enero de 2005, IBM decidió liberar 500 patentes de software y donarlas a la comunidad FLOSS, con el objetivo de promover la innovación y el trabajo conjunto. Ese mismo año, tanto EMN de software FLOSS, como grandes EMN de software privativo y de la industria electrónica (IBM, Red Hat, Asus, Google, Nec, Suse, Philips, Sony, Toyota) conformaron una red de invención abierta, la *Open Invention Network*, ofreciendo colaborativamente un *pool* de patentes de acceso gratuito y libre para promover la innovación alrededor de Linux.

¹⁷El informe 2012, publicado por la *Linux Foundation* sobre la evolución del *kernel*, ubicó a Microsoft entre las primeras 20 empresas que contribuyeron en su desarrollo en el último periodo (Corbett, Kroah-hartman y Mcpherson, 2012).

¹⁸El kernel o núcleo de los sistemas basados en GNU/Linux, es un desarrollo activo desde 1991. En la actualidad, se considera uno de los mayores proyectos de programación existentes, así como una de las mayores experiencias de construcción colaborativa de software. El informe de evolución 2015 indica que el *kernel* cuenta (en su versión 3.18) con unos 47,986 archivos y 19 millones de líneas de

Apuntamos a que los principios del modelo FLOSS en procesos de innovación merecen una consideración diferenciada del conjunto del SSI. Se propone un abordaje desde modelos de innovación abierta o innovación privada-colectiva, los cuales cuestionan una asociación con incentivos monetarios directos, y buscan superar el dilema entre incentivos y difusión de innovaciones. Estos permiten valorizar las innovaciones sociales, la innovación inclusiva o las soluciones creadas en forma comunitaria.

En el caso argentino, una tendencia reciente es la formación de emprendimientos con modelos de negocios basados en FLOSS. Las experiencias en este sentido han dado lugar a las asociaciones como la Cámara Argentina de Empresas de Software Libre (CADESOL) y la Federación Argentina de Cooperativas de Trabajo de Tecnología, Innovación y Conocimiento (FACTTIC). Pocos trabajos se han concentrado en el papel del FLOSS en el espacio productivo local (Robert, 2004) y sus potencialidades en términos de un desarrollo integrado nacional (Heinz, 2006). Además del ámbito productivo, en la actualidad observamos experiencias que dan cuenta de una difusión del software libre en la administración pública, programas educativos, universidades y organizaciones del tercer sector (Zanotti, 2015).

A lo largo del capítulo, se repasan en primer lugar discusiones sobre teorías de la innovación y las particularidades del SSI como industria. Se analizan luego los elementos del modelo abierto en términos de formación de comunidades, motivaciones y procesos de innovación. Se presentan con ello algunas aproximaciones a sus modelos de negocio, junto con datos sobre sus alcances y tendencias a nivel local y global. En su conjunto, la propuesta busca analizar la producción de innovación en FLOSS y sus principales dinámicas, a la luz de marcos analíticos novedosos.

La innovación y el desarrollo informacional

A mediados del siglo XIX, los innovadores representaban un desafío al *statu quo*, generando resistencia y una percepción negativa. Los propios innovadores tenían que explicar por qué las innovaciones eran algo bueno y útil para la sociedad, ya que la innovación no era objeto de la agenda científica (Kotsemir et al., 2013). Ya en el siglo XX surgen las primeras teorías sobre innovación, con el objetivo de explicar el

código. Entre 8 y 12 semanas es el tiempo promedio hasta el lanzamiento de su siguiente versión estable, que incorpora mejoras, nuevas funcionalidades y soporte de hardware. En el último año, incorporó 96.000 cambios de parte de 4.169 desarrolladores, que representan al menos 518 corporaciones (Linux Foundation, 2015).

cambio económico. Joseph Schumpeter, su referente fundamental, la definió como uno de los factores internos que originan la evolución del sistema económico capitalista. El concepto se encontraba íntimamente relacionado con cambios estructurales y cuantificables en mercados donde operaban firmas innovadoras:

El cambio tecnológico en la producción de mercancías que ya se encuentran en uso, la apertura de nuevos mercados o el hallazgo de nuevas fuentes de aprovisionamiento, la taylorización del trabajo, la mejor manipulación del material, la introducción de nuevas organizaciones económicas [...] en pocas palabras, cualquier `forma de hacer las cosas de manera diferente´ -en la esfera de la vida económica- son todos ejemplos de lo que entendemos con el término de innovación (Schumpeter, 1939, p. 63).

La idea del empresario innovador de Schumpeter ha sido ampliamente discutida, y actualizada a partir de aportes neo-schumpeterianos o evolucionistas. Las aproximaciones de Nelson y Winter (1982) vinculan la innovación al conocimiento y la tecnología, en relación con otros agentes y factores, en consonancia con transformaciones del entorno social y económico. Los procesos de innovación tecnológica fueron sumando así otras cuestiones sin interés comercial directo, como el cuidado del medio ambiente o la equidad social.

O'Sullivan y Dooley (2008) resaltan la innovación es más que la creación de algo novedoso. Incluye la explotación para beneficio, agregando valor para los consumidores que lo perciben positivamente. En definitiva, una innovación implica dos descubrimientos: un descubrimiento técnico (invención) y un descubrimiento de mercado. Tal como sostienen Allen y Potts (2015):

El problema emprendedor es tomar este potencial tecnológico y completar toda la información faltante: posibles usos; por quién; en qué niveles de precio; con qué modelos de negocio; producido y distribuido de qué manera; con qué costos y fuentes de abastecimiento; con cuál inversión complementaria; sujeto a cuál competencia; con cuáles barreras políticas y regulatorias; con qué apoyo público o impuestos; en qué constelación comparativa de ventajas y externalidades económicas; entre otras (p. 4). [traducción propia]

En la medida que el rol de la innovación para el desarrollo económico ha sido dilucidado, algunos organismos internacionales han elaborado manuales de referencia para estudiarla y estimularla. La tercera edición del Manual de Oslo (2006) define en este sentido cuatro fuentes de innovación: de producto, proceso, mercadotecnia y organización (pp. 58-61).

