

La economía del software libre y open source: Multinacionales, Pymes y Comunidades

Hernán Morero y Jorge Motta
Editores

**LA ECONOMÍA DEL SOFTWARE LIBRE Y
OPEN SOURCE: MULTINACIONALES, PYMES Y
COMUNIDADES**

**Hernán Morero y Jorge Motta
Editores**

Morero, Hernán

La economía del software libre y open source : multinacionales, pymes y comunidades / Hernán Morero ; Jorge Motta ; editado por Hernán Morero ; Jorge Motta. - 1a ed. - Ciudad Autónoma de Buenos Aires : Estudios Sociológicos Editora, 2020.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-3713-40-8

1. Software Libre. 2. Sociología. I. Motta, Jorge. II. Título.

CDD 005

Diseño de Tapa: Romina Baldo

Diagramación y corrección: Juan Ignacio Ferreras

© 2020 Estudios Sociológicos Editora

Mail: editorial@estudiosociologicos.com.ar

Sitio Web: www.estudiosociologicos.com.ar

Primera edición: febrero de 2020.

Hecho el depósito que establece la Ley 11723.

Libro de edición argentina.

El presente libro puede ser descargado desde el sitio web de nuestra editorial

**LA ECONOMÍA DEL SOFTWARE LIBRE Y
OPEN SOURCE: MULTINACIONALES, PYMES Y
COMUNIDADES**

Hernán Morero y Jorge Motta
Editores

Carina Borrastero
Ignacio Juncos
Florencia Manzo
Juan Gabriel Vélez
Agustín Zanotti



Esta obra está bajo una
Licencia Creative Commons Atribución-CompartirIgual 4.0
Internacional

ESI ESTUDIOS SOCIOLÓGICOS
EDITORIA

Estudios Sociológicos Editora

Estudios Sociológicos Editora es un emprendimiento de Centro de Investigaciones y Estudios Sociológicos (Asociación Civil – Leg. 1842624) pensado para la edición, publicación y difusión de trabajos de Ciencias Sociales en soporte digital. Como una apuesta por democratizar el acceso al conocimiento a través de las nuevas tecnologías, nuestra editorial apunta a la difusión de obras por canales y soportes no convencionales. Ello con la finalidad de hacer de Internet y de la edición digital de textos, medios para acercar a lectores de todo el mundo a escritos de producción local con calidad académica.

Comité Editorial / Referato

Pedro Lisdero. Doctor en Ciencias Sociales de América Latina (CEA-UNC) y Licenciado en Sociología (Siglo 21). Investigador del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) con sede de trabajo en el Centro de Investigaciones y Estudios sobre Cultura y Sociedad (CIECS), Unidad Ejecutora de CONICET, de la Universidad Nacional de Córdoba (UNC). Su campo de investigación se constituye a partir del cruce entre los estudios de la acción colectiva y el conflicto social, las investigaciones sobre el mundo del trabajo y la sociología del cuerpo y las emociones.

Roberto Merino. Doctor en Science Politique, Université Paris 8 (Vincennes Saint Denis), Francia. Master en Science Politique, Université Paris 8 (Vincennes Saint Denis), Francia. Profesor de la Universidad de las Artes y Ciencias Sociales (sede Valparaíso) y de la Universidad Bolivariana. Líneas de investigación: Experiencias concentracionarias y genocidio; Estado, estratificación y transformaciones políticas; Políticas de la violencia. Miembro del Comité de Redacción de la Revista Polis de la Universidad Bolivariana, Chile.

Pedro Robertt. Doctor en Sociología por la Universidad Federal de Rio Grande del Sur. Es Profesor Asociado de la Universidad Federal de Pelotas, actuando en los Programas de Pos-graduación en Sociología y en Ciencia Política. Sus trabajos actuales están direccionados a la etnografía del trabajo y a la epistemología y metodología de la investigación en ciencias sociales, con énfasis en el debate sobre reflexividad sociológica.

Este libro es resultante de los esfuerzos de investigación del Grupo de Economía de la Innovación y el Desarrollo (FCE-UNC) e investigadores del Programa CTS del Centro de Investigaciones y Estudios sobre Cultura y Sociedad (CIECS - CONICET y UNC). Aunque no hayan participado como autores de los capítulos, hay que destacar el trabajo de **Leiza Camilo Caro**, **Ana Valentina Fernández**, **Josefina Sonnenberg Palmieri** y **Pablo Ortiz**, en documentos preliminares, trabajo de campo, entrevistas, diseño de cuestionarios de encuesta y ejecución de relevamientos. El libro representa la culminación del proyecto de investigación PICT 2015-2703 *“Procesos de innovación en empresas de software libre y open source en Argentina”*, dirigido por el Dr. Hernán Morero; pero asimismo forma parte de una investigación de mayor alcance alrededor de las particularidades económicas del FLOSS que comprende varios proyectos de investigación del equipo. En particular, caben señalarse el Proyecto Consolidar Secyt-UNC 2018-2021 *“Procesos y tipos de innovación en industria y servicios: software libre, comercio electrónico y tecnologías 4.0 en empresas de Argentina”*, dirigido por el Dr. Jorge Motta; y el Proyecto UNRAF 2018-2019 *“Taxonomía de empresas de software según uso y producción de software libre y open source”*, dirigido por la Lic. Ana Valentina Fernández. Los autores agradecen especialmente la lectura y comentarios recibidos por la comunidad académica en diversos eventos a versiones preliminares de estos capítulos. Las opiniones expresadas en este documento son de exclusiva responsabilidad de los autores



Indice

Introducción.....6

Capítulo 1. Conceptos esenciales para el abordaje económico y productivo del FLOSS

1. El abordaje económico-productivo del FLOSS: los aspectos conceptuales y los avances de la literatura

1.1. Características económicas de la producción de FLOSS a nivel empresarial

1.2. La investigación económico-productiva sobre el FLOSS: un resumen de la literatura

2. El FLOSS en la industria del Software

2.1. La importancia del FLOSS en la industria global del sector de software

2.2. El uso y producción de FLOSS en el sector de software de la Argentina

3. Reflexiones Finales

Capítulo 2. La producción en comunidades FLOSS: empresas, Fundaciones y *governance*

1. Comunidades FLOSS

2. Empresas y la gobernanza FLOSS

3. Fundaciones FLOSS

4. Análisis de casos

4.1. GNOME

4.2. Fedora

4.3. Python

4. 4. LibreOffice

5. Discusión

6. Cierre

Fuentes Documentales

Capítulo 3. Grandes corporaciones de software y el FLOSS: cooperar en entornos abiertos como estrategia de competencia

1. Antecedentes y marco conceptual: las grandes corporaciones, el FLOSS y la open coopetition.

2. Resultados empíricos

2.1. El mercado global del software en la actualidad

2.2. Las empresas dominantes del mercado global del software con participación relevante en FLOSS

2.3. El caso de una empresa líder íntegramente FLOSS: Canonical

3. Discusión

Fuentes documentales

Capítulo 4. Las PyMEs de Software y el FLOSS

1. El uso y producción de FLOSS en Pymes de software

2. Estudios de casos de PyMEs FLOSS argentinas

A. GCoop

B. XTech

C. ECIC Systems

D. Kunan

E. Machinalis

F. TECSO

G. Entornos Educativos

3. Una síntesis de los casos de pymes estudiados

Conclusiones y Reflexiones Finales: La presencia del FLOSS en la industria del software y las limitaciones del *mainstream* en economía para comprenderlo

Bibliografía

Datos de autores y autoras

La economía del software libre y open source: Multinacionales, Pymes y Comunidades

Introducción

El *software* rodea nuestras vidas cotidianas, tanto como usuarios, así como consumidores, como trabajadores y como productores. Muchos de estos programas informáticos se hallan directamente contenidos en toda una serie de dispositivos electrónicos que nos circundan de un modo omnisciente. Son las instrucciones que permiten que intermedemos con todos los aparatos que interactuamos a diario y que éstos funcionen.

La manera en que se producen estos códigos que rodean toda nuestra vida ha sido revolucionada en las últimas décadas por la aparición del Software Libre y de Código Abierto o Free/Libre *Open Source* Software (FLOSS). Su producción en gran medida se realiza principalmente en proyectos de desarrollo de software en Comunidades de programadores, que ofrecen voluntariamente sus horas de trabajo y esfuerzo sin recibir “un pago” por ello y el software generado se produce con licencias de propiedad que tienden a prohibir su apropiación privada (las *copyleft*), puede descargarse gratuitamente y muchas veces impone que desarrollos ulteriores que se basen en él, deben estar igualmente disponibles para el público.

Por extraño que pueda sonar en un primer momento, tanto las empresas que desarrollan software y como las de otros sectores vinculados a la tecnología participan muy activamente en estas actividades de producción compartida de software. De hecho, los desarrollos que surgen de estas comunidades, de Linux y el software libre rodean continuamente nuestras vidas y son usados actualmente por importantes organizaciones privadas y públicas.

Por ejemplo, el software libre y de código abierto está presente en todos los dispositivos móviles que corren bajo Android y en los más 700.000 televisores vendidos cada día, pero además se utiliza en sistemas de defensa y submarinos nucleares, en muchos de los sistemas de control de tráfico ferroviario y aéreo

públicos.¹ Los servidores de Google, Facebook y Amazon implementan FLOSS (particularmente aplicando Apache) y ocho de cada diez transacciones financieras globales se realizan aplicando algún software de código abierto.² Hacia Linux han migrado su funcionamiento la bolsa de Nueva York (NY Stock Exchange), el London Stock Exchange, el operador financiero Deutsche Börse, la Qatar Exchange y la Shanghai Stock Exchange.³ Muchos de estos sistemas pueden procesar más de un millón de transacciones por segundo. Los grandes usuarios incluyen Gobiernos, Universidades, Ministerios, y con Linux funcionan grandes proyectos de alta tecnología, como el software de secuenciación del ADN, la Estación Espacial Internacional (*International Space Station*) y se utiliza para la distribución de más 10 *petabytes* de datos anuales que genera el Gran Colisionador de Hadrones operado por la CERN.⁴

En sus orígenes la comunidad FLOS pareció acotarse a ser un movimiento ideológico por la libertad en el acceso, uso y modificación de los programas informáticos. Sin dejar de serlo totalmente, actualmente, ha penetrado e impregnado a la industria global del software. Nada impide que una empresa descargue un software de este tipo y monte su oferta de servicios a partir de él, o lo ofrezca *customizado* a medida. Las grandes corporaciones de producción de software

1- Android es un sistema operativo basado en Linux para dispositivos móviles para celulares táctiles, tablets y, actualmente, se ha expandido hacia relojes inteligentes, televisores y automóviles. Ha sido lanzado con licencias GNU GPL v2 y Apache 2.0. En su segmento compite contra otras opciones propietarias de Apple, Microsoft, Nokia, Palm, BlackBerry y sistemas operativos FLOS como Tizen. Sin embargo, su participación de mercado oscila actualmente según la fuente entre el 75% (Statcounter) y el 90% (Xatara en base a Statista) de las ventas mundiales de dispositivos.

2- *Apache HTTP Server Project* es un servidor Web de acceso libre que es distribuido a través de una licencia *open source*. El desarrollo de este software se destaca por haber sido generado completamente por voluntarios, sin participación al menos directa de empresas. Desde abril de 1996, pocos meses después de su lanzamiento en 1995, Apache ha logrado convertirse en el servidor web más popular. Facebook por su parte funciona sobre una base de datos NoSQL, que es un proyecto OS llamado Apache Cassandra. Éste inicialmente fue desarrollado internamente en Facebook para impulsar la búsqueda de datos, y en 2008 fue liberado como proyecto OS, y desde 2009 es un proyecto comunitario de Apache. A este desarrollo, en la actualidad lo utiliza su competidor Twitter en su plataforma (Durand *et al.*, 2012). A su vez, Google tiene una infraestructura en el rango de 1.000.000 de servidores, los cuales están equipados con una versión a medida de Linux de esta clase.

3- EL LSE en 2011 implementó un motor de transacciones basado en Novell SUSE Linux Enterprise, denominado *Millenium Exchange*, sustituyendo a una plataforma basada en arquitectura de Microsoft.

4- En particular, la distribución Scientific Linux.

a nivel internacional y relacionados dedican personal y equipos permanentemente a colaborar en proyectos de desarrollo de software de la comunidad FLOS, de los cuales podría nutrir su cartera cualquier competidor, usuario o colaborador. Éstas además pueden lanzar su propio proyecto a la comunidad y liderarlos, contando con que otros programadores se sumen a colaborar y trabajar en él.

Uno de estos casos más llamativos, y que mejor ejemplifica la influencia del FLOSS en la industria del software, es el de Microsoft que pasó de tener una posición negativa acerca del desarrollo del software libre a ser, en 2016, un miembro platino de *Linux Foundation* que le implica ser colaborador de la comunidad financieramente y administrativamente.⁵ Tanto así ha llegado la extensión del compromiso de Microsoft con el aporte al desarrollo de software de código abierto que ha lanzado su primer sistema operativo Linux, Azure Sphere, una plataforma de seguridad orientada al Internet de las Cosas. Ya en 2009, esta gran corporación liberó con licencias GPLv2 unas 20.000 líneas de código de *drivers* de dispositivos a la comunidad del núcleo Linux, y liberó completamente su *plug-in* de servicios Microsoft Live Services para integrar sus servicios educativos con el sistema de gestión de cursos Moodle (un proyecto de la comunidad FLOS de plataformas educativas).⁶

Y no es el único gran actor de la industria global de software involucrado con acciones concretas: IBM en el 2005 liberó 500 patentes de software para los desarrolladores de FLOSS, y ese mismo año junto tanto a otras multinacionales de FLOSS como grandes empresas de software privativo y de la industria electrónica (IBM, Red Hat, Asus, Google, Nec, Suse, Phillips, 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 (Harison y Koski, 2010).

IBM cuenta incluso con otra serie de incursiones en el mundo del FLOSS: fue lanzador y patrocinante (“líder”) de un proyecto de desarrollo de la comunidad muy exitoso denominado Eclipse (una plataforma de servicios integrales para la programación, *Integrated Development Enviroment* o IDE) y que a fines de 2005 podía dar cuenta del 30% del segmento de mercado de IDE (Munga *et al.*, 2009).

5- En 2001 el CEO de Microsoft, Steve Ballmer, llegó a declarar que “*Linux es un cáncer que se pega a todo lo que toca*”.

6- Las distintas clases de licencias de software serán descriptas en el Capítulo 1 de este libro, específicamente en el Recuadro 1.

Desde mediados de la década de los 1990's abandonó sus esfuerzos de desarrollo de servidores web para volcar esos esfuerzos a colaborar en esta línea tecnológica con el proyecto FLOS Apache, quien mantiene una cuota de mercado pareja con Microsoft en el segmento de servidores y, por momentos, dominante. En esta línea, IBM liberó (500.000 líneas de código) de su programa Cloudscape a la Fundación Apache (producto que fue renombrado como Apache Derby) (Lerner y Schankerman, 2013) y ya en 2008 IBM contaba con más de 600 programadores dedicados íntegramente en más de cien proyectos OS (Dughera *et al.*, 2012). También Apple ha liberado el código fuente de su sistema operativo para servidores (Mac OS X Server, renombrado como Darwin) en 1999 y lo ha aportado a la comunidad de desarrolladores FLOSS (Bonaccorsi y Rossi, 2003).

Adicionalmente, en el año 2011 Hewlett-Packard (HP), colaborador activo de la comunidad Open Source y un gran aportante al desarrollo del proyecto Open Stack, anunció el ofrecimiento del software webOS bajo una licencia de código abierto para desarrolladores y empresas. Este software fue diseñado para los dispositivos HP y brindaba la facilidad de ser portable para múltiples plataformas y con una gran facilidad para el desarrollo de aplicaciones.⁷ Asimismo, en 2012 también pasó a ser un miembro platino de Linux Foundation. Además, HP liberó su aplicación Spectrum Object Model-Linker para ayudar a la comunidad OS a escribir software que conecte a Linux con la arquitectura de su hardware (Lerner y Schankerman, 2013).

Junto con HP, IBM y Microsoft, otros miembros platino de la Fundación Linux son Tencent, Fujitsu, Intel, Oracle, Samsung, AT&T, VMware (una subsidiaria de Dell), Cisco, Qualcomm, Hitachi, NEC y Huawei.⁸ Dentro de los miembros dorados puede mencionarse a Panasonic, Facebook, Toshiba y Toyota, entre otros. Sólo para ilustrar: en un espacio donde coinciden este tipo de actores, que es uno de los proyectos FLOSS actualmente más populares denominado OpenStack (ver Capítulo 3) colaboran más de 200 empresas, aportando equipos completos de programadores. Allí participan corporaciones tales como Intel, HP, AMD, Dell, IBM, Oracle y grandes empresas multinacionales FLOSS como Canonical, RedHat o SUSE Linux. Por ejemplo, las empresas manufactureras de

7- Este desarrollo terminó siendo adquirido por la firma LG en el año 2013 como plataforma para el desarrollo de un sistema operativo para su línea de televisores inteligentes. Desde esta firma también han coincidido en liberar el desarrollo del proyecto y permitir su descarga libre en la comunidad.

8- Los miembros dorados (*gold*) aportan anualmente 100 mil dólares, mientras que los miembros platino (*platinum*) donan aproximadamente unos 500 mil dólares al año.

hardware suelen participar para lograr una mayor compatibilidad de su oferta con el software disponible, como es el caso de HP que liberó su aplicación Spectrum Object Model-Linker para ayudar a la comunidad OS a escribir software que conecte a Linux con la arquitectura de su hardware.

Esta incursión del FLOSS en los últimos tiempos no solo se da en grandes empresas relacionadas a la industria del software y del hardware, sino que últimamente grandes estudios como DreamWorks, Epic Games, SideFX, Walt Disney Studios junto con grandes empresas como Google, Intel y otras se han convertido en miembros fundadores de “*The Academy Software Foundation*” con el objetivo de apoyar el desarrollo de estos códigos abiertos destinados a la industria cinematográfica, ayudando a los estudios a la incursión en, y administración de, proyectos de código abierto. Esta iniciativa nace como resultado de una asociación entre The Linux Foundation y The academy of Motion Picture, Arts and Sciences -el ente encargado de la organización de los Oscars- luego de que, a través de una encuesta realizada a varios estudios cinematográficos, se observó que más del 80% de ellos declaro utilizar algún producto FLOSS para las animaciones y/o efectos especiales de sus películas.

Además existen claros ejemplos de empresas líderes en productos exclusivamente de software libre y que han demostrado lo rentable que puede resultar el desarrollo de proyectos FLOSS, como es el caso de Red Hat, proveedor de soporte técnico corporativo, y Canonical, proveedor del sistema operativo Ubuntu.

Son muy numerosos los casos de software desarrollados internamente por grandes empresas del sector que luego son liberados a la comunidad y son accesibles a cualquier competidor. Por ejemplo Apple en 2005 convirtió a su aplicación WebKit y sus componentes a código abierto, y luego pasaron a colaborar en él Google, Nokia, Samsung y Motorola. Incluso en tecnologías emergentes, como Machine Learning, las corporaciones líderes han liberado sus principales herramientas como FLOSS: Google lo ha hecho con Tensorflow, Facebook con PyTorch y Microsoft con CNTK.

Además, al núcleo de Linux contribuyen con código de empresas que incorporan software en productos electrónicos tales como Nokia, AMD, Intel, HP y Samsung. De esta forma, el FLOSS está presente en productos de consumo final electrónicos. Incluso es posible que cualquier producto que incluya algún tipo de software, tenga algún componente OS.

Grandes empresas vinculadas al negocio de redes sociales tales como Spotify, NetLog y Facebook incorporan paquetes OS en sus funciones tecnológicas

(Teixeira, 2012) y, de hecho, autores llegan a afirmar que el FLOSS está presente prácticamente en todo producto de software, debido a la alta popularidad de las librerías que tienen licencias compatibles para ser mezcladas en productos cerrados (Bordeleau *et al.*, 2019).

Y no solamente los grandes líderes multinacionales globales son los que participan activamente en el mundo del FLOSS. Las comunidades y proyectos FLOSS son una fuente de código e insumos para cualquier pyme de software que de esa forma puede delinear una oferta particular de soluciones. Además, en la medida en que el OS ha permitido descomponer la cadena de valor de la producción de software, esto ha abierto a las posibilidades de modelos de negocios especializados en algunas etapas del proceso de producción, que antes solían permanecer íntegramente *in-house*. De esta forma, también se ha potenciado la incorporación del FLOSS en el negocio de las PyMEs de software.

Así, han sido expandidas las posibilidades y estrategias de negocios tanto de grandes como pequeñas empresas del sector de software. Mucho más, han ido mutando las estrategias de competencia y cooperación relacionadas a la industria. El FLOSS ha estimulado muy diversas respuestas en materia organizacional y estrategias de negocios (Dahlander y Magnusson, 2005, Harison y Koski, 2010).

Este panorama, especialmente por el grado de involucramiento de los grandes actores, sugiere que en el mundo del FLOSS se dirime, o por lo menos se disputa, el ritmo de cambio técnico y los principales patrones de innovación de las empresas de software a nivel global.

En economías periféricas, como la Argentina, aunque su participación en tal direccionamiento del cambio tecnológico de este sector sea menor, el FLOSS sí puede tener una particular importancia en términos de desarrollo (Moncaut y Robert, 2016; van Zwanenberg *et al.*, 2017). Por un lado, barre muchas de las barreras a la entrada a esta actividad facilitando los procesos de innovación. Por el carácter “abierto” de los programas informáticos que surgen de él, se torna más accesible el involucrarse en procesos de aprendizaje alrededor de ellos, disminuyendo la dependencia de conocimiento y recursos. Ese mismo carácter soluciona muchos de los problemas legales de propiedad intelectual vinculados a la “piratería”. Por otro lado, permite un gran ahorro de divisas, inmediato, por los ahorros en el pago de licencias extranjeras al uso de software privativo, pero secundariamente por su capacidad para impulsar procesos de aprendizaje que culminen en sustitución de importaciones. Durante la etapa de industrialización por sustitución de importaciones en países como el nuestro, el aprendizaje por

imitación y adaptación fueron fundamentales para el desarrollo productivo local. En la actualidad el FLOSS puede tener ese rol en la actualidad, potenciando las posibilidades de aprendizaje en la industria

Sin embargo, pese a que el FLOSS ha modificado las actividades de la cadena de valor de toda la industria del software, se ha extendido en las formas de producción e innovación tanto de grandes como pequeñas empresas y que tiene la capacidad potencial de favorecer los procesos de desarrollo en economías periféricas, la disciplina económica todavía no ha incorporado plenamente un tratamiento de sus especificidades productivas.

El origen de esta insuficiencia radica en ciertas características de la producción de FLOSS. La misma se realiza en proyectos de desarrollo de software en Comunidades de programadores, que ofrecen voluntariamente sus horas de trabajo y esfuerzo sin recibir “un pago” por ello y el software generado en estas comunidades se produce con licencias de propiedad que tienden a prohibir su apropiación privada y pueden estar disponibles para posibles competidores y diversos usuarios. Incorporar las mutaciones que el FLOSS ha desatado en las lógicas de cooperación, competencia y de regímenes de propiedad en el mercado, es un claro desafío para la disciplina económica. No hay empresa productora de software cuya actitud con respecto al uso o producción de FLOSS en su labor productiva sea neutral sobre su desempeño económico e innovativo.

Sin embargo, la economía realmente ha permanecido entre ajena e inerte en incorporar estas transformaciones como aspectos característicos del liderazgo tecnológico e innovativo en un sector *driver* de la era digital. Persisten en ella nociones, como la adherencia la atribución a la institución de los derechos de propiedad privada como condición necesaria para la existencia de innovación, a pesar que las prácticas cotidianas dentro de la industria del software vienen a poner en cuestión esta creencia. La extensión de este tipo de comportamiento productivo resulta “anormal” para la teoría económica tradicional (Robert, 2006a), y ha llevado a diseminar una visión del fenómeno como marginal en términos empíricos.

Esto ha generado vacíos y anomalías bastante importantes en la literatura y en las estadísticas. En especial, la producción del software libre y sus servicios derivados como actividad productiva dentro del sector de producción de software resulta absolutamente invisible en las estadísticas, así como no se cuenta con indicadores de su participación dentro del sector de software.

Por ello, se torna relevante un abordaje económico y productivo de la actividad del FLOSS, y este es el interés dominante que atraviesa este libro. El mismo es la culminación del proyecto de investigación PICT 2015-2703 “*Procesos de innovación en empresas de software libre y open source en Argentina*”, dirigido por el Dr. Hernán Morero, pero forma parte además de una investigación de mayor alcance alrededor de las particularidades económicas del FLOSS que comprende varios proyectos de investigación del equipo.⁹ En este momento, el libro procura constituir un aporte para el abordaje académico del FLOSS desde la perspectiva económico-productiva actualmente ausente en la literatura en castellano. Para ello avanzaremos en cuatro bloques temáticos (Figura 1), desarrollados en los 4 capítulos siguientes.

Figura 1. Bloques temáticos para la consideración económico-productiva del FLOSS



Fuente: elaboración propia.

El primer capítulo del libro tiene un carácter transversal, dado que ofrece los cimientos explicativos básicos que son aplicados y usados en los capítulos siguientes. Así, en el Capítulo 1, “*Conceptos esenciales para el abordaje económico y productivo del FLOSS*”, Hernán Morero reúne los aspectos conceptuales

9- Así, deben señalarse el Proyecto Consolidar Secyt-UNC 2018-2021 “*Procesos y tipos de innovación en industria y servicios: software libre, comercio electrónico y tecnologías 4.0 en empresas de Argentina*”, dirigido por el Dr. Jorge Motta; y el Proyecto UNRAF 2018-2019 “*Taxonomía de empresas de software según uso y producción de software libre y open source*”, dirigido por la Lic. Ana Valentina Fernández.

elementales para el abordaje académico de la actividad del FLOSS desde un punto de vista económico-productivo, desde diversas ramas disciplinarias que lo han estudiado a nivel empresarial. Asimismo, ofrecemos un estado del arte en la materia, que organiza y sistematiza en qué aristas ha estado concentrada, tanto la literatura económica, como otras clases de estudios, al investigar sobre la producción del FLOSS en las últimas décadas. Finalmente, en este capítulo ofrecemos los resultados parciales de la “*III Encuesta sobre Innovación en el sector de Software de la Argentina 2019*”, que responde a un relevamiento ejecutado a nivel nacional en base a un diseño de instrumento de recopilación estadístico diseñado por el Grupo de Economía de la Innovación y el Desarrollo (FCE-UNC) e Investigadores del Programa CTS del CIECS (Conicet y UNC) para la captación y diferenciación de la importancia cuantitativa de la actividad del FLOSS en el sector.

Las comunidades FLOSS, ese vasto océano de desarrollo sobre el que navegan y del que se nutren, pequeñas y grandes empresas de software, no ha sido plenamente incorporado por la economía como un espacio relevante de actividad productiva y económica. Es el siguiente paso del libro, en el Capítulo 2, “*La producción en comunidades FLOSS: empresas, fundaciones y governance*”, donde Juan Gabriel Vélez y Agustín Zanotti presentan en qué consisten estos actores productivos, cuáles son las formas en que organizan la producción de software, los Proyectos FLOSS, cómo pueden surgir estos proyectos, cómo se estilizan sus estructuras organizativas, la manera en que las empresas influyen en su gobernanza y los roles y funciones que tienen Organizaciones No Gubernamentales, tales como las Fundaciones FLOSS. Estas nociones son ilustradas en el funcionamiento y descripción de tres comunidades de desarrollo concreto: Fedora, Gnome, Python y LibreOffice.

En el Capítulo 3, titulado “*Grandes corporaciones de software y el FLOSS: cooperar en entornos abiertos como estrategia de competencia*”, nos adentramos en la interrelación entre los grandes líderes del sector y el mundo del open source. Allí, Carina Borrastero e Ignacio Juncos, muestran las maneras en que una serie de grandes actores privados de la industria global del software (Microsoft, Google, IBM/Red Hat, Oracle, SAP, Canonical y Tencent) mantienen esquemas de relacionamiento con la comunidad FLOSS y participan en numerosísimos proyectos colectivos de desarrollo de software. Este espacio de las Comunidades aparece como un terreno donde operan particulares y novedosas relaciones de cooperación y competencia: son casos de “*competencia abierta*” (*open coopetition*).

Pero no solamente las grandes empresas se nutren del FLOSS, en el Capítulo 4, titulado “*Las pymes de software y el FLOSS*”, Denise Gutiérrez Montecino y Florencia Manzo ofrecen una sistematización de estudios de casos de empresas argentinas de software que hemos realizado en nuestro equipo de investigación en los últimos años. Allí se puede apreciar la manera en que estos mismos espacios de colaboración donde convergen líderes del sector, que son las Comunidades FLOSS, brindan oportunidades de montar ofertas productivas a pymes del sector de software. Estos esquemas se dan a través de tipos diversos de relacionamiento con las comunidades, y contamos allí con experiencias de pymes de las Provincias de Córdoba, Buenos Aires y Santa Fe, que han articulado sus estrategias de negocios en complementariedad con el FLOSS.

Por último, este recorrido nos permite, al cierre del libro, arribar a una serie de reflexiones finales sobre la presencia empresarial alrededor de la actividad del FLOSS, pero especialmente la medida en que esta forma de producción ha alterado a la industria del software por completo. Ha transformado tanto las maneras en que se compete, rivaliza y coopera en el sector, como las formas en que se produce, pero sobre todo como se innova. Asimismo allí fundamentamos las razones por las que la economía ortodoxa, el *mainstream*, tiene limitaciones insalvables para comprender, dar cuenta y poder abordar analíticamente esta serie de transformaciones. Un libro como éste, se torna un aporte seriamente necesario para la investigación académica en economía.

Capítulo 1. Conceptos esenciales para el abordaje económico y productivo del FLOSS

Hernán Alejandro Morero

Para adentrarse al estudio y la investigación de la actividad de producción de FLOSS es necesario partir de algunas definiciones y precisiones básicas. Éstos van desde la propia distinción de lo que es un producto de software respecto de un servicio informático, cómo en ello se distingue software de tipo privativo y de tipo open source, hasta las maneras en que el FLOSS ha abierto los espacios y alcances de los modelos de negocios de las empresas, descomponiendo las actividades de la cadena de valor de producción informática. Presentar estos conceptos y definiciones básicas es el objeto y dedicación de la sección 1.1 de este capítulo. Estos conceptos tienen, además, una utilidad transversal a todo el libro, definiendo conceptos que luego serán utilizados y referidos en todos los capítulos, cuándo no ampliados en algunos de ellos.

También es interés aquí ofrecer un punto de partida de estado del arte para insertarse en la literatura de estudios del open source. Aunque la economía no ha incorporado las particularidades del FLOSS en su conceptualización de la actividad empresarial, sí pueden identificarse aportes desde otras ramas disciplinarias que lo han estudiado. La sección 1.1 procura sistematizar una revisión de cuáles han sido los principales intereses, tópicos e investigaciones empíricas sobre el fenómeno del FLOSS desde distintos campos (la sociología, el *management*, las ciencias informáticas y también interdisciplinariamente, la economía) que confluyen en lo que podemos denominar Estudios del OSS.

Por último, en este capítulo ofrecemos evidencia cuantitativa disponible sobre la importancia y participación del FLOSS en la industria del software. En la sección 2.1, en base a información secundaria, ofrecemos indicadores sobre la importancia del OS en empresas del sector en diversos países y algunos datos sobre el crecimiento en la cantidad de proyectos FLOSS y líneas de código escritas, a

nivel global. Para visualizar y visibilizar la actividad del FLOSS en el sector de software de la Argentina contamos con una fuente de datos primaria.

Nuestro equipo de investigación como parte de distintos proyectos de investigación alrededor de la medición de la innovación en la actividad del FLOSS, está llevando adelante desde fines de 2018 y principios de 2019, una tercera onda de estos dos antecedentes, que trata de cubrir las limitaciones de los anteriores. Esta tercera onda se está relevando a nivel nacional e incorpora un diseño que amplía y completa el abordaje del FLOSS en el sector del software. Como parte de las actividades de investigación del equipo entre 2018 y 2019, ejecutamos una encuesta a nivel nacional, que considera las particularidades del FLOSS y permite su identificación y distinción en el sector de software. En base a este relevamiento, denominado “*III Encuesta sobre Innovación en el sector de Software de la Argentina 2019*”, en la sección 2.2 presentamos estadísticas nacionales que permiten ver la participación del OS en la producción de software y el uso de OS.

1. El abordaje económico-productivo del FLOSS: los aspectos conceptuales y los avances de la literatura

En esta sección presentaremos una conceptualización sobre el abordaje económico de la producción de FLOSS y su contextualización en la industria del software. Esto nos lleva a desarrollar dos puntos: a) ¿Qué es y qué tiene de particular la producción de FLOSS?; y b) ¿En qué tópicos y de qué manera se han preocupado, tanto la economía como otras disciplinas, de la producción de FLOSS?

1.1. Características económicas de la producción de FLOSS a nivel empresarial

Un punto de partida para comprender lo que implica que el software sea de código abierto y qué efectos tiene ello sobre la manera en que se produce, está en la misma definición de lo que es el software en sí. Un software es un programa informático, pero como actividad productiva es algo más que la sola escritura de código, involucra toda una serie de actividades “inmateriales” a su alrededor. Por ello, es que una de las dificultades analíticas del estudio de este sector ha estado en los difusos límites entre lo que constituye un producto y lo que hace a un servicio informático, y en buena medida no es una cuestión totalmente zanjada en la literatura.

Buscando una alternativa operativa se puede considerar al software como un producto y definirlo como la licencia de un programa informático o parte de él, que es necesaria para su uso. La licencia puede ser única o duplicable. Cuando la licencia es única, se trata de un producto a medida. Cuando esta puede ser

duplicada todo lo deseado, se trata de un producto estandarizado o enlatado. Por su parte, los servicios informáticos son las actividades ofrecidas *alrededor de estos productos*, alrededor de un software determinado: desde la provisión de diversas actividades de consultorías, la propia implementación, el soporte o la capacitación sobre el uso de un software.

Estas dos distinciones, permiten configurar una matriz de productos y servicios (UNU MERIT y Berlecon Research, 2002), como grafica la Figura 1: por un lado quedan definidos productos estandarizados, servicios estandarizados y, por otro, a la provisión de soluciones, comprendiendo tanto el desarrollo de software a medida, que involucra conjuntamente productos y actividades de servicios, como a la provisión de servicios informáticos personalizados.

Figura 1.1. El software como producto y como servicio

	Productos	Servicios
Estandarizados o empaquetados	Software empaquetado	Servicios estandarizados
A medida	Soluciones (software a medida y servicios individuales)	

Fuente: UNU MERIT y Berlecon Research (2002)

Estas definiciones iniciales nos permiten avanzar en dos aspectos que son: i) entonces ¿cuándo un software es un FLOSS? ¿qué lo distingue?; y ii) ¿cómo estos bordes difusos entre servicios y productos informáticos se reflejan en la actividad productiva y económica de las empresas? ¿en qué afecta sobre ella el FLOSS?

Con respecto al primer aspecto, tomemos el software como un producto, tal y como lo hemos definido, como un programa informático con una licencia. Las licencias establecen términos y condiciones entre el que hace un software y quienes acceden a él, y pueden ser de distinto tipo (en el **Recuadro 1**, ofrecemos una breve clasificación de los principales tipos de licencias de software). Pueden ser cerradas/propietarias o incluso abiertas, que contrariamente a restringir el uso del software y su código, obliguen a quien lo utilice, que mantenga de modo público derivaciones que pueda obtener al cambiarlo y usarlo como insumo en nuevos desarrollos. Es el caso de muchas licencias que caracterizan al FLOSS.

Recuadro 1. Las licencias de software

Un programa informático puede adoptar distintos tipos de licencias.

i) Licencias propietarias / de código cerrado. Son aquellas que se especifican para evitar el acceso al código fuente del software. No permiten que el software sea modificado, desensamblado, copiado o distribuido de formas no planteadas en la propia licencia. Restringe el número de copias que pueden ser instaladas e incluso los fines para los cuales puede ser usado. Pueden incluirse aquí todo lo que son acuerdos de licencia de usuario final o *end-user license agreement* (EULA).

ii) Software de dominio público. Es cuando se publica un programa informático cediendo todos sus derechos patrimoniales y éste está disponible para cualquier ciudadano. Esto implica que cualquiera puede obtener todo su código, modificarlo e incluso publicar sus modificaciones bajo una licencia diferente

iii) Licencias de código abierto o FLOSS. Son aquellas que establecen ciertas condiciones para la visualización del código fuente de un programa informático y se distinguen por aplicar en diversas medidas el principio del *copyleft*. Las variaciones que existen entre ellas son prácticamente inabordables, pero a grandes rasgos pueden distinguirse aquellas que son *recíprocas* de las no recíprocas. **a) Licencias FLOSS recíprocas:** Éstas imponen que derivaciones posteriores de un software deban ser licenciadas bajo la misma licencia original del programa (tales como las licencias GNU-GPL). **b) Licencias FLOSS no recíprocas:** permiten que desarrollos posteriores que usan un código determinado usen otras licencias de la original, incluso de tipo propietarias, como por ejemplo la licencia BSD. Este tipo de licencias establecen otro tipo de cláusulas tales como la necesidad de dar créditos de atribución, protección de marca o proteger derechos previos de patentes.

iv) Licencias de contenido. Son aquellas que se aplican al contenido, como por ejemplo la documentación de un programa informático, y no al código del mismo. Son ejemplos de este tipo de licencias las creative commons o las GNU Free Document Licenses (GFDL).

Hay que distinguir las licencias de lo que son estrategias de licenciamiento. Por ejemplo, se suele decir que un software tiene una licencia dual. El **licenciamiento dual** no es específicamente una licencia, sino que es cuando el desarrollador de un programa lo ofrece bajo distintas licencias/condiciones para distintos grupos de usuarios. Así, provee con una licencia FLOSS a la comunidad un software, y la misma aplicación la provee con una licencia comercial a otros usuarios. Algo similar debe decirse sobre la práctica del **shareware**. Ésta se trata en realidad de una modalidad de distribución de los programas, donde el usuario puede acceder a modo de prueba a una serie de funcionalidades del software (o por un plazo determinado) y para acceder al uso completo debe pagar una licencia. El **freeware** es un software gratuito, pero que mantiene su licencia privativa, y su código fuente no es accesible, ni se puede modificar.

Como producto, un **programa informático se define como Software Libre** si sus usuarios tienen la libertad de ejecutar, estudiar, modificar y mejorar, copiar, y distribuir el producto (las 4 libertades del software libre), para lo cual su código debe estar abierto. Esta libertad se garantiza estableciendo con el software una forma particular de licencia, conocida como *copyleft*, que establece lo contrario de los derechos de propiedad privada: es un tipo de licencia sobre un software que obliga al que lo use, o lo modifique, a mantener libre y en la esfera pública el código que pudiera producir a partir de él. Esto mantiene el software protegido de ser apropiado por particulares, y garantiza que los desarrollos que se hagan a partir de él, también lo sean.

Un **programa es Open Source** (de código abierto) cuando el código fuente está disponible con sus versiones ejecutables. Para ser considerado un software libre además debe: i) estar disponible en la esfera pública; y ii) respetar las cuatro libertades básicas mencionadas. En gran medida la diferencia entre las corrientes de *Open source* y de Software Libre es “filosófica”. Desde un punto de vista operativo, a nivel productivo en empresas y en términos de su impacto económico, los términos pueden usarse indistintamente, o bien conjuntamente, como **FLOSS**. De este modo, un programa informático es FLOSS cuando es un Software Libre o es un Software Open Source.