Otras discusiones asociadas a la innovación toman en consideración dos aspectos principales: 1. incentivos a la innovación, en general referida al reaseguro de derechos, y 2. difusión de innovaciones (Dosi, Marengo y Pasquali, 2006).

Algunos autores presentan estas dos facetas como el *problema de la innovación*. De un lado, se refiere a los incentivos a la creación. En la medida que un proceso de innovación trae asociados costos y riesgos, si el conocimiento generado fuera inmediata y gratuitamente distribuido a todos, no resultaría conveniente destinar dinero a su desarrollo. En un esquema tradicional, estos incentivos vendrían dados por algún régimen de propiedad intelectual, que asegure derechos exclusivos de uso y comercialización por un periodo de tiempo determinado.

La otra cara se refiere a su difusión. En general, se asume que las innovaciones permiten incrementar la productividad o la eficiencia en los procesos económicos. De este modo, su no-difusión se considera una pérdida social. Esto al menos en términos ideales, ya que la difusión de innovaciones sólo se alcanza progresivamente: es necesario poseer capacidad operativa y conocimiento para utilizarlas, existen costos de adaptación y migración, entre otros factores (Lerner y Schankerman, 2010). A ello se suma otro fenómeno, definido como “derrame de conocimiento” (*knowledge spillover*), donde el conocimiento generado por un inventor ayuda a descubrir nuevas invenciones (Huber, 2012). Se produce entonces una retroalimentación positiva entre diferentes actores y la configuración de un ecosistema de innovación.

Varios autores coinciden en que el desarrollo informacional transforma en diferentes aspectos los procesos de innovación. Lessig (2001) sostiene que la difusión de Internet generó nuevos productos, nuevos mercados, nuevos medios de distribución (junto a nuevos formatos), nuevas demandas y nuevas formas de participación. La era de las comunicaciones globales termina de resaltar la importancia de modelos de *innovación abierta*, donde los desarrollos se llevan a cabo fuera de las fronteras de la firma e involucran una diversidad de colaboradores y motivaciones (Chesbrough, 2003; Schaarschmidt, 2012). Estos presentan una forma novedosa de entender los procesos de innovación, con connotaciones en términos de desarrollo, acción colectiva, apropiación de conocimientos y sinergias entre actores (Busarovs, 2013).

Las economías de compartición se nutren de la *producción entre pares*. Es ésta la innovación organizacional más significativa que ha emergido de la práctica social mediada por Internet, parte de un fenómeno mayor como es la cooperación *online*. *Red Hat, Google, Flickr, Vimeo, Photobucket, Tripadvisor, Booking.com, Yelp*, son todos ejemplos de modelos basados en la producción colaborativa. También *Wikipedia, SourceForge, OpenStreetMap*, entre muchos otros.

Benkler, Shaw y Hill (2015), la definen como una forma de creación abierta y compartida realizada por grupos en línea que: establecen y ejecutan objetivos de forma descentralizada; manejan un rango diverso de motivaciones, en particular no-

monetarias; se despegan de formas de propiedad exclusivas y relaciones contractuales; utilizan combinaciones de modelos participativos, meritocráticos y/o carismáticos.

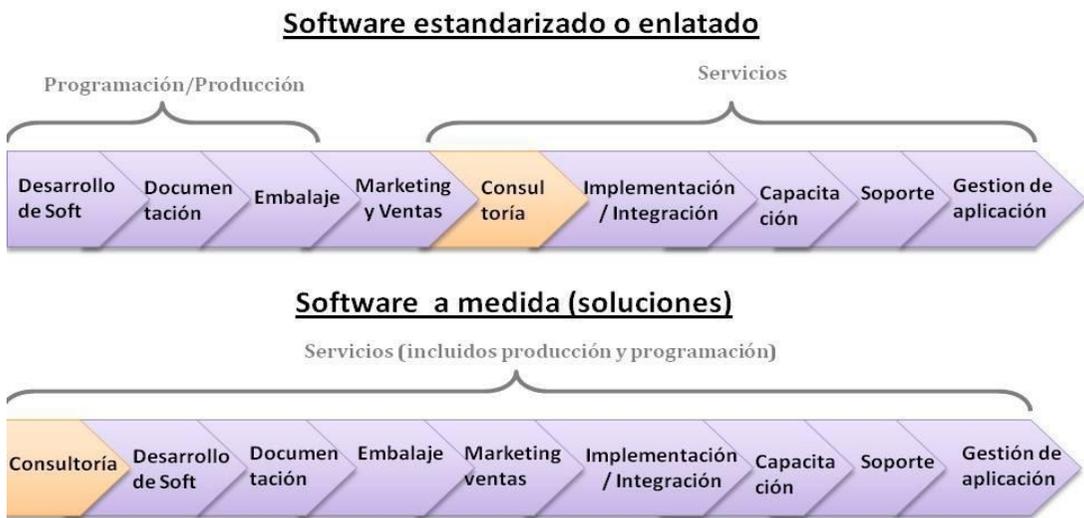
El caso del FLOSS demuestra, como veremos, una ecuación particular en cuanto a motivaciones, incentivos y reaseguro de derechos, en torno del cual se aglutinan proyectos con o sin fines de lucro.

Algunas especificidades de la industria del software

Los procesos de creación de software conllevan una variedad de tareas, que van desde la identificación de necesidades del cliente a la planificación y selección de herramientas, escritura del código, compilación, testeo, documentación, distribución. Estas tareas no se llevan a cabo necesariamente de un modo lineal. Es, sin embargo, posible realizar una estilización de la cadena de valor de la producción de software en etapas o actividades secuenciales según si la empresa desarrolla software enlatado, estandarizado o desarrolla software y ofrece soluciones a medida, como esquematiza la figura presentada a continuación¹⁹.

¹⁹Esta estilización sigue una propuesta del Floss Project de la Universidad de Naciones Unidas (UNU-MERIT), plasmada en su reporte de 2002 como resultado de una profunda investigación sobre FLOSS (Ghosh et al., 2002). En particular, esta estilización de la cadena de valor recoge tanto una reseña exhaustiva de la literatura y la investigación online realizada a esa fecha, así como muchas charlas de expertos y presentaciones en conferencias y rondas de negocio, como la Wizard of OS 2, en Berlín en octubre de 2001, o la LinuxWorld Conference de noviembre 2001 en Frankfurt. Consideramos que, si bien es perfectible, ayuda a distinguir de un modo aceptable las diferencias entre la producción FLOSS y propietarias en relación a distintas actividades de la producción de software y sus servicios.

Figura 1. Cadena de Valor de la producción de software



Fuente: Elaboración propia en base a UNU MERIT y Berlecon Research (2002)

El desarrollo de software es la escritura del código técnico para iniciar y controlar la funcionalidad de las máquinas, incluyendo la producción de herramientas de desarrollo. La documentación es la agregación y edición de información relacionada con la aplicación y uso de los productos de software²⁰. El embalaje o *packaging* es la agregación, integración y optimización de los productos de software o sus componentes (pudiendo incluir la corrección de errores). La agregación simple (p.e.: en un CD o en una web) no es *packaging*, desde este punto de vista, en cuanto no agrega valor. Estas tres actividades (desarrollo/documentación/packaging) se involucran en la producción de productos de software, antes de la etapa de marketing y ventas.

Luego, es posible continuar la cadena de valor con las actividades que son propiamente servicios informáticos. La consultoría informática involucra el análisis de situación del cliente, la definición de requerimientos (funcionales y técnicos) y la

²⁰Podemos mencionar tres tipos de documentación relevante: 1) El software en sí, incluye el código fuente en el caso de FLOSS; 2) Documentación electrónica, explicando el código dentro del software; y 3) Documentación editada sobre la funcionalidad y uso del software.

selección de soluciones de software acorde a ello. La implementación e integración de software incluye la instalación, configuración y customización del producto. La empresa puede ofrecer servicio de capacitación en el uso del software, a través de seminarios tradicionales, soluciones de *e-learning*, o incluso en el puesto de trabajo. El soporte incluye todas las acciones tendientes a ayudar al cliente a solucionar distintos problemas que vayan surgiendo en el uso. Por último, una serie de servicios pueden agruparse en lo que se denomina gestión de la aplicación o *application management*. Incluye todas las actividades para asegurar que el software se desempeñe en el modo deseado, como la gestión de la operación, el *upgrading*, mantenimiento y *versioning* del software.

El proceso descrito hasta aquí se adecua mejor a la producción de soluciones *enlatadas*, a lo sumo, customizables. Cuando la producción es *a medida*, cambia la secuencia de actividades, siendo la consultoría la etapa inicial del proceso de producción. En este caso, todo pasa a constituir un servicio que es la provisión de soluciones, incluyendo las actividades propias de programación como parte del servicio. Por otra parte, cada vez más se utilizan metodologías de desarrollo ágil, en ciclos de desarrollo que parten de una implementación “rápida y sucia”, pasan por un testeo inicial sobre potenciales usuarios y van luego perfeccionándose mediante un proceso de reescritura y pulido (Sutherland y Schwaber, 2012).

Algunos estudios del sector definen un conjunto de externalidades, entre las que recuperamos las siguientes: *externalidades de red (de usuarios)*, originadas debido a que los usuarios prefieren usar software que ya es usado por muchos otros usuarios; *externalidades de red (de desarrolladores)*, donde los desarrolladores, productores, empresas, reciben economías de red en la producción en un sistema determinado cuando aumenta el número de desarrolladores que escriben en esta plataforma; *externalidades indirectas derivadas de software complementario*, como resultado del éxito e introducción de una aplicación asociada (Lerner y Schankerman, 2010).

Los altos costos fijos iniciales asociados a la actividad, más el conjunto de externalidades mencionadas, pueden conducir a una concentración de mercado, con empresas que ofrecen paquetes de funcionalidades e intentan, de este modo, colonizar los diferentes “nichos”. A ello se suman, del lado de los consumidores, los altos costos de migración entre sistemas y aplicaciones. Estos quedan determinados por las necesidades de interoperabilidad con software y hardware, así como el costo de aprendizaje para los usuarios. En conjunto, estos generan un peligro de “*lock-in*”, esto es, quedarse encerrado en el viejo software, aunque este se encuentre atrasado en términos de productividad (Lerner y Schankerman, 2010).

FLOSS e innovación

a- El modelo abierto

Los orígenes de la producción de software se encuentran vinculados a laboratorios académicos, gubernamentales o de empresas, en los que, en gran medida, el intercambio de software, la modificación y la construcción sobre la base de otro software anterior, se consideraban una práctica común. En ese sentido, la producción de software es una actividad que encara un proceso de privatización a mediados de la década de 1970, junto con la escisión del software como comercializable independientemente del hardware.