Estas condiciones no implican necesariamente de que el software desarrollado deba ser gratuito.¹ Podemos distinguir de este modo, según la libertad de acceso al código fuente (su apertura) y el pago necesario para que el usuario acceda al programa, unas cuatro formas de software (UNU MERIT y Berlecon Research, 2002), como detalla la Figura 2. El software propietario o privativo, donde el código fuente no está disponible con el producto, sino que es distribuido de forma binaria, y no es gratuito, sino que se distribuye de modo comercial. También el código es cerrado para el *shareware* (un software gratis, pero de funcionalidades limitadas y código cerrado) y el *freeware* (un software que es gratuito, pero que no permite acceder a su código) y si bien ambos comparten que su distribución es gratuita, en el primer caso éste carácter está limitado a un período inicial, mientras que por el segundo no se cobra derechos de licencia en absoluto (por lo menos para la versión *freeware*). Por su parte tenemos dos tipos de FLOSS según su gratuidad monetaria, donde el producto se distribuye junto con el código

1- De hecho, es la razón por la que se incluye el término Libre dentro del acrónimo FLOSS, para evitar el carácter polisémico del término *Free* en inglés.

fuelle en ambos: el FLOSS comercial (que no es gratuito e involucra un pago) y el FLOSS no comercial (cuando es gratuito).

Figura 1.2. Formas de software según gratuidad y apertura del código

		Apertura del código fuente	
		Sí	No
Precio para el usuario	Gratuito	FLOSS no comercial	Freeware Shareware
	Pago	FLOSS comercial	Software privativo comercial

Desde el punto de vista económico, el aspecto más distintivo del FLOSS es su forma de producción. A diferencia de la forma “típica” de producción de software (privativo) – dentro de una empresa donde ésta forma un equipo de personal interno (programadores, testers, project managers, analistas funcionales, etc.) – para llevar adelante un proyecto de desarrollo de un programa informático, o de una funcionalidad, etc.; aquí la producción se realiza tanto *in-house*, como públicamente en proyectos colectivos de la comunidad. Ésta comunidad, donde los proyectos de desarrollo se realizan con programadores que se ofrecen voluntariamente (o como parte de su trabajo en empresas) a programar para el proyecto de desarrollo, se denomina **Comunidad FLOSS**.² Los **proyectos FLOSS** son el modo de llevar a cabo la producción en cada comunidad y desarrollar programas informáticos de forma colaborativa y colectiva, según diversos tipos de arreglos y acuerdos institucionales.

En los proyectos FLOSS participan tanto desarrolladores individuales por su cuenta, como académicos, organismos y dependencias públicas, como trabajadores de empresas. Las maneras en que las empresas, tanto grandes corporaciones, como pymes de software, participan de esta forma de producción son muy diversas y activas. Miles de nuevos software y servicios informáticos se producen a diario de esta forma sin intermediación de derechos de propiedad privada que resulte relevante para los procesos de desarrollo e innovación empresarial.

2- O Comunidad de Software Libre, o Comunidad Open Source.

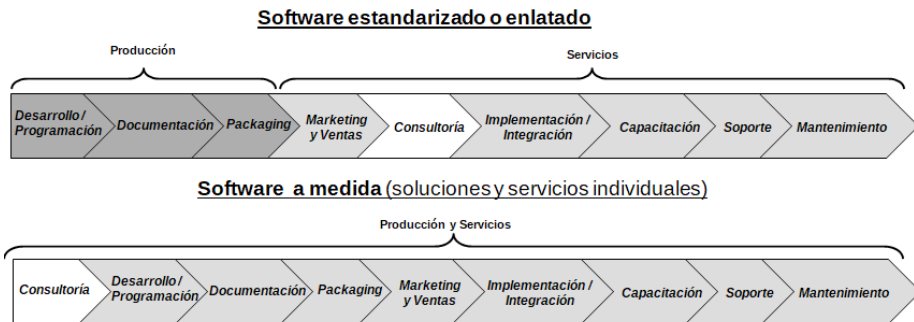
Esto nos lleva al segundo aspecto a abordar en esta sección. ¿Cómo es posible, entonces, que las empresas se involucren en actividades de desarrollo colaborativo, innoven y obtengan ingresos económicos en esta forma de producción donde los derechos de propiedad privada están ausentes? Las formas en que lo hacen son variadas, pero podemos agruparlas en dos formas de obtención de ingresos: **1) Pueden obtener ingresos de la venta y distribución de software de código abierto**, tanto si éste fue desarrollado por terceros (otra empresa, una comunidad de programadores, una dependencia pública o académica) como si fue desarrollado internamente; tanto si éste se trata de un producto enlatado o a medida. **2) Puede proveer una amplia diversidad de servicios alrededor de un software FLOS**, también tanto si éste ha sido desarrollado internamente como si ha sido creado por terceros (incluso un “competidor”).

Muchos de los servicios y actividades en las “etapas del proceso productivo” del software son similares si se realizan a partir de software privativo como si se realizan a partir de FLOSS. Pero no todas, ni en su mayoría.

Siguiendo una estilización de la cadena de valor de la producción de software en etapas productivas (ver figura 3) desarrollada por investigadores de la UNU-MERIT pueden distinguirse las siguientes actividades (que pueden o no realizarse dentro de una misma organización): el **desarrollo de software**, la **confección de documentación**, el **embalaje o *packaging***, el **marketing y ventas**, la provisión de servicios de **consultoría**, la **implementación e integración** de software; **capacitación** en el uso del software, el **soporte** y la **gestión de la aplicación o**

mantenimiento. 3 4 5 6 7 8 9 10

Figura 1.3. Cadena de Valor de la producción de software



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

3- Esta 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. 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 Berlin en octubre de 2001 o la LinuxWorld Conference de noviembre 2001 en Frankfurt.

4- Se trata de la propia escritura del código técnico, incluyendo la producción de herramientas de desarrollo.

5- Es la agregación y edición de información relacionada con la aplicación y uso de los productos de software. Pueden mencionarse 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.

6- Que es la agregación, integración y optimización de los productos de software o sus componentes (pudiendo incluir la corrección de errores).

7- Que va desde el análisis de situación del cliente, la definición de requerimientos (funcionales y técnicos) y hasta la selección de soluciones de software acorde a ello.

8- Es la instalación, configuración y customización del producto.

9- Que incluye todas las acciones tendientes a ayudar al cliente a solucionar distintos problemas que vayan surgiendo en el uso.

10- Son 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

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 alrededor de software propietario o de FLOSS. La actividad de desarrollo de software suele presentar características diferenciadas entre una forma de organizativa “más jerárquica” (siguiendo el principio de la “catedral”), o una más “horizontal y dispersa” (bajo el principio del “bazar”), aunque la realidad tiende a mostrar predominantemente la existencia de modelos híbridos y no es posible ser tajante en ese sentido.¹¹

El resto de las actividades de la “cadena de valor” sí presentan características muy diferenciadas respecto si presan vinculados a un FLOSS o si se realizan alrededor de software privativo. La Tabla 1 muestra y resume estas diferencias. La irrupción del FLOSS en la cadena de producción de software torna distintas las actividades de producción de documentación, el empaquetado, la distribución y marketing y las actividades de soporte; respecto a cuando estas mismas actividades se prestan vinculadas a un software privativo.

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, lo que la oferta de software exclusivamente privativo no habilita. El hecho de que, por un lado, una firma puede especializarse en una o algunas de estas actividades (y allí las combinaciones ya son muy variadas), pero además, de que las empresas puedan ofrecer algunas de estas actividades sólo a partir de FLOSS, otras sólo a partir de software privativo, y otras alrededor de ambos tipos de software; ha llevado que todo un conjunto de estudios y esfuerzos hayan fracasado en ofrecer una caracterización comprensiva de todos los modelos

11- Los principios del bazar y de la catedral son dos metáforas introducidas en un famoso ensayo por Eric Raymond, figura líder del Movimiento OS. La idea en el ensayo de Raymond (1999) caracteriza a la forma de desarrollo de software según un principio, que es el de la catedral, como un esquema verticalista asociado a la producción de software comercial (por lo menos hasta fines de la década de los 1990's); mientras que en un extremo opuesto se encuentra el software producido dentro del mundo Linux, con una dinámica más horizontal, equiparada al funcionamiento de un bazar. Contrariamente la construcción detallada, planificada y sistemática de los grandes sistemas de gran envergadura (como la edificación de una catedral), el desarrollo en el mundo Linux “se asemeja más a un bullicioso bazar de Babel, colmado de individuos con agendas y enfoques dispares (...), de donde puede surgir un sistema estable y coherente (...)” (Raymond, 1999).

de negocios a partir del FLOSS posibles.¹²

Una de las conclusiones del libro de Lerner y Schankerman (2013) “*The comingled code: Open source and economic development*”, es que las empresas de software mezclan actividades de desarrollo OS con software propietario en proporciones muy variables entre distintos tipos de empresas (en las que el OS llega en algunos segmentos, como el de medianas empresas, a representar el 53% de las firmas de software).¹³ También Bonaccorsi *et al.* (2006) muestran la prevalencia casi absoluta de modelos de negocio “híbridos” en las pymes de software, que mezclan software privativo y OS en su oferta productiva.

Por ello, cabe volver a señalar que los intentos de clasificar las empresas de software según la medida en que incluyen el FLOSS en su modelo de negocio han sido insatisfactorios para proveer una tipología comprensiva del sector.¹⁴ Han logrado, sí, caracterizar algunos casos emblemáticos, como el de las empresas que proveen versiones originales de sistemas operativos Linux (p.e. Canonical con Ubuntu o RedHat), o algunas pymes especializadas en brindar soporte, capacitación e implementación alrededor de un FLOSS de nicho, pero realmente las formas de establecer una oferta productiva en software son sumamente variables.

12- Ver, por ejemplo, las limitaciones señaladas en Morero *et al.* (2017a)H.A.</author><author>Motta, J.</author><author>Ortiz, P.</author><author>Vélez, J.G.</author></authors></contributors><titles><title>Pequeñas empresas de software libre (FLOSS y en Morero *et al.* (2017b)

13- Y nos referimos a una muestra que roza las 1900 empresas de unos 15 países en el mundo.

14- Entre ellos cabe listar una serie de trabajos: Hecker (1999), UNU MERIT y Berlecon Research (2002), Lerner y Schankerman (2013), Hauge *et al.* (2008) y Seppänen *et al.* (2007)

**Tabla 1.1. Actividades de la cadena de valor de producción de software.
Diferencias FLOSS y Software Propietario**

<i>Actividad</i>	<i>Producción / Programación</i>		
	<i>Desarrollo de Software</i>	<i>Documentación</i>	<i>Packaging</i>
<i>Provisión al proveedor de Software Propietario</i>	Los modelos pueden combinar modelos verticalistas u horizontales en ambos casos.	El software es usualmente vendido con la documentación, por lo que esta etapa suele integrar uno de los últimos pasos del proceso de desarrollo dentro de la empresa.	El software se vende empaquetado y es parte de uno de los últimos pasos del proceso de desarrollo.
	- La mayor diferencia entre ambos es la apertura del código fuente		
<i>Provisión al proveedor de un FLOSS</i>	Los FLOSS normalmente no son un producto finalizado, y la documentación no suele ser suficiente para los usuarios.	Este es una parte importante de muchos modelos de negocio basados en FLOSS que reúnen, complian y agregan el software, lo optimizan y lo venden como la agregación de los componentes, sus cambios y Docure porque quien desarrolla el software FLOS (p.e. una Comunidad OS) no es necesariamente quien lo promueve, quien le hace el marketing y quien lo vende y distribuye, porque estas	Diversas 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 tanto la distribución como la agregación de los componentes, sus cambios y Docure porque quien desarrolla el software FLOS (p.e. una Comunidad OS) no es necesariamente quien lo promueve, quien le hace el marketing y quien lo vende y distribuye, porque estas
	Es por eso que la producción de documentación junto con las ventas es una actividad común de las empresas que proveen FLOSS, tanto haya sido desarrollado por		

<p><i>Proveedor alrededor de un FLOSS</i></p>	<p>Como esto no es aceptable para todos los tipos de usuarios, surgen ofertas de soporte especializado, por parte de distribuidoras y empresas OSS independientes (p.e. soporte corporativo)</p>
<p><i>Proveedor Software Proprietario</i></p>	<p>Generalmente es ofrecido por una empresa especializada o el propio desarrollador del software. Suele hacer una distinción entre usuarios privados (asistencia simple, ayudas en los programas y telefónica) y corporativos (una ayuda más personalizada, con soporte de administradores).</p> <p>El soporte es ofrecido, en primer lugar, por los foros de la comunidad.</p>

No presenta diferencias significativas

No presenta diferencias significativas

Es claro, sin embargo, que existen maneras altamente heterogéneas a través de las cuales las empresas obtienen ingresos de la incorporación del FLOSS en sus actividades, y que ello tiene un impacto económico y productivo importante en el sector del software. En la Sección 2 mostraremos la dimensión de este impacto a través de distintas estadísticas disponibles, tanto de forma secundaria como primaria. Pero antes de ello, nos preguntamos ¿en qué aristas ha estado concentrada, tanto la literatura económica, como otras clases de estudios, en investigar sobre la producción del FLOSS? La subsección siguiente procura ser un mapa a los diversos focos de estudios y avances que ha habido en la investigación del FLOSS en las últimas décadas.

1.2. La investigación económico-productiva sobre el FLOSS: un resumen de la literatura

El estudio del FLOSS en su faceta económica y de producción, es un campo donde han convergido disciplinas desde puramente informáticas, hasta sociales e ingenieriles. Así lo reflejan las más de 14 *Open Source Systems Conferences* que desde 2006 ha realizado a lo largo del mundo el IFIP Working Group 2.13 on Open Source Software de la *International Federation for Information Processing*.¹⁵

La literatura puramente económica en relación al estudio del FLOSS ha estado cercenada a pocos tópicos. Por lo general, se ha tratado al FLOSS como un fenómeno anómalo y excepcional, difícil de comprender para la economía. Por ello, el principal tópico de investigación han sido las **motivaciones para involucrarse en esta actividad** (Robert, 2006b). De modo abrumador se han estudiado las motivaciones de los programadores individuales a “trabajar sin un pago directo” como parte de los proyectos OS de la comunidad, y en menor medida, las motivaciones de las empresas a colaborar en la comunidad FLOSS, aportando trabajadores, su tiempo de trabajo y recursos económicos a ello. Su preocupación central ha estado en cómo es posible esta “anomalía” de que distintos tipos de participantes renuncien a la posibilidad de lucrar económicamente través del monopolio que otorgan los derechos de propiedad intelectual de su creación y ofrecen, por el contrario, el producto de su trabajo como de propiedad colectiva.

Estos estudios han avanzado en explorar cuales son las motivaciones individuales para participar, listando: razones reputacionales (Dalle y Jullien, 2003, Lakhani y von Hippel, 2003, Lerner y Tirole, 2000), el aprendizaje y

15- <http://ifipwg213.org/>

mejoramiento de capacidades de programación (Lakhani *et al.*, 2002, Lakhani y von Hippel, 2003, Von Krogh *et al.*, 2003), la búsqueda de conectar necesidades sociales con funcionalidades de un software (Hars y Ou, 2001) , y el propio esparcimiento (Lakhani y von Hippel, 2003).

Por su parte, la investigación de las motivaciones de las empresas ha avanzado en clasificarlas en tres tipos de motivaciones: sociales, tecnológicas y económicas. Algunas investigaciones han hallado una preeminencia de éstas dos últimas sobre las motivaciones sociales (Bonaccorsi y Rossi, 2006, Bonaccorsi y Rossi Lamastra, 2004), mientras otras han encontrado una complementariedad entre los distintos tipos de motivaciones (Stam y van Wendel de Joode, 2007).

Ha habido diversos enfoques para conceptualizar la existencia del FLOSS como un bien público o un bien común. Con todo son aportes teóricos motivados a por qué el FLOSS existe, y no al cómo es, y a su importancia en la estructura económica. La economía sigue sin dar(se) cuenta de la relevancia y participación del FLOSS en el sistema de producción capitalista actual.

Luego, hay toda una serie de avances en la comprensión del funcionamiento económico, productivo e innovativo del FLOSS, que no ha provenido puramente de la economía, sino de un ámbito de lo que podríamos llamar *Estudios del OSS*, compuesto por distintos campos, que van desde la sociología, el *management*, a las ciencias informáticas y, también, interdisciplinariamente, la economía). Más allá de algunas cuestiones que tienen que ver con los aspectos de las características del licenciamiento OS o demostrar la superioridad técnica de los software OS sobre los privativos, a estos estudios debemos los avances en ciencias sociales a lo que es la comprensión económica del FLOSS, que sistematizamos en los siguientes cinco apartados temáticos.^{16 17 18}

16- Muchos estudios destacan como una de las más importantes innovaciones institucionales que ha introducido el FLOSS en el mundo del software ha sido la cuestión del licenciamiento y la propiedad intelectual. Por ello, diversos artículos (Mustonen, 2003, Skidmore, 2007)) exploran, describen y clasifican las licencias OS; y cuestionan directamente el enfoque tradicional de “fallas de mercado” para la innovación (Rossi, 2006).

17- La rivalidad entre el FLOSS y el software propietario ha sido trabajada como productos “competidores”, tanto en términos de calidad del software como de dinámica de competencia tecnológica. Un ejemplo paradigmático son las comparaciones entre Linux y Windows (Dalle y Jullien, 2000, Dalle y Jullien, 2003).

18- Priorizamos en estos cinco ítems las publicaciones, dando una menor importancia a las

a) Funcionamiento económico del FLOSS en general.

Hay un conjunto de libros de gran envergadura que luego de introducirnos a la historia del FLOSS, procuran dar una perspectiva económica global del fenómeno, desde su relación con el crecimiento y el desarrollo, sus dimensiones de oferta y demanda y cuestiones de política pública.

En este grupo deben destacarse el libro de Lerner y Schankerman (2013), *“The Comingled Code: Open Source and Economic Development”*, el manual de Amant y Still (2007) *“Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives”*, el libro de Weber (2004), *“The success of open source”*, y el de Bitzer y Schröder (2006), *The Economics of Open Source Software Development. Éstos constituyen, junto con los trabajos seminales de Richard Stallman y de Eric Raymond, puntos de partida para la investigación y comprensión económica del FLOSS* (Stallman, 1983, Raymond, 1999).

b) La forma de producción de los proyectos de desarrollo colectivo de FLOSS como un proceso de innovación.

La investigación ha logrado esquematizar de modo bastante adecuado el proceso productivo de los proyectos de desarrollo de FLOSS.¹⁹ Los mismos funcionan como un sistema de innovación distribuida caracterizado por:

- La ausencia de una unidad centralizada de toma de decisiones *ex-ante* y la existencia de un diseño y corrección del código concurrente. Esto es facilitado por la modularidad de los software y su código fuente, que permite que distintos grupos de desarrollo esté trabajando simultáneamente en distintos componentes que pueden ser ensamblados para lograr funcionalidades altamente complejas (Bonaccorsi y Rossi, 2003). Esto permite que el proceso de desarrollo se lleve adelante de modo concurrente y explotar una inteligencia colectiva distribuida, con resultados innovativos superiores a los esquemas de *software factory* tradicionales (Kogut y Metiu, 2001).

presentaciones en reuniones científicas. Sin embargo, a continuación, la complementamos con los aportes de dos importantes eventos académicos, que son las *OSS Conferences* y las Jornadas de Software Libre de la JAIIO.

19- Nos concentramos a nivel de proyectos colectivos FLOSS, pues también hay una serie de estudios emparentados con la ingeniería de software que están más emparentados en como son los procesos de desarrollo. Un listado de estos estudios puede consultarse en Aksulu y Wade (2010), aunque de allí podemos destacar a Baldwin y Clark (2006), Koch y Gonzalez-Barahona (2005) y MacCormack *et al.* (2006).

- La integración de los propios usuarios en el proceso productivo, creativo, de desarrollo, corrección y mejoras al software (Hippel y Krogh, 2003).

- La producción colaborativa basada en bienes comunes (*commons-based peer production*). Ésta describe un modo de producción económica donde un número importante de personas trabaja de modo cooperativo y voluntario, usualmente mediado por tecnologías digitales y la Internet (Benkler, 2017). Funcionan ante una aparente ausencia de jerarquías y se establecen comunidades de producción entre pares, que desempeñan funciones tales como la coordinación, la división del trabajo, la búsqueda y aceptación de colaboradores, la capacitación, la creación y cumplimiento de normas, resolución de conflictos y el establecimiento de metas y objetivos. En el FLOSS, a través de esta forma de organización los programadores que van a realizar diversas tareas en la producción se autoseleccionan y distribuyen entre las funciones (Rossi, 2006).

Todo esto conforma un “modelo comunal de innovación” (Lee y Cole, 2003) o un modelo de innovación “privado-colectivo” (Hippel y Krogh, 2003). La efectividad del modelo de producción de los proyectos FLOSS está en el hecho de que el resultado del proceso innovativo no es percibido completamente ni como un “bien público puro”, ni como un “recurso de uso común” (O’Mahony, 2003), sino que posee significativos beneficios que pueden ser apropiados privadamente aunque las innovaciones “se revelen” libre y gratuitamente, sin la necesidad concreta de establecer derechos de propiedad privada. De este modo, preconcepciones de la teoría económica *mainstream*, tales como el comportamiento *free rider*, no parecen tener relevancia empírica. Contrariamente a la idea tradicional que ata incentivos para innovar y propiedad privada, la liberación de una innovación puede no involucrar una pérdida de beneficios para el innovador, sino una ganancia por la difusión de la innovación y por la obtención de efectos de red. El proceso de innovación de las empresas que se involucran se ve impulsado y acelerado por la participación en lo que aparece como un pool de I+D colectiva (Dahlander y Magnusson, 2008), de modo que se reduce significativamente las inversiones requeridas para desarrollar un nuevo software.

c) El funcionamiento, formas de organización y gobernanza de las comunidades FLOSS

El carácter de un sistema de innovación distribuida de los proyectos FLOSS ha llevado a investigar la manera en que las comunidades donde se llevan a cabo

y se produce el software se organizan y funcionan. Este se trata de un tema central y crítico, pues abre un espacio de competencia y cooperación tecnológico-económica, que traspasa al mercado.

Por un lado, nos encontramos con una serie de conceptualizaciones y definiciones alrededor de las interrelaciones entre las Comunidades, sus proyectos de desarrollo y sub comunidades que los componen. Ello en ocasiones es abordado como el estudio de *Ecosistemas OS* (Poo-Caamaño *et al.*, 2016), donde proyectos y comunidades viven simbióticamente y, en ese sentido, sus evoluciones, estructuras y maneras de funcionar dependen mutuamente entre sí; o como *Constelaciones de prácticas* que están enraizadas (*embeded*) en dinámicas existentes de mercado y de flujos monetarios (Berdou, 2010). Así, diversos artículos estilizan las formas en que se inician y luego se desarrollan los proyectos FLOSS (Schaarschmidt *et al.*, 2015), y los clasifican y distinguen entre sí acorde a ello (Capra *et al.*, 2011, Capiluppi *et al.*, 2012). Aparecen así proyectos FLOSS tradicionales (donde no se involucran las empresas), proyectos FLOSS con involucramiento empresarial y proyectos FLOSS liderados por empresas (Capiluppi *et al.*, 2012). Los primeros dos son proyectos comunitarios y los últimos dos cuentan con esponsorio corporativo.

Por otro, están bien asentadas en la literatura las características que posee la estructura organizativa de los proyectos FLOSS. Éstos suelen estructurarse según lo que se denomina como modelo cebolla (*onion model*) (Nakakoji *et al.*, 2002), idealizado en dos grupos de desarrolladores: al centro y de mayor importancia el núcleo o core donde están las principales autoridades, y en sus alrededores una periferia de colaboradores y programadores (Crowston y Howison, 2005, Crowston y Howison, 2006, David y Rullani, 2008, Crowston y Shamshurin, 2017). Hay un cuerpo importante que ha logrado caracterizar cómo es y cómo se forma esta división del trabajo, el rol de los méritos y la dinámica intra-organizacional de la periferia (Rullani y Haefliger, 2013).

La forma de organización de las comunidades y sus proyectos se comprende plenamente a través de investigar su gobernanza. La gobernanza OS se define como los medios de lograr (parcial o totalmente) la dirección, control y coordinación de organizaciones o individuos autónomos para que contribuyan conjuntamente hacia un proyecto de desarrollo FLOSS (Markus, 2007). Esta gobernanza puede ser “espontánea”, recurrir a una serie de instrumentos internos más formales de gobiernos, o depender de terceras partes “externas” a la Comunidad (De Laat, 2007). La mayor parte de la literatura de gobernanza proviene de dos campos

disciplinarias de investigación: las ciencias políticas y el *management* corporativo (Schaarschmidt *et al.*, 2015). Entre los temas que se han investigado en este cuerpo pueden mencionarse (Scacchi y Jensen, 2008, Lee *et al.*, 2017): cuestiones relacionadas a cómo es la autoridad de los procesos de decisión, la asignación de recursos, el liderazgo, el control social y los mecanismos de coordinación, entre otras (Di Tullio y Staples, 2013, O'Mahony y Ferraro, 2007, Berdou, 2010, Schaarschmidt *et al.*, 2015, Shah, 2006, West y O'mahony, 2008).

d) Rol y participación de las empresas

Esta es una de las líneas de la literatura OS más relevante para el estudio que engloba este libro, por el simple hecho de que la empresa constituye el actor principal de las economías capitalistas modernas.

El primer interés dentro de este tema es, bien, ¿de qué manera y porqué se involucran las empresas en las Comunidades FLOSS? Ha surgido en respuesta a ello una diversidad de estudios describiendo o tipificando las formas de involucramiento de las empresas (Jullien y Zimmermann, 2011). Las formas van desde contar con trabajadores de la empresa pagos para colaborar en proyectos FLOSS de la comunidad (Dahlander y Wallin, 2006, Henkel, 2009, Berdou, 2010, Colombo *et al.*, 2013), la propia iniciación de proyectos FLOSS esponsorados por la empresa (Capiluppi *et al.*, 2012), la liberación de código anteriormente cerrado (Henkel, 2004), la “donación” de herramientas de desarrollo desarrolladas *inhouse*, el financiamiento y apoyo a Fundaciones que soportan al FLOSS, hasta la provisión de otro tipo de infraestructuras y recursos (hardware, ancho de banda, hosting, etc.).

Los tipos de involucramiento han sido clasificados de distintas maneras. Una clasificación que califica el grado de involucramiento según el nivel de beneficio mutuo que tienen las empresas en relación a la comunidad las tipifica en tres clases: relaciones de tipo simbiótica, de tipo comensalista y de tipo parasitaria (Dahlander y Magnusson, 2005). Otras clasificaciones distinguen a las formas de participación de las empresas en proyectos OS comunitarios (que son liderados por miembros de la comunidad y no han sido iniciados por empresas) en tres modelos, que pueden combinarse para una misma firma en distintos espacios FLOSS: un modelo de aporte de código (*coding*), un modelo de soporte y apoyo a la comunidad (*support*) y un modelo de gestión (*management*) de proyectos (Capra *et al.*, 2011). También la participación de las empresas puede distinguirse si éstas se acoplan a un proyecto iniciado por la comunidad donde hay involucramiento

de firmas (*industry-involved OSS projects*) o si directamente inician sus proyectos y los lideran (*industry-led OSS projects*) (Capiluppi *et al.*, 2012).

Un segundo interés ha estado en (de)mostrar que la participación de las empresas en el FLOSS se trata de una actividad “rentable”. Aquí es muy prolífica toda una literatura que muestra y explicita toda una serie de beneficios que tienen para las empresas actividades que van desde la liberación de software, hasta la colaboración en proyectos comunitarios y el apoyo al FLOSS. Por otro lado, hay un conjunto de investigaciones que se han empeñado en mostrar diversos “modelos de negocio” vinculados al FLOSS (Bonaccorsi *et al.*, 2006, Seppänen *et al.*, 2007, Hecker, 1999, UNU MERIT y Berlecon Research, 2002) que tienen las limitaciones que señalamos en la sección anterior: su incapacidad para dar cuenta de la heterogeneidad de combinación de actividades que habilita el FLOSS en confluencia con la hibridación predominante en las carteras de oferta del software.²⁰

Es esto último uno de los resultados más robustos e interesantes de estos esfuerzos: tanto el predominio de diversos grados de combinaciones entre software OS y privativo en la especialización de las empresas del sector (West, 2003, Bonaccorsi *et al.*, 2006, Lerner y Schankerman, 2013, Munga *et al.*, 2009), como de una amplia diversificación productiva impulsada por el OS (Colombo *et al.*, 2014), aparecen como rasgos característicos de la actividad del software actualmente a nivel global.

Por último, un tercer interés ha estado en la dinámica competitiva que ha abierto el FLOSS entre los líderes tecnológicos de la industria del software. Es lo que se ha dado en llamar *open coo-petition* (Teixeira *et al.*, 2016, Samuelson, 2006), representando el hecho de que grandes actores del sector aparecen colaborando en el ámbito del OS, por ejemplo en un gran proyecto, y rivalizando al mismo tiempo, compitiendo por la dirección tecnológica del mismo (o participando en proyectos competidores en una misma área, como pueden ser dos proyectos de software de infraestructura de computación en la nube tales como OpenStack y CloudStack). De esta manera el FLOSS ha abierto un espacio de cooperación y competencia paralelo al sistema de precios, en tanto que es factible argumentar incluso que los proyectos FLOSS maduros están directamente enraizados o

20- A ello habría que sumarse de que la idea de un “modelo de negocio” falla en capturar los efectos de derrame (*spill-over*) recibidos por una empresa al participar de un desarrollo colaborativo OS, que involucran todos los procesos de aprendizaje propios de una comunidad de práctica y los que ocurren a nivel de la organización (Berdou, 2010).

embebidos en estructuras de los mercados (Berdou, 2010). De esta forma, al dejar de lado de la disciplina económica el estudio del funcionamiento de las Comunidades de producción colaborativa como el FLOSS, el *mainstream* de la economía ha resignado el estudio de un importante componente de los mercados. Especialmente relevante en sectores que son los drivers del cambio tecnológico en las economías capitalistas actuales.

e) Rol y participación de los gobiernos: la necesidad de promoción pública del FLOSS.

Gradualmente el FLOSS ha entrado en la agenda de políticas públicas de gran cantidad de países. Con ello, ha surgido la necesidad de justificar la intervención pública, máxime porque el *mainstream* económico tiende a generar racionalizaciones de la aversión típicamente reaccionaria contra la intervención estatal y precisa argumentos para justificarla cuando se vuelve un aspecto necesario para las economías de mercado.

Así podemos mencionar tres tipos de argumentos dentro de la economía al respecto. En primer lugar, una línea tiende a justificar la acción pública por razones de eficiencia ante la existencia de “fallas de mercado” ocasionadas por las características del software como bien mercantil (Schmidt y Schnitzer, 2002). Los mercados de software poseen características tales como la existencia de efectos de red, grandes economías de escala y un rol ubicuo de la innovación, que requieren correcciones estatales para funcionar “eficientemente”.

Un segundo tipo de argumento relacionado versa en la superioridad técnica (calidad, seguridad, etc.) del FLOSS sobre el software privativo. Ello deriva en que fomentar su provisión se torne socialmente deseable, mientras los mercados por sí solos tenderían a subproducirlos, tal y como la economía ortodoxa concibe un bien público (Varian y Shapiro, 2003).

Finalmente, una línea argumentativa diferente enfatiza la importancia de promover el FLOSS para impulsar el desarrollo. Así, muchísimas investigaciones documentan y fundamentan las ventajas y beneficios de impulsar el FLOSS para economías periféricas. Las razones son variadas: elimina muchas de las barreras a la entrada a esta actividad debido a sus menores costos de ingreso (van Reijswoud y Mulo, 2007) y porque facilita los procesos de innovación dado el carácter “abierto” de los programas, soluciona muchos de los problemas legales de propiedad intelectual vinculados a la “piratería” (Dudley-Sponaugle *et al.*, 2007), reduce la dependencia tecnológica de la administración pública y los gobiernos a

un proveedor de software particular, permite un gran ahorro de divisas por el no pago de licencias extranjeras al uso de software privativo y secundariamente por su potencial para impulsar procesos de aprendizaje y sustitución de importaciones (Moncaut y Robert, 2016). En sí, porque el FLOSS ofrece elementos de autonomía e independencia tecnológica, que habilitan a la construcción de competencias domésticas.

Dentro de los aportes que no forman parte de publicaciones, sino que son presentaciones y comunicaciones en reuniones científicas especializadas, caben destacar dos espacios académicos y reseñar algunas de sus contribuciones, muchas de ellas complementarias a lo ya presentado en esta sección.

El primero de estos espacios académicos con las ponencias de las *Open Source Systems Conferences* de la IFIP.²¹ En este ambiente hay una predilección hacia la explotación cuantitativa de grandes bases de datos de proyectos FLOSS (como el uso de repositorios del código, registros de las contribuciones, del registro de versiones, listas de correo, etc.) y la aplicación de técnicas de big data y data mining para el análisis, además de estudios de casos (en su mayoría, más bien descriptivos y escasamente analíticos). A través de este tipo de metodologías, resulta relevante el trabajo que permite dimensionar la importancia del FLOSS a través de la medición de su crecimiento a partir del cómputo de la evolución de las líneas de código OS escritas y la medida en que aumenta la base global de proyectos FLOSS activos (Deshpande y Riehle, 2008, Wiggins y Crowston, 2010, Bordeleau *et al.*, 2019).²²

Respecto de la faceta económico-productiva del FLOSS, en estas conferencias pueden agruparse los artículos en 3 categorías, que han generado una serie de aportes que complementan la literatura previamente reseñada, especialmente de forma empírica en los siguientes tópicos:

- **El funcionamiento de las Comunidades FLOSS.** Aquí pueden hallarse:

- i) descripciones de la evolución de las estructuras de las comunidades y la intensidad de sus patrones de comunicación e información (Weiss *et al.*, 2006, Capiluppi y Michlmayr, 2007, Lindman, 2007, Kamei *et al.*, 2008, Crowston y Shamshurin, 2017);
- ii) Recuentos de las formas y estrategias de gobernanza que llevan adelante sus líderes de las comunidades como redes de

21- International Federation for Information Processing.

22- La sección siguiente se nutre en buena medida de esta clase de aportes.

interacción socio-técnicas (Jensen y Scacchi, 2010, Mäenpää *et al.*, 2017); y iii) Explicaciones de la importancia del involucramiento de distintas clases de organizaciones para el éxito y funcionamiento de las comunidades, tales como las empresas (Capiluppi *et al.*, 2012), las Universidades, el Gobierno y las Fundaciones (Forrest *et al.*, 2012, Lindman y Hammouda, 2017).

- **El involucramiento de empresas en la producción de FLOSS y en las comunidades OS.** En esta temática predomina la narrativa e ilustración de estudios de casos, la mayoría utilizando fuentes cuantitativas, pero complementadas con recursos cualitativos. Tres grandes aportes deben señalarse: i) Los beneficios concretos que cuentan las empresas por liberar sus desarrollos, (Lindman *et al.*, 2009, Kim *et al.*, 2012, Capra *et al.*, 2009) de usar componentes OS incluso para la producción de software privativo, de participar en las comunidades grandes y también las maneras (tales como el contar con programadores contratados para colaborar, dar infraestructura y personal para los proyectos, etc.) (Berdou, 2006, Martínez-Romo *et al.*, 2008, Lindman *et al.*, 2009, Capra *et al.*, 2009, Melian y Mähring, 2008, Teixeira, 2012); ii) Descripciones de experiencias concretas de primera mano sobre la participación e influencia de grandes corporaciones en muchos desarrollos OS (Aaltonen y Jokinen, 2007, Robles *et al.*, 2007, Jaaksi, 2007, Durand *et al.*, 2012, Lindman *et al.*, 2008) y la manera en que cooperan y compiten grandes actores de la industria del software al interior de proyectos FLOSS de gran dimensión, tales como OpenStack (Teixeira, 2017).²³ iii) Algunas experiencias de pymes productoras de FLOSS: aunque son menores, pueden contarse estudios muy relevantes sobre casos de pymes noruegas que han sido capaces de crear comunidades FLOSS alrededor de sus desarrollos y mantenerse en actividad hasta la actualidad (Hauge y Ziemer, 2009), y la descripción de varios modelos de negocios FLOSS de mipymes africanas (Sowe y McNaughton, 2012).

- **La descripción del involucramiento de otros grandes usuarios y desarrolladores de FLOSS: Administraciones Públicas y Universidades.** En este campo las ponencias son más bien descripciones de experiencias, tanto de organizaciones públicas que utilizan FLOSS y como han realizado las migraciones (Ven y Verelst, 2006, Ven *et al.*, 2006, Brink *et al.*, 2006,

23- Los casos incluyen proyectos concretos, como el propio núcleo de Linux o Debian o experiencias concretas, tales como la de Nokia, Samsung, Facebook, Google o Phillips Medical Systems.

Kuechler *et al.*, 2012, Robles *et al.*, 2016, Conklin, 2006), su uso en cuestiones infraestructurales, tales como la administración del tráfico aéreo o el establecimiento de redes de información de salud pública (Hardy y Bourgoi, 2006, Staring y Titlestad, 2006), y su implementación en el sector educativo en distintos países (Brink *et al.*, 2006, Sowe *et al.*, 2007, Petrinja y Succi, 2012). Claramente estos estudios tienen un sesgo temático hacia las cuestiones de uso e implementación.

Fuera de estas líneas temáticas, caben mencionarse algunos antecedentes sobre las disparidades de género en el sector del software y en el mundo del FLOSS (Kuechler *et al.*, 2012, Robles *et al.*, 2016). Luego, en estas conferencias hay otra serie de ponencias que se alejan de la temática económico-productiva o son más bien de índole técnico o de corte metodológico. Allí entran ponencias que evalúan y presentan diversas características de los software FLOS, de los proyectos OS y su dimensión (tales como la calidad de los programas, formas de testeo de calidad, aplicaciones concretas con distintos fines, los ciclos de vida de los proyectos y sus patrones de forking, uso de métricas para la selección de código, etc.), el análisis de prácticas y modelos de desarrollo vinculadas (como la relación con las metodologías ágiles o el uso de *sprints* como forma de desarrollo, herramientas de gestión de proyectos, otras cuestiones más técnicas como problemas de autenticación recomendaciones para la adopción de librerías; etc.); aspectos legales y de licenciamiento, la enseñanza del FLOSS. Por último hay un cuerpo de ponencias alrededor de cómo investigar al FLOSS haciendo uso de técnicas de big data y de procesamiento de lenguaje natural a través del acceso a repositorios de repositorios, describiendo técnicas para implementar y limpiar bases, qué indicadores usar sobre un proyecto (Sowe *et al.*, 2007, Conklin, 2006, Petrinja y Succi, 2012, Noll *et al.*, 2012, Ribeiro *et al.*, 2018).

El segundo espacio académico relevante de ponencias y comunicaciones especializadas ha sido un evento análogo a las *OSS Conferences*, en Argentina o las Jornadas de Software Libre (JSL) que formaron parte de las JAIIOs (Jornadas Argentinas de Informática), organizadas por la Sociedad Argentina de Informática (SADIO), entre los años 2005 y 2013. Los aportes que pueden destacarse de estas Jornadas se centran en ponencias sobre presentación de experiencias y proyectos, por lo general con escaso trabajo analítico y carácter más bien empírico.