El movimiento de software libre surge a principios de los 1980s desde los centros de desarrollo académicos como el MIT, como una reacción a esta privatización. Nace allí el movimiento inaugurado por Richard Stallman, el cual crea la *Licencia Pública General (en inglés abreviado GPL)* y funda la *Free Software Foundation (FSF)*, institución que comienza el desarrollo del sistema libre GNU (Stallman, 1983). Las licencias tipo GPL obligan a que las creaciones derivadas utilicen condiciones de uso similares, garantizando de este modo que el código permanezca abierto y no pueda ser apropiado con exclusividad.

Posteriormente, a fines de los 1990s, Eric Raymond estableció las virtudes tecnológicas y económicas del modelo de desarrollo en código abierto, apoyando el SL, pero permitiendo aún un modelo de negocios basado en el software propietario (Raymond, 1999). Desde nuestra definición, los proyectos FLOSS son desarrollos basados en comunidades que interactúan vía web. Estas comunidades se caracterizan por ser distribuidas y dispersas. Los miembros presentan normas y una cultura compartida que, junto con regímenes de licencias permisivas, garantizan que el conocimiento circule libremente.

Un proyecto FLOSS es típicamente iniciado por un individuo o un grupo pequeño en busca de una solución a una necesidad, ya sea individual, comercial o social. La estructura organizativa de los proyectos suele dividirse en un grupo nuclear y otro periférico más amplio. El *core* está formado por el/los líder/es del proyecto (donde suelen estar los "iniciadores") y un conjunto de mantenedores dedicados a evaluar las modificaciones del código realizadas por colaboradores, corregir errores, mantener la calidad del software. Por su parte, la periferia está formada por un número de desarrolladores cuya función es el testeo del software, la detección e información de errores y la generación de mejoras o parches del código fuente.

La división de los participantes en esta organización no es estructurada, sino que se basa en principios meritocráticos (Zanotti, 2012). Los proyectos funcionan sin un

proceso formal de *recruiting*, y el vínculo entre los desarrolladores y el proyecto es en general voluntario, no mediada por contratos formales. Los proyectos se valen de herramientas de desarrollo e infraestructura²¹, lo que incluye listas de correo para intercambio y discusión de ideas.

En función de estas características, los proyectos FLOSS han sido pioneros en una serie de prácticas que se difundieron luego al resto del SSI, como los ciclos incrementales, la distribución de versiones *beta* o los encuentros de desarrollo intensivo como *hackatones* o *sprints*. Tales formas de organización suelen vincular de manera fluida aportes voluntarios y de empresas de diversa envergadura, a la vez que promueven el patrocinio y la gestión de recursos (Zanotti, 2014).

La contribución a proyectos FLOSS beneficia de diferentes modos a desarrolladores y empresas. Por el lado de los desarrolladores, el placer personal de contribuir en la creación del código, el valor de uso del software, el aprendizaje y la capacidad de vincularse con personas con altas habilidades, son algunas motivaciones individuales. Estas pueden resultar luego redituables en términos económicos. (O'Mahony, 2003). La literatura coincide en que las motivaciones son heterogéneas y se mezclan diferentes incentivos monetarios y no-monetarios. Estudios recientes avanzan en definir perfiles motivacionales entre comunidades y culturas, interacciones entre diferentes perfiles, así como decisiones en el diseño de las comunidades que puedan moldear las contribuciones o mejorar su performance (Benkler, Shaw y Hill, 2015).

Por el lado de las compañías comerciales, éstas pueden interactuar con proyectos FLOSS de varias maneras. Algunas tendencias incluyen:

- Hacer que los empleados destinen horas de trabajo a contribuir con proyectos, formal o informalmente. Esto mejora las habilidades de los programadores y las herramientas pueden luego incorporarse a los proyectos de la firma.
- Ofrecer servicios alrededor de proyectos *open source*, lo que se conoce como “vivir simbióticamente”.
- Cuando un desarrollo está quedando atrasado o no logra imponerse en el mercado, es una tentación abrir el código para que cobre nuevo impulso.
- Las pequeñas empresas sienten desconfianza de interactuar con grandes firmas en entornos propietarios, porque pueden quedar atrapados en sus

²¹Los proyectos tienen usualmente un software de control de versiones, que permite a los contribuidores insertar nuevo código, realizar testeos de mejoras y volver a versiones anteriores fácilmente. Estas herramientas vuelven al proceso de desarrollo más interactivo y ágil.

tecnologías o relaciones comerciales. Entornos FLOSS permiten evitar esta dependencia y facilitan la migración.

- Existe un cierto recelo frente a que empresas privadas dirijan un desarrollo FLOSS. En función de ello, los proyectos asumen generalmente la forma de fundaciones u organizaciones sin fines de lucro, y canalizan de este modo sus aportes. Esto garantiza además una gestión democrática (Lerner y Schankerman, 2010; Colombo et al., 2013; 2014).

El desarrollo de una infraestructura colectiva de compartición de herramientas permite generar productos (a partir de sucesivos mejoramientos del código) de alta calidad y genera aprendizaje en los participantes. Estos procesos se basan en una evaluación crítica por parte de los miembros de la comunidad (Lee y Cole, 2003).

Junto a ello, el FLOSS permite generar innovación en relativa ausencia de mecanismos de apropiación privada. Lerner y Schankerman sostienen que “el *open source* podría solucionar la tensión entre la necesidad de proveer incentivos a la innovación y el deseo de promover la difusión del uso de tecnologías de punta” (2010, p. 8). Su extensión permitiría entonces que las mejores recetas fueran puestas a disposición de todos a bajo costo, al tiempo que programadores y empresas tendrían incentivos para contribuir voluntariamente en el desarrollo.