Se destacan las presentaciones de aplicaciones para la Administración Pública, tales como la experiencia de la Municipalidad de Rosario (Giardini y Aguzzi,

2011, Boidi y Giardini, 2006), diversos casos de aplicaciones en áreas tributarias tanto a nivel nacional en AFIP (Reingart, 2012) como a nivel provincial para el caso de la DGR de Misiones (Gonella y Schelske, 2010), así como la evaluación de la implementación de OS en dependencias como el Organismo Regulador del Sistema Nacional de Aeropuertos (ORSNA) (D'Agostino, 2010) o la Dirección Nacional de Vialidad (Sirvente, 2006).

Un punto en el que ha habido avances que han presentados en las JSL son una serie de intentos incipientes en la medición del uso de FLOSS en empresas (Rubбини, 2010; Rubбини, 2009) y en la caracterización por medio de encuestas de los modelos de negocio (Castello *et al.*, 2009). Este último estudio encontró en una muestra chica de empresas argentinas FLOSS una escasa aplicabilidad de los modelos de negocios estilizados por Hecker (1999) y Raymond (1999), lo que verifica los resultados de estudios en el tema que hemos mencionado.

En síntesis, desde diversos campos disciplinares se cuenta con un buen conocimiento de cómo funcionan internamente las Comunidades FLOSS, cómo se clasifican y distinguen sus proyectos, qué motiva a sus intervinientes a participar, cómo se relaciona todo este campo con el mundo corporativo, cuáles son sus lógicas y de manera ilustrativa y anecdótica, cuál es su dimensión económica y productiva.

Maria Alessandra Rossi (2006) plantea que hay buenas razones para sostener que la complejidad de un abordaje económico del OS no está tanto en el hecho de porqué los actores contribuyen libremente a la producción de un bien que en enorme medida estará disponible “para todos”, sino en la complejidad de la estructura institucional del FLOSS y su habilidad para organizacionalmente evolucionar a lo largo del tiempo. Claramente el *mainstream* en economía no cuenta con instrumentales teórico-analíticos adecuados para abordar estas dimensiones. La ortodoxia económica sigue estancada en la “excepcionalidad” del carácter del software como un bien y mantiene una visión del FLOSS como un fenómeno anómalo, como una “falla” del mercado. Pese a ello, la participación de las empresas en el OS parece ser un aspecto bastante común del funcionamiento del mercado de software.

Aunque, en ese sentido, la teoría económica tradicional no incorpore su comportamiento como un aspecto característico de los procesos de competencia y cooperación capitalista, el FLOSS sigue avanzando a lo largo de los años en su importancia y participación en la industria del software. La sección siguiente

procura dar una idea cuantitativa de esta dimensión. Ello lo hacemos en dos pasos: primeramente en base a diversas fuentes secundarias tratamos de ofrecer diversas medidas cuantitativas de la importancia del FLOSS en la industria global del software (sección 2.1), mientras a través de un relevamiento realizado por nuestro equipo de investigación en empresas del sector de software de la Argentina, procuramos dar algunos indicios de su participación en el sector productivo de nuestro país (sección 2.2).

2. El FLOSS en la industria del Software

2.1. La importancia del FLOSS en la industria global del sector de software

Si bien es difícil la cuantificación de la dimensión del FLOSS en la producción mundial de software y servicios informáticos debido a la escasa información estadística que distingue su participación, pueden mencionarse algunos datos.

Estimaciones para inicios de la década de los 2000's sostenían que el 56% de las 2500 empresas estadounidenses más grandes utilizaban algún tipo de FLOSS, especialmente en sus servidores web (Bonaccorsi y Rossi, 2003). En base a una encuesta en 15 países (Lerner y Schankerman, 2013) hallaban que cerca del 40% de las firmas de software contribuían en proyectos FLOSS. Un relevamiento realizado en España a 1932 empresas de software (ESFA-SI 2011), revelaba que el 39% de ellas habían comercializado productos bajo licencias FLOSS en 2010 y que el 18% de sus ingresos podían atribuirse a este tipo de actividad (CENATIC, 2011).^{24 25}

En base a una encuesta realizada en 2007 a unas 570 empresas de software de Noruega, se encontró que cerca del 47% de ellas integraba componentes OS en sus productos desarrollados, que el 16% tenían empleados que colaboraban en proyectos OSS de la comunidad y un 5% contaban incluso con proyectos OSS propios (Hauge *et al.*, 2008). También en los países nórdicos, Nikula y Jantunen (2005) muestran que el 44% de las empresas de software finlandesas incorporaban OSS en su oferta de negocios. Acorde a un estudio de Gartner del

24- La encuesta se realizó en Brasil, Chile, China, Francia, Grecia, India, Israel, Kenia, México, Polonia, Rusia, Sudáfrica, Singapur, Tailandia y Turquía; y la muestra completa comprende 1894 respuestas positivas.

25- Éste fue realizado por la disuelta fundación Centro de Excelencia de Software de Fuentes Abiertas, a través de su Observatorio Nacional del Software de Fuentes Abiertas (ONSFA). A su trabajo se deben dos de los instrumentos de encuestas más destacados para identificar a la actividad empresarial FLOSS (CENATIC, 2011, CENATIC, 2009).

año 2008, alrededor del 85% de las empresas de software utilizan algún tipo de FLOSS en sus procesos de desarrollo (Zuckerfeld, 2012).

Una línea de estudios argumenta que tratar de estimar el volumen de ingresos provenientes de productos FLOSS y su participación en la industria, así como análogamente la proporción de empresas OS en el sector de software, subestima el crecimiento del FLOSS. ¿Por qué? Básicamente, porque el software OS actualmente forma parte de muchos (sino la mayoría) de productos de software propietarios de código cerrado, como insumo en sus procesos de desarrollo. Por tanto, medir los ingresos que generan los software puramente OS subestima la dimensión productiva y económica que éste tiene en la industria del software. Esta gama de estudios ha preferido medir el crecimiento total del OS a través del crecimiento en la cantidad de líneas de código y de proyectos.

A nivel de proyectos, desde la literatura de Estudios OS la evidencia empírica sobre la expansión del OS aparece como tan abrumadora, que las investigaciones se han abocado a discutir si el ritmo en que el OS crece es exponencial, cuadrático o superlineal.²⁶ Deshpande y Riehle (2008) han estudiado el carácter del crecimiento en cantidad de líneas agregadas y en cantidad de proyectos agregados mensualmente sobre una muestra de más de 5.100 proyectos OS activos y populares para un período de 10 años (1995 y 2006). Este estudio muestra que el crecimiento de la cantidad de líneas y de proyectos que se *agregan* por mes, crecen más bien de modo exponencial, de forma que la cantidad total de ambos se duplicaría en aproximadamente unos 14 meses.²⁷

Según Open Hub, a mayo de 2019, de la información disponible existían alrededor de 52.800 proyectos OS activos en el mundo con diverso nivel de intensidad en su dinamismo.²⁸ La participación de las empresas en estos proyectos

26- Por ejemplo, Succi *et al.* (2001) encuentra una tasa de crecimiento lineal para dos proyectos OS (Apache y GNU compiler compiler) y Robles *et al.* (2005) verifican que los proyectos NetBSD, FreeBSD, OpenBSD y otros 18 siguen un patrón de crecimiento lineal. Ambos estudios, junto con Godfrey y Tu (2001) confirman que el kernel de Linux crece superlinealmente. Por último, Koch (2005) sobre la base de 4047 proyectos muestra que un modelo de crecimiento cuadrático se adecua mejor a su crecimiento que uno lineal.

27- Se toman proyectos sobre cierto umbral de in-links a las webs de los proyectos. Se trata de una muestra que representa cerca de un 28% del total de proyectos OS activos a nivel mundial. A 2007, de los 150.000 proyectos registrados en sourceforge se estimaba que unos 18.000 proyectos estaban activos.

28- De un total de cerca de 218.440 proyectos OS para los que se cuenta información sobre sus niveles de actividad. Además, existen otros 246.135 proyectos de los que no se registran datos, por

de desarrollo colectivo es relevante. Capra *et al.* (2009) han encontrado sobre una muestra de 1300 proyectos OS alojados en sourceforge.net, en casi un 31% de ellos una o más empresas se han involucrado en el desarrollo de los mismos, tanto a través del apoyo a los mismos, como escribiendo código y, aunque en una menor medida, coordinando y gestionando los proyectos.²⁹

2.2. *El uso y producción de FLOSS en el sector de software de la Argentina*

En Argentina el gran crecimiento del FLOSS no ha pasado desapercibido. Ya a mediados de los 2000s, Rubbini (2009) señalaba la existencia en el país de una centena de oferentes de servicios basados en FLOSS y más de 5.000 desarrolladores especializados en esta área distribuidos en todo el país. Actualmente se evidencia 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, 2006a, Robert, 2006b) 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).

Contamos con dos antecedentes de relevamientos tecnológicos como parte de la participación en proyectos de investigación con otros equipos del resto del país, que tienen ciertas limitaciones para nuestro objetivo, pero que sirven como unos indicios pioneros. El primero de ellos brindó información sobre el uso o el desarrollo de OS para el año 2011 en empresas de software.³⁰ Así, este

lo que no es posible saber si están activos o no, totalizando la base de Open Hub a mayo de 2019 unos 494.575 proyectos OS creados globalmente.

29- Y es preciso destacar de que Capra *et al.* (2009) consideran en su trabajo solamente proyectos OS “comunitarios”. Esto es, no analizan los proyectos OS comerciales, que son liderados y lanzados directamente por una empresa comercial.

30- Esta encuesta se realizó en el marco del Proyecto de Investigación dirigido por Gabriel Yoguel (UNGS) “Capacidades de absorción y conectividad en sistemas productivos y de innovación locales” de la Fundación Carolina (id. 386317), durante 2011 en CABA, Conurbano Bonaerense, Córdoba y Santa Fe y cubre el período 2008-2010.

relevamiento señala en base a una muestra de 257 empresas del sector, 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 (Morero y Borrastero, 2015). Sin embargo, este diseño de encuesta no permite distinguir el uso, de la producción de FLOSS.

El segundo de ellos, discrimina la información sobre uso y sobre producción de programas OS. Fue una encuesta tecnológica realizada entre 2016 y 2017 alrededor de proyectos de investigación coordinados por investigadores de la UNGS, y realizada conjuntamente con investigadores de la UNC, UNMP, UNSAM, UNS, UNICEN y UTN-FRCU. Esta encuesta fue realizada a 183 empresas de software de la Argentina.³¹ Con respecto al uso de OS, los resultados son bastante contundentes: 9 de cada 10 empresas de software de la Argentina utilizaban alguna herramienta o insumo OS. Con respecto a la producción, la encuesta indagó sobre la realización de actividades de desarrollo de FLOSS como parte de su oferta. Una proporción menor con respecto al uso desarrollaba software de ese tipo: el 12% de las firmas entrevistadas desarrollaban FLOSS.

Se trató de un primer indicador de referencia, pero hay razones para considerar que este porcentaje subestima la participación del FLOSS en la actividad productiva e ingresos de las empresas de software. La razón de ello es la tremenda importancia de las actividades de servicios dentro de los modelos de negocios alrededor de FLOSS. El montaje de una oferta de servicios alrededor de un software open source, muchas veces genérico, pero libremente disponible, así como la provisión de desarrollos a medida a partir de este tipo de software o de código gratuitamente accesible en la comunidad; son prácticas bastante extendidas en modelos de negocios de pymes de software. Los relevamientos anteriores no han incluido mediciones de oferta de servicios informáticos a partir de FLOSS, solo actividades de desarrollo de productos.

Nuestro equipo de investigación como parte de distintos proyectos de investigación alrededor de la medición de la innovación en la actividad del FLOSS, está llevando adelante desde fines de 2018 y principios de 2019, una tercera onda de estos dos antecedentes, que trata de cubrir las limitaciones de

31- La encuesta se realizó de manera online a través de la plataforma LimeSurvey y fue realizada a empresas de software de CABA, Provincia de Buenos Aires (Bahía Blanca, Mar del Plata y GBA), Provincia de Córdoba, Provincia de Santa Fe y Provincia de Entre Ríos

los anteriores. Esta tercera onda se está relevando a nivel nacional e incorpora un diseño que amplía y completa el abordaje del FLOSS en el sector del software.

Este relevamiento, denominado “*III Encuesta sobre Innovación en el sector de Software de la Argentina 2019*”, tiene un diseño en cuatro módulos. Incorpora a los estándares típicos de medición de la innovación de las encuestas tecnológicas que siguen al Manual de Oslo (lo que habilita a comparabilidad, entre ondas y respecto a otras encuestas), precisiones al abordaje de la especialización productiva con distinciones según la producción y/o comercialización de FLOSS, provisión de servicios a partir de FLOSS, así como de la intensidad de las actividades de liberación de desarrollos de software, la importancia del FLOSS en su actividad comercial, y la utilización de FLOSS para sus procesos de desarrollo.³²

Al momento de redacción de este capítulo, el relevamiento cuenta con 105 respuestas válidas. Esto involucra empresas del sector de software de la Ciudad Autónoma de Buenos Aires (46.7% de la muestra), las Provincias de Buenos Aires (17.1%), de Córdoba (18.1%) y de Santa Fe (18.1 %).³³

En este nuevo relevamiento, al considerar no solamente el desarrollo de programas, sino también la prestación de servicios, los resultados muestran una importante participación del FLOSS en el perfil productivo del sector de software argentino. La evidencia en materia de oferta señala que más del 42% de las empresas del sector de software de la Argentina realizan desarrollo de productos FLOSS o prestan servicios a partir de un FLOSS, tanto si éste ha sido producido internamente como si se trata de un software desarrollado por terceros (p.e.: desarrollado por una comunidad). Esto configura un panorama donde 2 de cada 5 empresas de software del país realizan productos o servicios FLOSS. De éstas, en promedio, alrededor del 55% de sus ventas provienen de estas actividades, en rangos que van desde empresas que cuentan con una participación marginal de sus ingresos desde el OS (sólo 7 casos con ventas OS hasta el 20% de su facturación) hasta firmas que se especializan totalmente en FLOSS o en porcentajes muy

32- Asimismo el diseño posee un módulo de Competencias y Organización del Trabajo reducido, algunas incorporaciones para la medición de la innovación que pueden ser propias del OS, y preguntas de vinculaciones típicas que siguen al Manual de Oslo. Sobre éstas últimas, se ampliaron a actores propios de la actividad del FLOSS, así como se incorporaron preguntas sobre las formas de relacionarse con la comunidad FLOSS.

33- El período de relevamiento culmina en diciembre de 2019, con una muestra objetivo de 180 empresas, dependiendo de la tasa de respuesta. Si ésta es baja respecto de la meta, se descartarán aleatoriamente casos para asegurar la representatividad geográfica nacional.

elevados (18 casos obtienen más del 80% de sus ventas del OS).³⁴

Tabla 1.2. Desarrollo de productos FLOSS y/o Prestación de servicios a partir de un FLOSS

	Cantidad de empresas	Porcentaje
Desarrollan/Prestan	43	42,2 %
No Desarrollan	62	59,0 %
Total	105	

Los tipos de actividades de desarrollo OS y de servicios a partir del FLOSS que realizan tienen frecuencias variadas, como ilustra la Tabla 3 a continuación. Con todo, la importancia de las actividades es relevante: el 18% de las empresas argentinas de software desarrollan programas completos de código abierto, un 32% de ellas realiza desarrollo de módulos o partes de programas OS, el 21% de las empresas del sector producen otro tipo de insumos OS (tales como librerías, bases de datos, etc.) y algo más del 26% herramientas de desarrollo o de infraestructura para uso interno.

Respecto de los servicios, más del 29% de las empresas de software argentina prestan servicios de consultoría tecnológica en base a algún FLOSS, y casi el 25% de ellas presta servicios de consultoría de sistemas (en infraestructura, servidores, redes, etc.) alrededor de un software de código abierto.

34- Recordar de que la muestra está aún en un parcial grado de avance, por tanto estos porcentajes se tratan de un indicador de importancia. Además, se trata de una muestra y no tienen carácter censal. Por lo que representan un indicador, más que una participación de la población del sector.

Tabla 1.3. Tipo de actividad de desarrollo de software y de prestación de servicios que realiza

	Cantidad de Empresas	Porcentaje sobre el Total de Empresas de Software (n=105)	Porcentaje sobre las empresas que proveen servicios o productos FLOSS (n=43)
Desarrollo de programas completos OS	19	18,1 %	44,2 %
Desarrollo de módulos o partes de programa OS	34	32,4 %	79,1 %
Desarrollo de Otros insumos OS (librerías, bases de datos)	22	21,0 %	51,2 %
Desarrollo de Herramientas de desarrollo/infraestructura interno OS	28	26,7 %	65,1 %
Desarrollo de FLOSS embebido en hardware	7	6,7 %	16,3 %
Presta servicios de consultoría tecnológica en base a un FLOSS	31	29,5 %	72,1 %
Presta servicios de consultoría de sistemas en base a un FLOSS	26	24,8 %	60,5 %

Por último, los resultados con respecto al uso de FLOSS en las empresas de software del sector son abrumadores. Como puede apreciarse en la Tabla 4, casi 4 de cada 5 empresas argentinas de software utilizan FLOSS en sus procesos de desarrollo, **independientemente de si venden soluciones propietarias o de código abierto**. Dos tercios de la muestra usan herramientas o productos OS, un 41% de las empresas utilizan código OS en el desarrollo de nuevo código y la mitad de las empresas de software usan algún otro tipo de insumo OS.

Tabla 1.4. Uso de herramientas o insumos open-source en los procesos de desarrollo

	Cantidad de Empresas	Porcentaje sobre el Total de Empresas de Software
Utiliza herramientas/productos OS para el desarrollo de software	70	66,7 %
Utiliza código OS en el desarrollo de código nuevo	43	41,0 %
Utiliza otro tipo de insumo OS (librería, server, bases de datos, etc)	53	50,5 %
Utiliza algún tipo de FLOSS	80	76,2 %
No utiliza FLOSS	25	23,8 %
Total	105	

Estas estadísticas preliminares de nuestra encuesta revelan un uso muy significativo del FLOSS en las empresas de software de la Argentina, así como una participación importante en su oferta productiva. El hecho de que en algo más de cuatro de cada diez empresas del sector al menos parte de sus ventas provengan de ofrecer desarrollos de software de código abierto o de servicios alrededor de un FLOSS, desarrollado por una comunidad o internamente, habla de un fenómeno

que dista de ser excepcional o “anormal” en el comportamiento competitivo de las empresas del mercado local.

3. Reflexiones Finales

La presencia corporativa dentro de la actividad del FLOSS se ha tornado un hecho notorio. Esto es respaldado tanto por estadísticas globales de la industria del software, como por estadísticas locales del sector en la Argentina y estudios teórico-conceptuales sobre el rol económico-productivo del OS. Aunque esta presencia empresarial en el FLOSS se ha manifestado crecientemente, la economía se ha preocupado en estudiar mucho más cómo se involucran los individuos que las empresas en los proyectos OS (Lerner y Schankerman, 2013). Los prejuicios subyacentes al *mainstream* en economía, que asumen la ubicua presencia de la guía de la optimización de la utilidad en el comportamiento social o la dogmática necesidad de los derechos de propiedad privada para el funcionamiento de la actividad económica e innovativa, tienden a abordar al comportamiento de las empresas en el FLOSS como “anormal”, y han diseminado una visión del fenómeno como marginal o excepcional en términos empíricos.

El presente capítulo aporta diversos elementos que contribuyen a la visibilidad cuantitativa del FLOSS, tanto en la industria global del software, como nacional. Asimismo, se han presentado y sistematizado distintos aportes de un cuerpo de literatura que trasciende disciplinarmente a la economía, que permitan conceptualizar al FLOSS en su dimensión productivo-económica no como un hecho excepcional o anómalo, sino como un fenómeno corriente en la competencia capitalista actual en sectores tecnológicos.