En base a estas características, el FLOSS se vincula a conceptos como *innovación de los usuarios*, *innovación democrática*, *innovación abierta*, *innovación colectiva*. Allen and Potts (2015) proponen asimismo el término *comunes de innovación*, para entender los espacios comunitarios como espacios donde se comparte no sólo conocimiento sino además información respecto de potencialidades emprendedoras. Tal información, ampliamente dispersa, es difícil de ponderar en sus partes y sólo significativa en su conjunto. Su valor se determina por la experimentación y el ensayo-error, por lo que los *comunes de innovación* pueden pensarse como espacios que mantienen el camino abierto y evitan las *clausuras* de las implementaciones tecnológicas.

b- Modelos de negocio

Los estudios sobre SSI concluyen a menudo que, a nivel de empresas, una tendencia frecuente es entremezclar FLOSS y software propietario, diversificando modelos de negocio. De este modo, muchas empresas desarrolladoras venden productos propietarios mientras, al mismo tiempo, contribuyen al desarrollo de código abierto (Lerner y Schankerman, 2010). Por otra parte, la mayor parte de las actividades de servicios (consultoría, implementación, capacitación y la gestión de la aplicación) no

tienden a presentar diferencias significativas si son provistas basadas en software propietario o FLOSS.

El resto de las actividades de la cadena de valor sí presentan características diferenciadas entre ambos modelos. En su estructura de costos, en términos generales, el modelo propietario conlleva una alta inversión inicial. Por el contrario, al tratarse de bienes informacionales, los costos de distribución son bajos. En el caso FLOSS, en la medida que se reutiliza código ya disponible, los niveles de inversión inicial son menores. Algunas investigaciones consideran, sin embargo, que existen mayores costos asociados a su implementación, cambio de sistemas, aprendizaje, interoperabilidad y soporte (Lerner y Schankerman, 2010).

La actividad de desarrollo de software suele presentar asimismo características diferenciadas entre una forma organizativa “más jerárquica” (siguiendo el principio de la “catedral”), asociado al software propietario, y una más “horizontal y dispersa” (bajo el principio del “bazar”), asociada al FLOSS (Raymond, 1999).

En el caso del software propietario, éste es usualmente vendido con la documentación, por lo que tal etapa suele integrar uno de los últimos pasos del proceso de desarrollo dentro de la empresa desarrolladora, quien además lo vende y realiza el marketing. El FLOSS brinda la oportunidad de separar estas últimas actividades de quienes desarrollan el software, ofreciendo modelos de negocios basados en el *packaging* y las ventas, con empresas especializadas en reunir el software y venderlo empaquetado. Buena parte de las empresas FLOSS se dedican a la etapa de distribución, *marketing* y ventas, tanto de sistemas operativos originales como de software especializado o de nicho. Esto involucra la agregación de los componentes, su adaptación y software complementario.

Por último, hay una importante distinción en las actividades de soporte. En el caso de FLOSS, este es ofrecido en primer lugar por los foros de la comunidad. Como esto no es aceptable para todo tipo de usuarios, surgen ofertas de soporte especializado, por parte de distribuidores y empresas OSS independientes. En el caso propietario, generalmente es ofrecido por una empresa especializada o el propio desarrollador del software.

Todas estas diferencias productivas del FLOSS han potenciado la posibilidad de desintegrar estas etapas/actividades de una misma unidad productora, dando origen a la oportunidad de diversos modelos de negocios especializados en una o algunas de estas etapas. Esto ha impulsado una tendencia desmonopolizadora en el sector de software en general.

c- Alcances y mediciones

Investigaciones recientes revelan que la participación del FLOSS en la industria va en aumento. Lerner y Schankerman (2010), en base a una encuesta realizada en 15 países, incluyendo entre ellos a Brasil, México y Chile, concluyen que cerca del 40% de las firmas involucradas en el desarrollo de software contribuyen en estos proyectos. Del conjunto, los países emergentes destacan entre los que más utilizan *open source* de manera exclusiva, siendo ejemplos de ello Brasil, Kenia y Rusia.

La encuesta de 2016 de BlackDuck²² indica, asimismo, que dos tercios de las empresas de SSI animaron a sus trabajadores a contribuir con proyectos de código abierto en el último año, lo cual representa un incremento del 5% respecto al año anterior. El 59% de las compañías afirmaron que el *open source* es una ventaja competitiva, mientras que un tercio mantiene personal tiempo completo en estos proyectos. El informe concluye que:

el código abierto es hoy en día de forma inequívoca un motor de innovación. Lo que está haciendo es impulsar tecnologías como sistemas operativos, la nube, el Big Data y el Internet de las Cosas. Existe una nueva generación de compañías de código abierto que están ofreciendo soluciones atractivas al mercado (Hammond, Santinelli, Billings y Ledingham, 2016, p. 8). [traducción propia]

La información disponible para el caso argentino, en base a una encuesta realizada en 2011²³ indica coincidentemente que más del 60% de las empresas de software utiliza o aporta al desarrollo en código abierto. De éstas, casi el 97% lo utiliza en sus procesos productivos y alrededor de dos tercios desarrollan módulos o programas completos FLOSS. El estudio remarca un desempeño fuertemente innovador en el conjunto del sector, así como una escasa prioridad a la aplicación de patentes o derechos de propiedad (Morero y Borrastero, 2015).

Desde el punto de vista económico, la actividad en comunidades de FLOSS plantea el problema de la ausencia de transacciones monetarias medibles y cuantificables (Ghosh, 2003). En general, presenta el problema de cómo medir una actividad económica no monetaria, lo que genera una desventaja para la mayoría de las investigaciones sobre este fenómeno, dada por la ausencia de datos empíricos, fácticos y verificables a gran escala. Esta dificultad se ha extendido, naturalmente, al

22La encuesta constó en 1313 casos de 64 países, e indagó sobre temas como: involucramiento y participación en proyectos FLOSS, inversiones y negocios abiertos, mejores prácticas, seguridad, y previsiones futuras (Hammond, Santinelli, Billings y Ledingham, 2016)

23Esta se realizó en el marco del Proyecto de Investigación “Capacidades de absorción y conectividad en sistemas productivos y de innovación locales” de la Fundación Carolina (id. 386317), dirigido por Gabriel Yoguel (UNGS). Esta encuesta fue aplicada a 250 empresas de software sector de la Ciudad Autónoma de Buenos Aires, Conurbano Bonaerense, Córdoba y Santa Fe y cubre el período 2008-2010.

estudio de un fenómeno económico como es la innovación. Ocurre que la mayor parte de la bibliografía de la naturaleza del proceso innovativo en FLOSS centra su análisis en estudiar el proceso de desarrollo a nivel de proyectos o de comunidad²⁴.

Esta forma de desarrollo del proceso de innovación dificulta especialmente su mensurabilidad, dado que ocurre difuminado en una comunidad distribuida y dispersa, incluso globalmente. Esto ha llevado a que la mayoría de los relevamientos disponibles sobre la producción FLOSS, o bien toman como unidad de relevamiento a los desarrolladores y trabajadores informáticos (WIDI 2001²⁵, BCG 2002²⁶, FLOSS 2002²⁷, Floss-US Stanford Report 2003, UNGS-SADIO 2004²⁸, FLOSS WORLD Project 2007²⁹), o bien proponen una unidad de relevamiento más reducida, como la descomposición autoral del código fuente.

Gran parte de la innovación FLOSS y su impacto económico (tanto en términos de empleo, extensión de innovaciones de productos u organizacionales, generación directa e indirecta de ventas y exportaciones, etc.) resultan así invisibilizados en las estadísticas tecnológicas, cuyos relevamientos están diseñados a nivel organizacional y nacional³⁰.

Si bien no hay relevamientos tecnológicos existentes a nivel mundial enfocados en FLOSS a nivel de empresa, sí es posible identificar preliminarmente algunos relevamientos sobre la actividad productiva, que pueden ser tenidos en cuenta como antecedentes³¹. En esa línea, se destacan las encuestas realizadas por el Proyecto

24 Kogut y Metiu (2001) estudian los casos de Linux y Apache. Von Hippel and von Krogh (2003) y Von Hippel (2005) presentan los casos de Apache y fetchmail. Lee and Cole (2003) y Tuomi (2006) el caso de Linux en general. O'Mahony (2003) estudia las formas de protección en 6 proyectos comunidades: GNU, Linux kernel, Apache, Debian, Gnome y Linux standard base.

25 Encuesta online realizada a 5478 desarrolladores.

26 Muestra seleccionada de un listado oficial de desarrolladores de proyectos FLOSS alojados en SourceForge.com, realizada entre 2001 y 2002, sobre 695 desarrolladores.

27 Encuesta online realizada a 2784 desarrolladores de FLOSS en 2002.

28 La encuesta se realizó mediante el suministro de un cuestionario online entre 2003 y 2004 a 167 trabajadores informáticos de la Argentina.

29 Reporte publicado en UNU MERIT (2007).

30 Un intento por estimar el valor creado por el FLOSS es el reporte de Licquia y McPherson (2015). El mismo establece que se necesitarían unos 1.356 desarrolladores durante 30 años para volver a crear las bases de códigos presentes en los proyectos colaborativos actuales de la Linux Foundation. El valor económico total de esta obra se estima en más de 5 billones de dólares. Es interesante remarcar que el informe sólo considera el costo laboral involucrado en la creación del software, y no el valor agregado en los diferentes procesos, mercados, productos y servicios creados a partir de éste.

31 Ver una revisión de ellos en Morero, Fernández y Sonnenberg Palmieri (2016).

European Libre Software Survey (ELISS) en Europa³² y las llevadas adelante por el Centro de Excelencia de Software de Fuentes Abiertas (CENATIC) en España³³.

Cierre

El presente capítulo buscó articular indagaciones teóricas en torno al FLOSS y sus particularidades en los procesos de innovación al interior de la industria del software y los servicios informáticos. Encontramos que las dinámicas de código abierto acompañan ciertos abordajes novedosos en materia de innovación, los cuales pueden ser comprendidos en cercanía con formas de innovación abierta o social de bienes comunes informacionales.

Para comprender tales procesos, comenzamos por desentrañar algunas particularidades del SSI, analizando las etapas que configuran su cadena de valor y algunas externalidades. Observamos que el modelo de código abierto tiene sus orígenes en los años 1980, a partir del movimiento de software libre y la creación de instrumentos legales y técnicos. La producción FLOSS involucra un ecosistema de comunidades donde interactúan gurús tecnológicos, expertos en dominios específicos e innovadores-usuarios (Pal y Madan Mohan, 2002). En tal ecosistema, individuos y empresas obtienen conocimiento, comparten experticia y contribuyen en proyectos. La comunidad es, a su vez, abundante en capacidades de programación y provee asistencia a los usuarios³⁴. Ofrece recursos de marketing, canales de distribución, contactos con potenciales clientes y oportunidades para construir una reputación en nichos de mercado.