Ello abre el camino, o bien constituye un punto de partida, para un programa de investigación más amplio que comience a abordar más directamente formas de cooperación y competencia en el terreno de la producción abierta y colaborativa, y su impacto en el ritmo y calidad de la innovación. Sus prácticas se expanden a otra serie de actividades productivas, que van desde la fabricación de semillas a la de medicamentos, pasando por la investigación genética, la elaboración de enciclopedias, libros y la producción electrónica en plataformas de hardware de código abierto, hasta la decodificación del genoma humano (Dughera *et al.*, 2012, van Zwanenberg *et al.*, 2017). Dado que comprender el rol de las características y ritmo de la innovación en las economías capitalistas resulta un

aspecto central para el diseño de estrategias de desarrollo en economías periféricas como la Argentina, el estudio de las tensiones que el *open source* plantea entre las formas de propiedad privada y colectiva en la generación de innovaciones, son aspectos relevantes para ser estudiados y abordados en este contexto.

Capítulo 2. La producción en comunidades FLOSS: empresas, fundaciones y *governance*

Juan Gabriel Vélez y Agustín Zanotti

El capítulo constituye un acercamiento al funcionamiento de las comunidades FLOSS, abordando sus aspectos productivo y organizativo. Los estudios existentes han puesto el foco en diferentes aspectos de las comunidades: las interacciones entre los miembros, las diferentes participaciones y formas de membresía, el papel de las empresas, las motivaciones para la participación voluntaria, entre otros. El objetivo en este capítulo es indagar, a partir de la revisión de literatura y de fuentes secundarias, los modos de *governance* al interior de comunidades a partir de la articulación de proyectos FLOSS y empresas.

La importancia del sector informático en el mundo y en particular en Argentina, lleva a la necesidad de comprender el funcionamiento de las comunidades FLOSS. En la actualidad, este cumple un rol decisivo en los desarrollos tecnológicos y en muchas innovaciones relevantes del sector. Los estudios sobre FLOSS requieren de herramientas analíticas que permiten abordar este fenómeno emergente y sus potencialidades.

Podemos definir a las comunidades FLOSS como instancias de producción entre pares, definida como “una forma de creación abierta y compartida realizada por grupos online que: establecen y ejecutan objetivos de forma descentralizada; manejan un rango diverso de motivaciones, en particular motivaciones no-monetarias; y se separan de formas de propiedad exclusivas y relaciones contractuales” (Benkler *et al.*, 2015). Las comunidades desempeñan funciones organizacionales “clásicas” como: coordinación, división del trabajo, reclutamiento, entrenamiento, creación y cumplimiento de normas, resolución de conflictos y mantenimiento de fronteras, pero lo hacen de un modo diferente.

Indagar las relaciones de poder hacia el interior de las comunidades FLOSS abre una serie de preguntas: cuál es el rol que juegan las fundaciones; cómo

funcionan los mecanismos de mérito y status en la toma de decisiones de los proyectos comunitarios; qué estrategias se dan las empresas para incidir en las comunidades. Para ello nos enfocamos en estudiar comunidades FLOSS en particular: los casos de GNOME, Fedora, Python y LibreOffice.

Este capítulo se compone de una revisión de literatura y una exploración de casos referidos a comunidades FLOSS. En las primeras dos secciones se analizan los modos de organización habituales al interior de este tipo de comunidades y las relaciones que se tejen entre empresas y proyectos FLOSS.

En la exploración de casos, trabajamos con cuatro comunidades de desarrollo: GNOME (entorno de escritorio), Fedora (distribución), Python (lenguaje de programación) y LibreOffice (suite de oficina). Cada una alberga a su vez numerosos sub-proyectos. La manera de pensar estas estructuras es la de una *ecología de sub-comunidades* (Weiss *et al.*, 2006), antes de grandes comunidades homogéneas. Esta idea permite visualizar las interrelaciones que se producen al interior de estas comunidades, donde se comparte una gobernanza común, y existe una activa interdependencia, ya que hay artefactos desarrollados de uso común.

Caracterizamos de manera exploratoria los modos de gobernanza de estas comunidades, y el rol que ocupan las fundaciones en apoyo a los proyectos. Los datos utilizados provienen de fuentes secundarias, de análisis previos sobre los estudios de casos y sitios oficiales de las comunidades consultadas.

1. Comunidades FLOSS

Según Benkler (2017), el FLOSS puede incluirse en un grupo de bienes comunes informacionales. Se trata de una familia de arreglos institucionales que ocupan un rol central en las economías actuales, mayor a la atención que han recibido de parte de la ciencia económica. Ellos se basan en derechos de uso simétricos, por oposición a derechos exclusivos asimétricos.

Los bienes comunes informacionales conllevan implicancias para las teorías de la innovación. En general se presentan en escenarios donde existen:

- (a) **externalidades altas persistentes:** innovación incremental, efecto “sobre hombros de gigantes”, derrame de conocimiento (*spillovers*), beneficios que superan el mayor uso.
- (b) **incertidumbre respecto al mercado o los usos:** entornos cambiantes, diversos, imprecisos o necesidad de compartir riesgos;

(c) **finalidades sociales:** el modelo *mertoniano* se impone sobre el managerial, diversidad de motivaciones: investigación, flexibilidad, estatus, placer de investigar (Benkler, 2017).¹

En ese sentido, algunos autores sostienen que una dimensión clave de la producción de FLOSS es la colaboración que se establece en el marco de una economía de regalo o intercambio de dones (en inglés, *gift economy*). La idea, con origen en la antropología, se basa en la reciprocidad mutua que genera entre las partes una lógica de dones y contradones. Con el crecimiento de la importancia comercial de los proyectos FLOSS, este modelo economía transitó hacia una hibridación con la economía de mercado (Berdou, 2010).

Desde un abordaje más cercano a la teoría económica, Hippel y Krogh (2003) refieren un modelo de innovación privado-colectivo, como una combinación del modelo de la inversión privada y la innovación de acción colectiva. El modelo se basa en el supuesto de que los innovadores que crean bienes de acceso público se benefician más que los *free-riders* que solamente consumen dichos bienes. Si bien el resultado de la inversión es igualmente accesible a todos, los innovadores se benefician a través del proceso de creación de estos bienes, en donde adquieren conocimientos, habilidades, reconocimiento y participan en redes que pueden luego capitalizarse.

La comunidad FLOSS está conformada por un ecosistema de actores que llevan a cabo proyectos de software de manera descentralizada. Cada comunidad puede ser conformada o bien por un puñado de programadores, o bien cientos o miles de ellos a nivel global, conectados a través de redes y plataformas de producción colaborativa. En este trabajo nos enfocamos exclusivamente en las comunidades de desarrollo de software, dejando de lado otros espacios de usuarios o activistas de cultura libre, en ocasiones vinculados a la producción informática.

Los desarrolladores FLOSS liberan versiones estables luego de cada ciclo de desarrollo, y en general mantienen accesibles versiones en desarrollo y el código fuente generado. El código utiliza licencias no restrictivas, pudiendo a grandes rasgos distinguirse entre licencias recíprocas (también conocidas como *virales*), que obligan a mantener las condiciones de código abierto, y licencias permisivas,

1- Nos referimos aquí al *ethos comunista* de la ciencia enunciado por Merton, que define al conocimiento científico como una propiedad colectiva producto de la colaboración social, donde se aprecian la honestidad, el reconocimiento y la originalidad (Merton, 1977).

que habilitan desarrollos cerrados.²

Los desarrollos surgen de una persona o grupo que busca implementar una solución particular, y va integrando el aporte de un creciente número de desarrolladores y personas encargadas de otras tareas conexas. De esta forma pueden generarse estructuras organizativas complejas, institucionalizadas en grandes fundaciones incluso. Pero también es posible que un proyecto FLOSS sea iniciado por una firma, que puede retener el control y la administración del proceso, o bien ampliar la participación a otros actores.

Es común encontrar en la literatura de FLOSS la descripción de la estructura organizativa de los proyectos dividida en dos grupos principales, uno denominado *core* y el otro periferia. Denominado en la literatura como el *onion model* o modelo cebolla. También se refiere como regla 90-9-1 de desigualdad participativa, en la que los usuarios activos (9%) y muy activos (1%) constituyen la menor parte, mientras la mayoría sólo observa o consumen información (Nielsen, 2006).

En el *núcleo del proyecto* se encuentran los líderes o mantenedores, a menudo iniciadores, y los principales contribuidores. Las actividades que realizan son aceptar y rechazar las modificaciones al código fuente que son realizadas por la periferia. Además, se encargan del mantenimiento de la calidad del software y el establecimiento de mecanismos de selección de mejoras (Weiss *et al.*, 2006).

Por su parte, la periferia está formada por un amplio número de colaboradores activos – cuya función es el testeado del software, la detección e información de errores, y la generación de mejoras o parches del código fuente– y pasivos, que sólo hacen uso de los sistemas. También en la periferia se ubican una serie de actividades como la elaboración de documentación o traducciones, que son necesarias para facilitar el acceso y reclutamiento de nuevos miembros. En general este trabajo se considera de inferior jerarquía y los no-programadores difícilmente asumen roles centrales en la comunidad (Berdou, 2010).

La división de los participantes en esta organización no es estructurada, sino que se trata de sociedades meritocráticas, donde los miembros comparten una cultura y unas normas que hacen al consenso del uso y explotación que se puede hacer del software producido, de las licencias implicadas, sus actualizaciones, etcétera. Los proyectos funcionan sin un proceso formal de recruiting. La relación entre el proyecto y los desarrolladores es voluntaria y no mediada por contratos

2- Un repaso acerca de los tipos de licencias y sus modificaciones recientes se encuentra disponible en Wheeler (2017), así como en el Capítulo 1 de este libro.

formales. Más allá de eso, quienes se vinculan con estos espacios pueden poner en valor indirectamente sus contribuciones, ya sea como una forma de capital social, tecnológico o simbólico (Zanotti, 2014).

2. Empresas y la gobernanza FLOSS

Algunas ventajas del modelo de producción FLOSS fueron enunciadas por el modelo bazar y la ley de Linus (Raymond, 1999). Con el tiempo, el interés de las empresas se proyectó en varias comunidades, desarrollando formas de vinculación diferentes.³ Capra *et al.* (2011) distinguen entre proyectos comunitarios o comerciales respecto a quien es el iniciador. Si bien en ambos la participación de empleados de empresas es significativa, en los proyectos comerciales son las empresas las que disponen del acceso al código base y definen la estrategia de evolución del proyecto. Proponen tres modelos para comprender este involucramiento que agrupan las diversas formas en que las empresas participan de las comunidades.

· *Modelo de aporte de código*: principalmente a través de la escritura de código y actividades relacionadas; resolución de *bugs*, la personalización de software o provisión de distribuciones empaquetadas particulares. También comprende situaciones donde las empresas donan una base de código pre-existente, abriendo su código; o bien lanzan un proyecto nuevo desde cero fundando la comunidad, y luego aportando las tareas de codificación descritas. Este es un modo indirecto de controlar los desarrollos de la comunidad, ya que haber hecho aportes de códigos otorga cierto poder sobre la coordinación del proyecto.

· *Modelo de management*: tareas de coordinación y *management*, especialmente durante el período de aparición de una comunidad. Sugerencia de requerimientos y de funcionalidades; planificación y diseño de la aplicación o simplemente la coordinación del desarrollo. Pueden tener el objetivo de direccionar el desarrollo del producto para ganar ventajas competitivas, o ser capaz de establecer acuerdos de cooperación con competidores en áreas de negocio estratégicas y diferenciadas. Asumen tareas de administración y de coordinación.

3- La Ley de Linus establece que “Dada una base suficiente de desarrolladores asistentes y beta-testers, casi cualquier problema puede ser caracterizado rápidamente, y su solución ser obvia al menos para alguien’. O, dicho de manera menos formal, ‘con muchas miradas, todos los errores saltarán a la vista’” (Raymond, 1999).

· *Modelo de soporte y apoyo a los proyectos OS de la comunidad*: provisión directa o indirecta de apoyo financiero, apoyo logístico con recursos de hardware, alojamiento web, apoyo de marketing a través de publicidad en sus webs o motores de búsqueda. Puede incluir participar de los foros *online* para mantener activa la comunidad, *testing* del producto a través del uso en un ambiente de producción real, oferta de cursos de entrenamiento para usuarios, reportes de *bugs* a diario o escritura de documentación, manuales de usuario o guías de instalación.

Las empresas participan en proyectos comunitarios pensando con un sentido estratégico de cómo intervenir en la dirección de determinada tecnología. Tales modelos no se plantean como excluyentes. Las empresas pueden combinar estos tres tipos de modelo de involucramiento en proyectos de la comunidad, participando en múltiples niveles.

El modelo de aporte de código, es el que debiera recoger la relación que se teje con la comunidad por medio del reclutamiento de programadores que pasan a ser empleados de empresas. Sin embargo, en el análisis concreto de comunidades como GNOME, tal como veremos más adelante, se establecen complejidades que emergen en este proceso de interpenetración entre las relaciones sociales y comerciales-económicas (Berdou, 2010).

Esto ha afectado los mecanismos de gobernanza de las comunidades. El concepto de ‘gobernanza’ difiere de ‘gobierno’ en ser un sistema de conducción que involucra una gama de actores y redes más amplias que las existentes en una organización estructurada jerárquicamente. En el caso del FLOSS, las principales características de la gobernanza están dadas por: la apertura, la creación de redes, la participación y la transparencia en los proyectos. Estos proyectos no operan en jerarquías estrictas de comando y control, sino “en un entorno mucho más flexible que (...) permite la existencia de múltiples equipos de participantes que trabajan simultáneamente en una variedad de direcciones, inclusive opuestas” (P2PF, 2019). Esto ha dado lugar al concepto de gobernanza de pares:

En una empresa tradicional, es la administración, encabezada por el CEO y la junta directiva, la que controla la empresa y proporciona el ímpetu para el desarrollo continuo. Si bien la visión del liderazgo es igualmente integral para el eventual éxito de cualquier proyecto de código abierto, su control continuo es más frágil y depende de su relación y respuestas con la comunidad. (Nyman y Lindman, 2013)

Nyman y Lindman (2013) identifican tres niveles de gobernanza, referidos a 1. el desarrollo de software propiamente dicho, 2. la gestión de la comunidad y 3. el ecosistema en que se inserta. Hacen hincapié en la posibilidad de realizar bifurcaciones (*forks*), los cuales implican tomar el código e iniciar una escisión respecto del proyecto original. En la medida en que los *forks* implican según el caso oportunidades o dificultades, son un principio regulador de la gobernanza. La misma marca de este modo relaciones de poder hacia el interior de los desarrollos, especialmente en la toma de decisiones y el peso relativo que pueden mantener actores empresariales, así como la identidad y la influencia que implica ciertos límites para actores externos.

Evangelia Berdou (2010) propone examinar el funcionamiento organizativo de las comunidades FLOSS para penetrar en el análisis de las relaciones que se tejen en estos modos de producción. A partir de aportes teóricos de la sociología económica como el concepto de *embeddedness*, el desarrollo de tecnologías para el gobierno y la noción de comunidad de práctica, establece una crítica a ciertos estudios *sobresocializados*.⁴ Para salir de esta dicotomía, Berdou establece el análisis de las comunidades en su relación mercantil, en las dinámicas de mercado y flujos monetarios donde están insertas (*embedded*).

A partir de ello propone la noción de *tecnologías de comunidad* como abordaje al fenómeno de las comunidades FLOSS. Estas tecnologías vendrían a estar constituidas por tres dimensiones: i) un programa meritocrático que estructura el modo de organización; ii) la invocación a la idea de comunidad para movilizar recursos; iii) las herramientas y técnicas para la gestión de la comunidad. Con este aparato teórico se propone la comprensión del sentido social en el desarrollo de FLOSS, identificando reglas, procesos y prácticas que dan sustento a definiciones específicas de mérito y status, que son claves en el acceso y control de la comunidad (Berdou, 2010).

Berdou evidencia la estrategia de las empresas de contratar desarrolladores reconocidos en la comunidad. En estas comunidades, los *maintainers* empleados por firmas tienen acceso a aspectos claves del desarrollo de los proyectos. Al contar con recursos externos, estas personas tienen mayores chances de invertir tiempo

4- Esto se refiere a la crítica de Mark Granovetter (1992) de aquellos estudios que se proponen como alternativos a los mecanicismos propios de la economía convencional, pero que caen en otro tipo de determinismo. En este caso, el error sería pasar a una visión donde los procesos sociales, las normas y valores de la comunidad FLOSS, son determinantes de la conducta de las personas, perdiendo de vista las estructuras políticas y económicas existentes.

en consolidar su posición en la comunidad, asistir a los eventos sociales, mejorar sus habilidades técnicas y asumir roles de mayor responsabilidad (Berdou, 2010).

En parte, esta presencia de las firmas por medio de la contratación genera beneficios en las comunidades porque permite retener a personas capacitadas y comprometidas. También les permite a las empresas contar con mayor capacidad de incidir en los procesos de desarrollo. Sin embargo, se establecen ciertos límites a esta capacidad de influenciar de las firmas.

Mientras más establecida esté una comunidad, es mayor la cantidad de firmas que colaboran, y menos probable es que una en particular tenga capacidad de capturar la agenda de desarrollo. También es menos probable que los desarrolladores empleados sean capaces de llevar a cabo las agendas comerciales de sus empleadores. Incluso, pueden desarrollar estrategias para evitar tomar trabajos que puedan ir en contra de los intereses de la comunidad. Algunas de estas tensiones son recogidas y reguladas por medio de la acción de las fundaciones.

3. Fundaciones FLOSS

La tendencia constatable en los principales proyectos FLOSS es unirse a una fundación existente, o bien la creación de una propia. Hay dos tipos de organizaciones sin fines de lucro en la comunidad FLOSS. La primera es la fundación de proyectos grandes. Los casos más conocidos y estudiados son la *Linux Foundation*, *Apache Software Foundation* o la *Eclipse Foundation* (Riehle y Berschneider, 2012). Otros ejemplos se refieren a nichos o proyectos intermedios, como la *GNOME Foundation* o la *Document Foundation*. Estas apoyan y trabajan en estrecha colaboración con un gran proyecto institucionalmente independiente.

El segundo tipo se refiere a la promoción o cuestiones de interés para muchos o todos los proyectos de software libre. La *Open Source Initiative*, *Software Freedom International* y la *Free Software Foundation* son ejemplos de este tipo. Esta última también apoya el Proyecto GNU, por lo que se puede clasificar en ambos campos (P2PF, 2019).

Una fundación tiene muchos beneficios: actúa como representante legal ante conflictos, protege el aporte individual de los desarrolladores; le da credibilidad al proyecto, y lo hace más independiente de personas en particular, lo que aumenta el interés empresarial y le otorga sostenibilidad, permite recolectar fondos (Riehle y Berschneider, 2012). Las fundaciones además patrocinan el desarrollo de proyectos, coordinan esfuerzos, proveen apoyo en asuntos legales y financieros, y gestionan las relaciones con las empresas (Berdou, 2010).

Existen una gran cantidad de fundaciones, cientos de ellas. La alta heterogeneidad respecto a las configuraciones institucionales y sus modos de funcionamiento busca ser captado en la propuesta de Riehle y Berschneider. Los autores proponen seis bloques en su modelo de fundaciones:

Tabla 2.1. Modelo de fundaciones de desarrollo FLOSS

Categorías	Atributos
General	Propósito
	Constitución
	Miembros
Filosofía	Postura de Open Source
	Modelo de desarrollo
Propiedad Intelectual	Licencia del proyecto
	Concesión de licencia de patente
	Propiedad de derechos de autor
Gobernanza	Membresía de la junta
	Membresía del proyecto
	Carrera de una persona dentro de la fundación
	Participación de personas jurídicas
Financiamiento	Financiamiento
	Proyectos
Operaciones	Infraestructura
	Back office

Elaboración Propia en base a Riehle y Berschneider (2012)

En el aspecto más formal (General) se establece la pertenencia a la fundación, si va a ser para personas naturales o jurídicas; si el beneficio de la fundación está planteado en términos del público general o sólo sus miembros. Luego hay dos bloques de posturas en torno al desarrollo de software y derechos intelectuales (Filosofía y Propiedad Intelectual).