Como observamos, estas innovaciones no se monetizan directamente y se difunden en redes, lo cual genera que varias de sus manifestaciones hayan sido invisibilizadas en los estudios sectoriales de innovación. Sin embargo, numerosas indagaciones

32El proyecto ELISS generó dos ondas: la primera, en 2003 relevó a más de 140 empresas proveedoras de software OS italianas, mientras que la segunda (ELISS II) incluyó a firmas de Finlandia, Alemania, Portugal y España, además de Italia, totalizando una muestra de 920 empresas de software y 150 empresas FLOSS.

33 El Observatorio Nacional del Software de Fuentes Abiertas (ONSFA) realizó una encuesta en 2009 a 141 empresas españolas proveedoras de FLOSS sobre aspectos productivos, actividades innovativas y la relación con la comunidad, así como un relevamiento más amplio ESFA-SI en 2011 a 775 empresas FLOSS (de un total relevado de 1932 empresas de software).

34Una cuestión interesante aquí es que, cuando una empresa no cuenta con el suficiente dinero para atraer programadores ‘estrella’ por vías monetarias, este modelo de desarrollo ofrece la oportunidad de contarlos alrededor de “una buena idea, atractiva”, seduciendo a desarrolladores a ‘codear’ para satisfacer las propias necesidades técnicas y prácticas (Von Krogh, 2003).

recientes destacan una tendencia de difusión creciente del FLOSS en las dinámicas productivas SSI.

Bibliografía

ALLEN, D. y POTTS, J. (2015). The Innovation Commons – Why it Exists, What it Does, Who it Benefits, and How. Paper presented at the International Association for the Study of the Commons biennial global conference, Edmonton, Canada, 25-29 May, 2015. DOI: <http://dx.doi.org/10.2139/ssrn.2617141>

BENKLER, Y.; SHAW, A. y HILL, B. M. (2015) “Peer Production: A Form of Collective Intelligence”. En Malone, T. W. & Bernstein, M. S. (Eds.). *Handbook of collective intelligence*. Cambridge: MIT Press.

BOLDRIN, M. y LEVINE, D. K. (2008). *Against Intellectual Monopoly*. Cambridge University Press, New York.

BOLLIER, D. y HELFRICH, S. [comps.] (2014). *Wealth of the Commons: A World Beyond Market and State*. Levellers Press, London.

BUSAROV, A. (2013). "Open innovation: current trends and future perspectives". *Humanities and Social Sciences*, 21 (2), 103–119. [Citado: 7 de junio 2017].

CHESBROUGH, H. W. (2003). *Open Innovation: The new imperative for creating and profiting from technology*. Harvard Business School Press, Boston.

COLOMBO, M. G.; PIVA, E. y ROSSI-LAMASTRA, R. (2013). Authorising Employees to Collaborate with Communities During Working Hours: When Is It Valuable for Firms?. *Long Range Planning*, 46 (3), 236-57.

COLOMBO, M. G.; PIVA, E. y ROSSI-LAMASTRA, R. (2014). Open Innovation and within-Industry Diversification in Small and Medium Enterprises: The Case of Open Source Software Firms. *Research policy*, 43 (5), 891-902.

CORBETT, J., KROAH-HARTMAN, G. y MCPHERSON, A. (2012) *Linux Kernel development. How Fast is it Going, Who is Doing It, What Are They Doing and Who is Sponsoring the Work. February 2012*. USA: Linux Foundation.

CORBETT, J., KROAH-HARTMAN, G. y MCPHERSON, A. (2015) *Linux Kernel development. How Fast is it Going, Who is Doing It, What Are They Doing and Who is Sponsoring the Work. February 2015*. USA: Linux Foundation.

DAHLANDER, L. y MAGNUSSON, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research policy*, 34(4), 481-93.

DAVID, P. A.; WATERMAN, A. y ARORA, S. (2003). *Floss-US Stanford Report: The free/libre/open source software Survey for 2003*. Stanford University.

DOSI, G., MARENGO, L. y PASQUALI, C. (2006). How much should society fuel the greed of innovators?: On the relations between appropriability, opportunities and rates of innovation, *Research Policy*, 35, 8.

FEENBERG, A. y FRIESEN, N. [Eds.] (2011). *(Re)Inventing the Internet. Critical Case Studies*. The Netherlands: Sense Publishers.

GHOSH, R. A. (2003). "Clustering and Dependencies in Free/Open Source Software Development: Methodology and Tools." *First Monday*, 8(4).

GHOSH, R. A., GLOTT, R., KRIEGER, B., y ROBLES, G. (2002). Free/libre and open source software: Survey and study. UNU-MERIT.

HAMMOND, J.; SANTINELLI, P.; BILLINGS, J. A. y LEDINGHAM, B. (2016) *Future of Open Source Survey Results* (presentación) (en línea). North Bridge + Blackduck. Disponible: <www.blackducksoftware.com/2016-future-of-open-source> [Consulta: 4/5/2016].

HARISON, E. y KOSKI, H. (2010). Applying open innovation in business strategies: Evidence from Finnish software firms. *Research policy*, 39(3), 351-59.

HEINZ, F. (2006). ¿Competitividad Informática, o Competitividad Social? En BORELLO, J.; ROBERT, V. y YOGUEL, G. (2006). *La informática en la Argentina. Desafíos a la especialización y a la competitividad*. Buenos Aires: Prometeo-UNGS.

HESS, C. (2008). "Mapping the New Commons". *12th Biennial Conference of the International Association for the Study of the Commons*. University of Gloucestershire, England.

HUBER, F. (2012). Do clusters really matter for innovation practices in Information Technology? Questioning the significance of technological knowledge spillovers. *Journal of Economic Geography*, 12(1), 107-126.

KOGUT, B. y METIU, A. (2001). "Open Source Software Development and Distributed Innovation." *Oxford Review of Economic Policy*, 17(2), 248-64.

KOTSEMIR, M. N.; ABROSKIN, A. y MEISSNER, D. (2013). Innovation Concepts and Typology—an Evolutionary Discussion. Higher School of Economics Research Paper No. WP BRP, 5.

LEE, G. K. y COLE, R. (2003). From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development. *Organization science*, 14 (6), 633-49.

LERNER, J. y SCHANKERMAN, M. (2013). *The comingled code: Open source and economic development*. The MIT Press, Cambridge.

LESSIG, L. (2001). *The future of ideas: The fate of the commons in a connected world*. New York: Random House.

LESSIG, L. (2008). *REMIX. Making Art Commerce Thrive in the Hybrid Economy*. London: Penguin Books.

LICQUIA, J. y MCPHERSON, A. (2015). A \$5 Billion Value: Estimating the Total Development Cost of Linux Foundation's Collaborative Projects. Linux Foundation. Disponible: www.linuxfoundation.org/sites/main/files/lfpub_cp_cost_estimate2015.pdf

MANUAL DE OSLO (2006). *Guía para la recogida e interpretación de datos sobre la innovación*. 3ra. edición. Ocede - Eurostat. Tragsa. Disponible en: www.fia.cl/Portals/0/UPP/Documentos/Manual%20de%20Oslo.pdf

MORERO, H. y BORRASTERO, C. (2015) Modificaciones en la organización del trabajo en empresas productoras de software abierto y formas de apropiación. En *Actas Vincular Córdoba. Jornadas de articulación público-privada para la innovación*. Universidad Nacional de Córdoba, Secretaría en Ciencia y Tecnología. Córdoba, 20 y 21 de agosto.

MORERO, H. A.; FERNANDEZ, V. y SONNENBERG PALMIERI, J. (2016). *Las encuestas internacionales a empresas de software libre y open source (Free/Libre Open Source-FLOSS)*. Córdoba/ Rafaela: CIECS (UNC-CONICET) / UNRaf.

NELSON, R. y WINTER, S. (1982). *An Evolutionary Theory of Economic Change*. The Belknap Press of Harvard. University Press. Cambridge.

O'MAHONY, S. (2003). "Guarding the Commons: How Community Managed Software Projects Protect Their Work." *Research policy*, 32(7), 1179-98.

O'Sullivan, D., & Dooley, L. (2008). *Applying innovation*. Sage publications.

PAL, N. y MADAN MOHAN, T. (2002). *Competing on open source: Strategies and practise*. Indian Institute of Management Bangalore, Bangalore.

PEREIRA, I., SEICEIRA, F., EGREJA, C., & ABRANTES, P. (2007). *O software livre e as empresas informáticas: Das tensões latentes às soluções híbridas*. Centro de Investigação e Estudos de Sociologia, e-working paper nº 28/2007, Lisboa.

RAYMOND, E. (1999). "The Cathedral and the Bazaar." *Knowledge, Technology & Policy*, 12(3), 23-49.

ROBERT, V. (2004). "Límites y potencialidades de la difusión de software libre en un país en desarrollo. El caso de la Argentina". En *Actas 33 Jornadas Argentinas de Informática e Investigación Operativa. Simposio sobre la Sociedad de la Información*. UNC, Córdoba.

SCHAARSCHMIDT, M (2012) *Firms in Open Source Software Development: Managing Innovation Beyond Firm Boundaries*

SCHUMPETER, J. (1939 [2002]). *Ciclos económicos: Análisis teórico, histórico y estadístico del sistema capitalista*. Zaragoza: Prensas Universitarias de Zaragoza.

STALLMAN, R. (1983). *El manifiesto de GNU*. [en línea] Disponible en: <http://www.gnu.org/gnu/manifesto.es.html>

SUTHERLAND, J. y SCHWABER, K. (2012) *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*, Cambridge, MA: Scrum Inc.

TUOMI, I. (2006). *Networks of Innovation. Change and Meaning in the Age of the Internet*. Oxford University Press, New York.

UNU MERIT. (2007). "FLOSSWORLD. Free/Libre and Open Source Software: Worldwide Impact Study",

UNU MERIT y BERLECON RESEARCH. (2002). *FLOSS FINAL REPORT*. The Netherlands: European Commission.

URIONA, M.; MORERO, H. A. y BORRASTERO, C. (2013). 'Catching up' en servicios intensivos en conocimiento: el caso de la producción de software y servicios informáticos de Argentina y Brasil. *Revista Iberoamericana de Ciencia, Tecnología y Sociedad*, 24 (8), 117-46.

VERCELLI, A., y THOMAS, H. (2008). "Repensando los bienes comunes: análisis socio-técnico sobre la construcción y regulación de los bienes comunes". *Scientiae Studia*, 6 (3), 427-442.

VON HIPPEL, E. A. (2005). *Democratizing innovation*. Cambridge, Mass; London: MIT Press.

VON HIPPEL, E. y VON KROGH, G. (2003). "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science". *Organization Science*, 14 (2), 209-223.

VON KROGH, G. (2003). "Open-Source Software Development." *MIT Sloan Management Review*, 44(3), 14-18.

ZANOTTI, A. (2012). Comunidades de software libre en Argentina: Algunas exploraciones y vectores de análisis. *Global Movements, National Grievances*, 602.

ZANOTTI, A. (2014). Desarrollo colaborativo de software libre en la Argentina: la experiencia de un hackatón cordobés. *Question*, 44 (1), 373-384.

ZANOTTI A. (2015). El software libre y su difusión en Argentina: mercado, Estado, sociedad. *Poliantea*, 11 (21), pp. 147-166. doi: 10.15765/plnt.v11i21.707

Sitios Web

Open Hub, the open source network, <www.openhub.net/> [Consulta: 4/5/2016].