Respecto de la Gobernanza, establecen las posibles diferencias en torno al modo de elección de la junta directiva. Puede ser democrática, autocrática o

meritocrática, de igual manera que la decisión de pertenencia de un proyecto a la fundación. Por otra parte, se establece el esquema de evolución que puede tener dentro de la fundación, una persona natural o jurídica, según sea el caso. Esta trayectoria describe idealmente el modo en que se puede asumir roles cada vez más importantes en la estructura de decisión.

En relación al financiamiento de las fundaciones, hay cuatro mecanismos habituales: i) pago por membresía; ii) *sponsorship* (patrocinio); iii) donaciones y subvenciones; y iv) ingresos provenientes de una empresa subsidiaria, que es el menos habitual. La última dimensión es Operaciones, la cual implica el modo de gestión cotidiana de la fundación y el uso de infraestructura propia o la que aportan los miembros.

4. *Análisis de casos*

A continuación presentamos cuatro casos de proyectos de desarrollo FLOSS, a partir de los cuales podemos analizar trayectoria de gobernanza. Consideramos que la misma depende del tipo de desarrollo que se trate, motivo por cual incluimos:

- **un entorno de escritorio:** es una implementación de la metáfora del escritorio conformada por un conjunto de programas que se ejecutan en la parte superior de un sistema operativo, que comparten una interfaz gráfica de usuario (GUI) común. Permite los usos más habituales y la configuración del hardware del dispositivo. Generalmente cuenta con utilidades y aplicaciones para la gestión, visualización y edición de diferentes formatos, además de conectividad y navegación en red.

- **una distribución GNU/Linux:** es un sistema operativo hecho de una colección de librerías de software, basadas en el kernel de Linux y, a menudo, en un sistema de administración de paquetes. Las distribuciones se basan en el proceso permanente de empaquetado de una diversidad de proyectos, proveyendo a sus usuarios de nuevas versiones y actualizaciones de seguridad.

- **un lenguaje de programación:** es un lenguaje formal que proporciona una serie de instrucciones que permiten a un programador escribir secuencias de órdenes y algoritmos que a modo de controlar el comportamiento de un dispositivo electrónico, con el objetivo de que produzca diversas clases de datos y resultados.

- **una suite ofimática:** es un conjunto de aplicaciones utilizadas en tareas de oficina, para crear, modificar, organizar e imprimir documentos, planillas de cálculo, presentaciones y bases de datos, entre otras.

4.1. GNOME

GNU Network Object Model Environment (Entorno de Modelo de Objeto de Red), conocido como GNOME, es uno de los entornos de escritorio más populares para sistemas operativos Unix. El proyecto fue iniciado por los programadores mexicanos Miguel de Icaza y Federico Mena en octubre de 1997. GNOME está disponible en las principales distribuciones GNU/Linux, siendo en algunas el escritorio por defecto. El proyecto es dirigido por la *GNOME Foundation* y pone un fuerte énfasis en la filosofía de software libre, formando parte oficial del proyecto GNU.

Los desarrollos de GNOME se organizan a través del Proyecto GNOME. Este contempla dos ciclos de desarrollo anuales, que se mantienen regulares desde 2002, publicando ediciones estables en marzo y septiembre (GNOME, 2019b). El escritorio alcanza en 2019 en su versión 3.34 y se encuentra en preparación la versión 4.0. Su repositorio Github cuenta al momento con 429 subproyectos vinculados, que forman parte del conjunto de sus componentes y aplicaciones *core* y *extras* (Gnome Github, 2019).

El GIMP ToolKit (GTK+) es la base de GNOME. Desde sus inicios, este kit de herramientas usa una Licencia Pública General Reducida (LGPL) de GNU, que permite que el software que se vincule con ella utilice un conjunto mucho más amplio de licencias, incluidas licencias de software propietario. Además, GNOME está autorizado bajo la LGPL para sus bibliotecas y Licencia Pública General de GNU (GPL) para sus aplicaciones (GNOME, 2019d).

En el año 2000 se creó la GNOME Foundation. De acuerdo a su Carta de fundación, entre sus objetivos encontramos:

- La Fundación coordinará las versiones de GNOME y determinará qué proyectos forman parte de GNOME.
- actuará como una voz oficial para el proyecto GNOME, proporcionando un medio de comunicación con la prensa y con organizaciones comerciales y no comerciales interesadas en el software GNOME.
- puede producir materiales educativos y documentación para ayudar al público a aprender sobre el software GNOME.
- puede patrocinar conferencias técnicas relacionadas con GNOME y representar a GNOME en conferencias relevantes patrocinadas por otros,
- ayudar a crear estándares técnicos para el proyecto y promover el uso y desarrollo del software GNOME. (GNOME, 2019a)

La Fundación está compuesta por contribuidores que pertenecen como

miembros y tienen el derecho de postularse a cargos ejecutivos en la junta directiva, así como participar en la elección anual. La membresía es personal y la junta directiva es el organismo que gobierna la fundación. Cuenta además con una junta asesora formada exclusivamente por empresas y organizaciones. Las pequeñas firmas hacen un pago anual de U\$S11.500 y las medianas y grandes de U\$S 23.000. Actualmente integran la junta asesora Canonical, Debian, Endless, FSF, Google, Private Internet Access, Red Hat, Sugar Labs, SUSE, The Document Foundation, System76.

Esta junta asesora se encarga de representar los intereses de estas organizaciones, reuniéndose regularmente con la junta directiva para discutir la dirección de los proyectos y prestar apoyo. Esto último es en relación a asuntos de marketing, de respuestas de los clientes, finanzas y prestar otros tipos de recursos. Se interpreta la participación de las empresas como un mecanismo que canaliza la demanda de los usuarios, los requisitos específicos de los usuarios finales y que tienen mayor contacto por intermedio del mercado (Berdou, 2011).

En lo que respecta al financiamiento, los pagos provenientes de la junta asesora representan una porción menor de los ingresos. La mayor parte proviene de las donaciones. El proyecto llegó a recibir donaciones anónimas de 1 millón de dólares en 2018 (GNOME News, 2018 25 de mayo).

La GNOME Foundation actuó en defensa de la propiedad de la marca GNOME, la cual le fue disputada en una oportunidad. También debió afrontar demandas por violaciones de patentes, siendo el último caso en 2019 (GNOME, 2019f). Además, participa de becas de programación como *Google Summer of Code*, programa Desde 2006, lleva un programa de pasantías con rondas organizadas dos veces al año. Su objetivo es involucrar a personas de grupos subrepresentados, tanto por su lugar de pertenencia, su etnicidad o su diversidad de género (GNOME, 2019e). También patrocina y organiza eventos, como GUADEC, que es la conferencia anual de los usuarios y desarrolladores de GNOME.

En su operatoria, la fundación cuenta con un Director ejecutivo; Director de operaciones; Coordinador de programas, Gerente de iniciativas estratégicas; Líder de equipo de desarrollo GTK; Ingeniero jefe de sistemas e Ingeniero DevOps (GNOME, 2019c).

4.2. Fedora

Fedora es una distribución basada en GNU-Linux extendida entre los usuarios de software libre. El Proyecto Fedora surgió en 2003 bajo el patrocinio de Red Hat Inc., firma líder dedicada al desarrollo, implementación y soporte de tecnologías libres. Opera de este modo una articulación con el espacio comunitario, en el que la empresa provee de código, infraestructura y recursos, mientras se beneficia de la innovación, experimentación y el desarrollo motorizado por sus usuarios. Recientemente, Red Hat fue adquirida por IBM en julio de 2019.

La última versión estable es Fedora 30, publicada en abril de 2019. En cuanto a su gobernanza, el proyecto atravesó diferentes modalidades organizativas a lo largo de su historia, las cuales evidencian el lugar de la empresa y la comunidad en las decisiones y dirección del desarrollo.

El proyecto comenzó organizándose con la idea de una fundación, la *Fedora Foundation*. La misma se definió con los objetivos de:

- proporcionar una entidad sin fines de lucro para organizar y administrar voluntarios.
- garantizar que el trabajo de estos voluntarios permanezca siempre libre.
- proporcionar un brazo de recaudación de fondos para el desarrollo y protección de Fedora y proyectos de código abierto relacionados.
- proporcionar una entidad para la asignación de derechos de autor, de modo que lo que es libre también fuera defendible en un tribunal de justicia.
- financiar solicitudes de patentes para inventores en la comunidad de código abierto, de modo que individuos dedicados puedan ayudar a construir un escudo protector de patentes alrededor del código de código abierto. (Fedora Project Sitio web, 2019)

La entidad legal fue creada en 2005 con estatutos muy básicos y flexibles, y la designación de una junta para que lo ejecute temporalmente. Pero al poco tiempo quedó en evidencia que no se adecuaba a los requerimientos del proyecto. En 2006, la fundación fue reemplazada por la *Fedora Project Board*, que incluía tanto Red Hat como liderazgo comunitario voluntario. La Junta de Fedora comprendía cinco miembros nombrados por Red Hat y cinco miembros elegidos por la comunidad. Además, el líder del Proyecto Fedora tenía poder de veto sobre cualquier decisión de la junta.

Eso, a su vez, fue reemplazado por el *Fedora Council* en 2014. El Consejo de Fedora es actualmente el organismo de liderazgo y gobierno comunitario de alto

nivel. El Consejo está compuesto por una mezcla de representantes de diferentes áreas del proyecto, roles designados por Red Hat y un número variable de escaños relacionados con los objetivos del proyecto a mediano plazo. En el modelo actual, todos los miembros con derecho a voto pueden bloquear asuntos, con una razón válida, y se insiste en un proceso de establecimiento de consensos (Fedora Docs, 2019).

La comunidad también está involucrada en la organización de niveles más bajos de liderazgo, tanto el *Engineering Steering Committee* (Comité Directivo de Ingeniería) como el *Mindshare Committee*, responsables de la supervisión técnica y comunitaria, respectivamente. Ambos son organismos elegidos por la comunidad que administran porciones significativas del proyecto.

El proyecto facilita la comunicación en línea entre sus desarrolladores y miembros de la comunidad a través de listas de correo públicas y páginas wiki. También coordina dos eventos principales, conocidos como la *Conferencia de Usuarios y Desarrolladores de Fedora* (FUDCon) y *Flock* (o Flock to Fedora). FUDCon es un evento que se realiza en diferentes ubicaciones en las dos regiones designadas, América Latina y Asia/Pacífico. Por lo general, gira en torno a alguna combinación de conferencias, eventos sociales y un hackathon (Fedora Project Wiki, 2019b). Flock es un evento similar que reemplazó a FUDCon en Norteamérica y Europa/Medio Oriente (Flock To Fedora, 2019). Los eventos programados de manera flexible y centrados en tareas, conocidos como *Fedora Activity Days*, también reúnen a muchos contribuyentes en varias regiones. A su vez, grupos locales se organizan para promover la comunidad. Son precedidos por un Community Manager y embajadores, “contribuyentes experimentados del proyecto Fedora cuya responsabilidad es la de representar, promover y expandir el Proyecto Fedora y sus ideales a la gran comunidad de software libre y de código abierto, y al público en general” (Fedora Project Wiki, 2019a). El proyecto cuenta con comunidades en diferentes países de la región incluyendo: Argentina, Brasil, Chile, Colombia, El Salvador, Guatemala, México, Nicaragua, Panamá, Paraguay, Perú, Venezuela y Uruguay

4.3. Python

Python es un lenguaje de programación multiparadigma creado a finales de los ochenta. Alcanzó la versión 1.0 en enero de 1994 y en la actualidad se encuentra en su versión 3.7, con cambios significativos respecto a su versión anterior. Python ha ganado popularidad en el último tiempo, destacando en

áreas como desarrollo web, desarrollo de interfaces de usuario, análisis científico y numérico (incluyendo aprendizaje automático, big data), desarrollo de software y administración de sistemas. El desarrollo de este lenguaje se organiza de forma comunitaria. El repositorio oficial mantiene unos 191 mil proyectos, algunos mantenidos en forma oficial y otros producidos de manera independiente por la comunidad (PyPI, 2019).

El lenguaje posee licencias de código abierto protegidas a nivel global por la Python Software Foundation (PSF), encargada de gestionar las licencias de código abierto para Python versión 2.1 y posteriores, junto a las marcas comerciales asociadas con Python. La fundación mantiene la infraestructura, el sitio web el índice de paquetes, la documentación de Python y muchos otros servicios en los que se basa la comunidad (PSF, 2019a).

La principal fuente de fondos para el funcionamiento de la fundación proviene del aporte de organizaciones como *sponsor*. Para ello, debe haber aprobación por parte de un comité y se cuentan con diversos niveles de aportes (desde 500 a 150.000 dólares al año).

También organiza anualmente la conferencia PyCon de América del Norte, apoya a otras conferencias de Python en el mundo y financia el desarrollo relacionado con Python, con subvenciones y financiamiento para proyectos especiales. Ha contribuido económicamente con proyectos relacionados con el desarrollo de Python, tecnología relacionada con Python y recursos educativos en más de 50 países (PSF, 2019b).

La expansión de Python a nivel mundial fue acompañado de la proliferación de grupos de usuarios locales, como lo es Python Argentina (PyAr): “Nuestro objetivo es nuclear a los usuarios de Python. Pretendemos llegar a personas y empresas, promover el uso de Python e intercambiar información.” (PyAr Sitio web, 2019)

El grupo comenzó a organizarse en 2004 a partir de entusiastas que venían utilizando este lenguaje para propósitos diversos, quienes comenzaron a reunirse mensualmente en Capital Federal para dar forma al espacio, pautar actividades y conformar la lista de correo. PyAr realiza actividades de difusión, desarrollo y soporte. Entre ellos se destacan los *Pycamps*, *Python bug days*, *Pydays*, campañas y más. También comenzó en los últimos años a editar una publicación digital sobre este lenguaje. Sus miembros mantienen a su vez una variedad de proyectos de desarrollo que se nutren del soporte comunitario.

PyAr llegó así a ser la mayor comunidad Python de habla hispana a nivel global y el grupo especializado en un lenguaje de programación más grande del país. La comunidad argentina fue la primera en organizar una *PyCon* (Conferencia de Python) en la región, con presencia de invitados extranjeros destacados.⁵

4. 4. LibreOffice

LibreOffice es una suite de oficina que nació como un *fork* de otro proyecto llamado OpenOffice. Este a su vez tenía como código base a la suite ofimática StarOffice, cuyo desarrollo inicial data del año 1985.

Inicialmente los desarrollos eran realizados en el marco del proyecto bajo el liderazgo de Sun Microsystems. La gobernanza del proyecto generó una serie de fricciones en torno a la aceptación de aportes y la relación con empresas, como IBM, que fue mellando la confianza en el proyecto. Esta situación llegó a un límite cuando Sun es comprada por Oracle, empresa que ya había tenido comportamientos negativos con respecto a la comunidad FLOSS. Por ello, y ante la posibilidad de la discontinuidad o la imposición de mayores restricciones, un grupo de desarrolladores decide abandonar el proyecto OpenOffice para dar nacimiento a LibreOffice.

El proyecto LibreOffice es alojado en la Document Foundation, una organización sin fines de lucro con sede en Alemania creada en 2010. Su objetivo es producir una suite ofimática independiente del proveedor con soporte ODF en un entorno de desarrollo libre del control de la compañía (Document Foundation, 2019).

El proyecto es gobernado por la comunidad y pone un fuerte énfasis en la filosofía de software libre, reflejado en el respeto de las cuatro libertades básicas (referidas en el Capítulo 1). Asimismo, incluyen en su visión una serie de valores como el respeto a las lenguas nativas, al empoderamiento de los ciudadanos, acceso al software sin costo. Mantienen una postura de desarrollo en torno al software libre ligada a la creencia de que puede proveer *mejor calidad, confiabilidad, seguridad y flexibilidad que las alternativas propietarias* (Document Foundation, 2019).

Esta fundación es controlada por la propia comunidad de LibreOffice, teniendo la función de proteger los desarrollos vía el uso de licencias de código abierto o

5- Algunos casos de empresas argentinas con un modelo de negocios basado en este lenguaje son presentados en el Capítulo 4.

FLOSS, recíprocas (como GNU Lesser GPL) y no recíprocas (Mozilla Public License), así como de contenidos (Creative Commons Attribution-ShareAlike License).

En lo que refiere al aspecto organizativo, los miembros son personas que ingresan a título individual. Cuenta con una Junta Directiva, que representa la fundación y lleva la administración general de proyectos y equipos. Pueden nombrar cargos para que lleven a cabo aspectos administrativos o de funcionamiento básico de la fundación. Además tiene un Comité de Membresía, que administran solicitudes y renovaciones de membresías, así como supervisar la elección de los miembros de la Junta Directiva. Y un *Board of Trustees*, compuesto por todos los miembros que contribuyen activamente al proyecto. Hay que hacer una aplicación para poder pertenecer y se siguen principios meritocráticos, no admitiéndose exclusivamente donaciones para poder pertenecer.

Luego está la dirección de ingeniería (*Engineering Steering*), que es un cuerpo que presta apoyo a la Junta pero no de manera formal. Está compuesto por aquellos que tienen las mayores capacidades técnicas.

El Consejo Asesor, formado por un conjunto de organizaciones que contribuyen al financiamiento y prestan consejo a la fundación. Actualmente lo integran la Free Software Foundation, BMP-Conseil, KDE, la Ciudad de Munich, Kopano, los UK Government Digital Services, Red Hat, Software in the Public Interest, CIB Software, Collabora, RPA RusBITech, Adfinis SyGroup, GNOME y Google.

5. Discusión

Caracterizamos de manera exploratoria cuatro comunidades y sus modos de gobernanza, para encontrar particularidades en sus trayectorias. Podemos observar la relación entre el origen y manejo de los proyectos, así como las formas institucionales que asume su gobierno. Cabe resaltar que la gobernanza se relaciona también con el tipo de proyecto de que se trate, habilitando diferentes conexiones y modelos de negocio con empresas. De acuerdo con Scharschmidt y Von Kortzfleisch (2015), los proyectos FLOSS pueden surgir de diversas maneras. Algunos de ellos son iniciados por una persona o un grupo de personas, que van creciendo hasta volverse una comunidad de desarrolladores, mientras otros son iniciados por empresas. A su vez, respecto de la administración y control, los proyectos FLOSS pueden tener detrás a una única firma o a una multiplicidad de firmas.

En el caso de GNOME, se trata de un escritorio surgido con fuertes ideales libertarios y devenido una de las principales herramientas FLOSS. Su modelo de fundación mantiene un principio democrático en la elección de las autoridades, pero permite la capacidad de incidir de empresas y ubica en roles jerárquicos de los proyectos a desarrolladores pagos. La fundación canaliza recursos tanto de empresas líderes de mercado como emprendimientos de menor escala.

La distribución Fedora resulta de interés porque es uno de los casos en que el modelo de fundación fue dejado de lado en favor de un mayor control por parte de su compañía madre. El diseño de Junta y luego Consejo directivo dejó institucionalizado el predominio de la empresa, la cual cuenta sin embargo con un reconocimiento especial dentro del ecosistema FLOSS. La comunidad se mantiene, sin embargo, para el desarrollo, actividades de difusión y promoción.

En el lenguaje Python, la comunidad se organiza en dos niveles. El primero es el desarrollo de los módulos que constituyen el propio lenguaje, lo que aumenta sus prestaciones y eficacia, y los desarrollos liberados por la comunidad que hacen uso de este lenguaje. La fundación concentra sus esfuerzos en promover el primero de los componentes, mediante subvenciones y conferencias, obteniendo recursos del patrocinio. Mientras tanto, favorece la puesta en disponibilidad de muchas implementaciones, mediante un catálogo de desarrollos.

El caso de Libreoffice puede ser leído en oposición al de Fedora. Luego de sucesivos ciclos de desarrollo bajo la órbita de una empresa madre, el desarrollo es bifurcado (fork) hacia un nuevo proyecto liderado ahora sí por una fundación. Este último refuerza las definiciones libertarias asociadas al FLOSS y los formatos abiertos, y logra una impronta más comunitaria. De este modo, más allá del patrocinio de empresas líderes del sector, no hay firmas que tengan en la actualidad una influencia decisiva sobre la gestión de esta comunidad.

La Tabla a continuación resume las apreciaciones presentadas:

Tabla 2.2. Clasificación de los proyectos de acuerdo a origen y administración

Origen	Gestión - administración de los proyectos		
	Única empresa por detrás	Multiplicidad de empresas detrás	Relativa independencia de Firmas
Iniciado por firma/s - Proyecto Comercial	Fedora (Consejo)		LibreOffice (Fundación)
Iniciado por Comunidad FLOSS - Proyecto Comunitario		GNOME (Fundación) Python (Fundación)	

Elaboración propia en base a Schaarschmidt *et al.* (2015) y Capra *et al.* (2011).

6. Cierre

A lo largo del texto avanzamos en una caracterización de comunidades de software, asociadas a un modelo de desarrollo de bienes comunes informacionales basado en la colaboración masiva entre pares. Observamos en qué medida la proliferación del FLOSS en la industria ha impactado en los principales proyectos y sus modos de organización y dirección.

Definimos la gobernanza como una forma de gestión organizacional que se diferencia de las empresas tradicionales. El término gobernanza de pares apunta a una forma de control más participativa y abierta, que depende de su relación con la comunidad y el ecosistema en que se encuentran insertos los desarrollos de software.

Resulta relevante establecer el rol que juegan las fundaciones en estos procesos. Esta configuración ha llegado a ser habitual en los desarrollos FLOSS, regulando la multiplicidad de empresas detrás de los proyectos y generando, al mismo tiempo, una relativa independencia de firmas. El apoyo a fundaciones pueden ser parte de la estrategia de las empresas para involucrarse en la comunidad FLOSS, generando un marco de confianza para la participación de voluntarios. Establecen, a su vez, límites a la capacidad de apropiación y direccionamiento del trabajo de las comunidades por parte de empresas particulares.

Las estrategias de las empresas van más allá de las cuestiones de formalidad en las fundaciones, cómo se ve para el caso de contratación de desarrolladores de peso en los proyectos FLOSS. Por ello la importancia de superar ciertos esquemas,

para dar paso a nociones más complejas que den cuenta del modo de organizarse en las comunidades FLOSS.

Fuentes Documentales

- Document Foundation (2019). Overview. Recuperado de <https://www.documentfoundation.org/foundation/overview/> (Acceso: 9 de octubre 2019).
- Fedora Docs (2019). *Fedora Council Charter*. Recuperado de <https://docs.fedoraproject.org/en-US/council/> (Acceso: 9 de octubre 2019).
- Fedora Project (2019). *Foundation*. Recuperado de <https://fedoraproject.org/wiki/Foundation> (Acceso: 9 de octubre 2019).
- Fedora Project Wiki (2019a). *Ambassadors/es*. Recuperado de <https://fedoraproject.org/wiki/Ambassadors/es> (Acceso: 9 de octubre 2019).
- Fedora Project Wiki (2019b). *FUDCon*. <https://fedoraproject.org/wiki/FUDCon?rd=FUDcon> (Acceso: 9 de octubre 2019).
- GNOME (2019a). GNOME Foundation Charter Draft 0.61 (23 October 2000). Recuperado de <https://wiki.gnome.org/FoundationBoard/Resources/Charter> (Acceso: 9 de octubre 2019).
- GNOME (2019b). Release Notes. Recuperado de <https://help.gnome.org/misc/release-notes/> (Acceso: 9 de octubre 2019).
- GNOME (2019c). The GNOME Foundation. Recuperado de <https://www.gnome.org/foundation/> (Acceso: 9 de octubre 2019).
- GNOME (2019d). GNOME Foundation Guidelines on Copyright Assignment. Recuperado de <https://wiki.gnome.org/FoundationBoard/Resources/CopyrightAssignment/Guidelines> (Acceso: 9 de octubre 2019).
- GNOME (2019e). Outreachy/History. Recuperado de <https://wiki.gnome.org/Outreachy/History> (Acceso: 9 de octubre 2019).
- GNOME (2019f). GNOME Foundation facing lawsuit from Rothschild Patent Imaging. Recuperado de <https://www.gnome.org/news/2019/09/gnome-foundation-facing-lawsuit-from-rothschild-patent-imaging/> (Acceso: 9 de octubre 2019).
- GNOME GitLab (2019). *Details*. Disponible en: <https://gitlab.gnome.org/GNOME>
- GNOME News (2018, 25 de mayo). Anonymous Donor Pledges \$1M Donation Over Two Years. Recuperado de <https://www.gnome.org/news/2018/05/anonymous-donor-pledges-1m-donation-over-two-years/> (Acceso: 9 de octubre 2019).

- Peer 2 Peer Foundation (P2PF) (2019). Peer Governance. Recuperado de https://wiki.p2pfoundation.net/Peer_Governance (Acceso: 9 de octubre 2019).
- Python Software Foundation (PSF). (2019). Python Software Foundation. Recuperado de <https://www.python.org/psf-landing/> (Acceso: 9 de octubre 2019).
- Python Argentina (PyAr). (2019a). Python Argentina. Recuperado de <https://www.python.org.ar/> (Acceso: 9 de octubre 2019).
- PyAr (2019b). PSF Annual Report. Recuperado de <https://www.python.org/psf/annual-report/2019/> (Acceso: 9 de octubre 2019).
- PyPI (2019). The Python Package Index. Recuperado de <https://pypi.org/> (Acceso: 9 de octubre 2019).

Capítulo 3. Grandes corporaciones de software y el FLOSS: cooperar en entornos abiertos como estrategia de competencia

Ignacio Juncos y Carina Borrastero

En el contexto de crecimiento general de la industria del software a nivel global, la aparición del FLOSS ha modificado las actividades y modelos de negocio de la cadena de valor de toda la industria del software, lejos de limitarse a constituir solo un movimiento ideológico como era considerado en sus orígenes. En la actualidad es prácticamente imposible realizar negocios sin encontrar software creado con métodos de desarrollo *Open Source* (OS) (Lindman *et al.*, 2008) que se basan en el trabajo colaborativo y mayoritariamente no remunerado de millares de desarrolladores distribuidos por el mundo que aportan sus conocimientos a proyectos abiertos no fundados en la lógica de la propiedad privada.

En este marco, el ámbito OS ha ido virando progresivamente desde esquemas de sostén de las comunidades antes basados exclusivamente en trabajo voluntario, hacia esquemas de sostén privado o combinado en los que empresas de toda envergadura valoran el FLOSS y amplían su participación en este campo cada vez más notoriamente, aportando financiamiento y recursos humanos en volúmenes variables (Capiluppi *et al.*, 2012).

En este capítulo se propone un trabajo exploratorio orientado a analizar específicamente la relación de grandes actores privados de la industria global del software con la comunidad FLOSS, y a intentar comprender por qué estas empresas realizan inversiones significativas a una actividad cuyos beneficios pueden ser inmediatamente apropiados por la competencia. Partiendo del concepto de “*open cooptation*” de Teixeira *et al.* (2016), avanzamos en la comprensión de este fenómeno como lógica de organización predominante del mercado del software en la actualidad.

Hay profundas razones para focalizarse en la importancia de las empresas y, en particular, de las grandes multinacionales líderes del mercado del software

como Microsoft, Google, Oracle, etc. para el desarrollo de las comunidades y proyectos FLOSS. Por un lado, además de participar en proyectos comunitarios ya existentes, estas empresas incluso lanzan sus propios proyectos a la comunidad y los lideran, abriendo así sus desarrollos y contando con que otros programadores se sumen a trabajar en ellos voluntariamente. Por otro lado, empresas de gran porte y alta rentabilidad basan su modelo de negocio directa e íntegramente sobre software de código abierto, como el caso de la ex Red Hat o Canonical. Estos comportamientos resultan particularmente llamativos al analizarlos desde la teoría económica convencional, que no permite reconocer ni conceptualizar el hecho de que las empresas, incluso líderes, dediquen importantes recursos a proyectos en los que participan con sus competidores y no existe apropiabilidad privada de los productos. Sin embargo, como se verá aquí, el fenómeno es mucho más frecuente y extendido de lo que podría suponerse.

A su vez, en los últimos años el mercado ha asistido a un fenómeno significativo en dirección a la ampliación de la influencia del FLOSS en el mercado, como es la adquisición de compañías OS por parte de multinacionales como IBM o MercadoLibre que comienzan a entrar o se expanden en el desarrollo de software con tecnologías de avanzada como *cloud computing*, inteligencia artificial, big data y otras.

De manera que existen razones para considerar que la sostenibilidad a largo plazo de las contribuciones OS descansa hoy en la coexistencia en el mundo del FLOSS de las comunidades tradicionales junto a un sector de desarrollo de software “orientado a” u “orientado por” el mercado (Lerner y Schankerman, 2013).

En el trabajo se emplea una metodología cualitativa, basada en la revisión bibliográfica y documental de fuentes secundarias, principalmente artículos académicos y de prensa especializada. Los criterios de selección para conformar el corpus de empresas a investigar fueron dos, orientados según la identificación de dos modelos de negocio diferentes por parte de las grandes empresas de software con participación relevante en FLOSS:

- Empresas multinacionales productoras de software y líderes del mercado que generan software de tipo “privativo” y contribuyen al FLOSS como política corporativa de alta relevancia.
- Empresas de gran tamaño, rentables y con presencia global que basan su modelo de negocios íntegramente en FLOSS.

Con la finalidad de identificar las empresas ajustadas al primer criterio utilizamos rankings globales de ventas e innovación de la industria del software e indagamos en fuentes secundarias sobre sus vinculaciones con la comunidad FLOSS.¹ Adicionalmente aplicamos el criterio *ad hoc* de identificar entre los actores principales empresas orientales con vinculación relevante con FLOSS, dado el peso cada vez mayor de firmas de dicha procedencia en el ámbito tecnológico en general: bajo este criterio fue identificada una compañía, seleccionada como caso de estudio.

Las empresas ajustadas al segundo criterio fueron identificadas en virtud de su visibilidad pública y de su amplio reconocimiento por parte de la comunidad FLOSS.

En función del conjunto de criterios, las compañías conformadas como unidades de análisis son las que se detallan en la Tabla 3.1:

Tabla 3.1. Unidades de análisis

Empresas líderes que contribuyen a FLOSS		Empresas íntegramente FLOSS
Occidentales	Oriental	
Microsoft	Tencent	Red Hat
Google (Alphabet)		Canonical
Oracle		
IBM		

Cabe aquí una aclaración importante sobre el caso de Amazon, que se ajustaría parcialmente al primer criterio de selección por su carácter de líder global en nube pública bajo desarrollo propio, aunque no completamente en cuanto a que no se verifica participación alguna de la empresa en FLOSS, e incluso se observa a través de las fuentes secundarias consultadas cierto desdén por su parte hacia el ámbito del OS. De manera que no ha sido incluida en el corpus, deliberadamente sesgado hacia las empresas que participan en FLOSS.²

1- Como el realizado por la revista FORBES, Fortune y la consultora industrial global Strategy&.

2- La firma es mundial y popularmente conocida por su actividad y liderazgo en comercio electrónico, pero cabe señalar que es también líder al interior del mercado de producción de software en el segmento de nube pública, precisamente en razón de sus capacidades previamente desarrolladas en sistemas de almacenamiento y gestión de información de grandes volúmenes de datos. Desde 2006, con su división Amazon Web Services (AWS) la firma se dedica a proveer almacenamiento y servicios de computación en la nube para gestionar aplicaciones y carga de trabajo, tratándose del

Con la finalidad de indagar sobre las vinculaciones específicas de estas firmas con el FLOSS, el corpus de información obtenida se sistematizó y analizó en torno a los siguientes ejes e interrogantes:

- a) ¿A qué actividades se dedican en el mercado de software? ¿Cuál es el modelo de negocio de cada actor?
- b) ¿Los modelos de negocios difieren entre aquellas firmas que sólo utilizan FLOSS y las que se estructuran en base a productos de software privativo?
- c) ¿Cómo intervienen las productoras de software privativo en el ámbito del FLOSS?
- d) ¿Cuáles son las motivaciones de dicha intervención?
- e) ¿Qué obtienen los actores de su participación en FLOSS?
- f) ¿Se relacionan entre sí? ¿De qué maneras?

En una segunda instancia de la investigación exploramos las características generales del mercado global de software en el que estas firmas se encuentran involucradas: relevancia de cada actor en el mercado, dominios tecnológicos de especialización, tendencias tecnológicas, vínculos de competencia.

En última instancia, la estrategia se dirige a analizar la configuración y virtuales tendencias del mercado global del software en función del vínculo de sus principales actores con el FLOSS.

El texto se estructura del siguiente modo. En la sección siguiente se sistematizan los antecedentes y marco conceptual del estudio. En la tercera sección se presentan los resultados empíricos de la investigación en dos bloques: uno destinado a señalar las principales características y envergadura del mercado global del software en la actualidad y otro que contiene los resultados empíricos por empresa en torno a los ejes mencionados. Por último, en la sección Discusión se analizan los resultados a la luz de los conceptos seleccionados.

1. Antecedentes y marco conceptual: las grandes corporaciones, el FLOSS y la open competition.

Diversos estudios se han desarrollado con el objeto de identificar quienes son los principales contribuyentes de las comunidades *Open Source* (Aaltonen y Jokinen, 2007, Robles *et al.*, 2007) y otros han caracterizado los modos en que las empresas se relacionan con las comunidades (Berdou, 2006, Capra *et*

conjunto de sistemas más usados por las empresas del mundo (consumidoras finales y de software) y los gobiernos hasta el momento (Xataca, 2019a).

al., 2009). A su vez, también se han desarrollado diversos estudios de caso de grandes empresas y su relación con el *Open Source Software* (Lindman *et al.*, 2008, Morgan y Finnegan, 2007, Jaaksi, 2007).

En un estudio realizado por Aaltonen y Jokinen (2007) se muestran los aportes realizados por empresas a la “Linux Kernel Community”³ entre Julio 2005 y Julio 2006. Las 10 firmas que más colaboraron fueron: Steel Eye Technology, IBM, Google, Intel, Novel, OSDL, Cicso, Debian, Alcatel y Red Hat. Por su parte, Robles *et al.* (2007) estudian la contribución de distintas compañías en un proyecto concreto de la comunidad (la distribución Debian GNU/Linux) en un período de 7 años (1998-2005). Para la versión 3.1 del sistema Debian, el top 10 de firmas contribuidoras fue: Sun Microsystems, IBM, Red Hat, Silicon Graphics, SAP, MySQL AB, Netscape Communications, Ximian, Realnetworks y AT&T.

Otros trabajos (Bonaccorsi y Rossi, 2003, Capiluppi *et al.*, 2012, Capra *et al.*, 2009) han avanzado en sistematizar modelos” de involucramiento de las empresas en los proyectos de la comunidad según los tipos de aportes que realizan (ver Capítulo 2). Capiluppi *et al.* (2012) proponen una taxonomía de modos de producción de OS en tres grandes tipos: “*Proprietary Software*”, “*Traditional OS*” (producido en comunidades “puras” sin colaboración de empresas) y “*Sponsored OS*” que implica la participación de empresas en dos modos: “*Industry-led OS*” donde las empresas lideran los proyectos, e “*Industry-involved OS*” donde las empresas aportan en comunidades preexistentes. A través de análisis de sustentabilidad y complejidad estos autores miden los efectos de la participación de empresas en proyectos abiertos, llegando a la conclusión de que aquellos que involucran empresas en cualquier modalidad son más complejos y sustentables en el tiempo que aquellos que no, lo que brinda una pauta importante de la relevancia de esta participación a diferencia de tiempos anteriores en que los proyectos se sostenían predominante y efectivamente sobre la base del trabajo voluntario de desarrolladores independientes.

Por su parte, Forrest *et al.* (2012) plantean que, si bien no se puede responder si la participación de grandes empresas/organizaciones es positiva o negativa para los proyectos/comunidades, sí se observa que el tipo de contribución es claramente diferencial y de mayor envergadura, con sesgos marcados en distintos sentidos:

3- Comunidad en la que desarrolla el código del núcleo del sistema operativo Linux, y participan más de 10.000 desarrolladores y posee más de 20.000.000 de líneas de código.

- En los grandes proyectos, las contribuciones mayoritarias y más importantes son de usuarios corporativos.

- Las corporaciones tienden a contribuir más con código que con reportes de errores (*bugs*), lo que el autor evalúa como menos comprometido con la comunidad/proyecto en sí y más orientado a influir sobre los proyectos en alguna dirección vinculada a sus intereses comerciales.

- En ese tipo de proyectos resulta difícil “trackear” interacciones privadas entre contribuyentes, lo que puede ser indicio de menor transparencia en la gobernanza de este tipo de proyectos, respecto a otros más pequeños o en los que participen menos las corporaciones.

La serie de antecedentes sistematizados es relevante para reconocer el hecho de que las empresas, y en particular las grandes empresas de software privativo, participan intensamente en FLOSS, las comunidades las acogen, y la propia sustentabilidad de estas últimas depende en buena medida de la participación empresarial, configurándose modelos de producción híbridos en términos de procedencia predominante de los contribuyentes.

No obstante, resulta necesario a nuestros fines avanzar en una conceptualización que permita revelar la lógica bajo la cual las empresas líderes del software global abrazan cada vez más estrechamente el FLOSS, en el contexto de una competencia voraz por ganar el mercado. Es decir, proponemos desplazar la mirada desde las comunidades hacia las empresas mismas.

Buena parte de la literatura que aborda la competencia frente a la cooperación pone el acento en los conceptos de propiedad intelectual, concesión de licencias cruzadas, secreto y custodia (Bengtsson y Kock, 2014, Gnyawali y Park, 2009, Ritala y Hurmelinna-Laukkanen, 2013). Es decir, trabaja desde una concepción restringida de las dinámicas de cooperación y competencia.

Teixeira *et al.* (2016) proveen herramienta de utilidad para abordar comprensivamente dicha relación, a veces aparentemente contradictoria. Por un lado, muestran que aunque la competencia y la cooperación han recibido mucha consideración individualmente, se ha prestado poca atención a la cuestión fundamental de la interacción entre los dos conceptos (Chen y Miller, 2015, Iansiti y Levien, 2004a). Los autores abordan el caso de este entorno abierto en el que empresas competidoras cooperan en el desarrollo conjunto de una infraestructura de nube para grandes volúmenes de datos, constatando que allí competencia y cooperación ocurren en forma simultánea. Así, sugieren que la transparencia en el desarrollo y la debilidad de los derechos de propiedad intelectual –es decir, las

características propias de los ecosistemas OS- permiten que las firmas transfieran y reciban información y recursos más fácilmente al interior de múltiples alianzas. A partir de este estudio desarrollan el concepto de “*open-coopetition*”, propuesto anteriormente por Teixeira y Lin (2014), para describir las características y fundamentos de la cooperación entre competidores en el ámbito del software libre.

Teixeira *et al.* (2016) plantean que muchas empresas competidoras, particularmente en la industria de alta tecnología, se han involucrado cada vez más en diferentes tipos de relaciones interorganizacionales de cooperación (alianzas contractuales, empresas conjuntas, consorcios, organismos de normalización y comunidades de código abierto) para mejorar su dotación de recursos y gestionar la incertidumbre estratégica. En consecuencia, las organizaciones están integradas en redes dinámicas y cohesivas en las que las acciones individuales y organizativas están influenciadas tanto por su red como por su posición en la red, que a su vez puede ser fuente importante de ventajas competitivas. Los principales impulsores de la cooperación entre empresas son: la reducción de costes mediante la racionalización de los productos y las economías de escala, la distribución de los riesgos totales y los costes totales de los grandes proyectos, el acceso a nuevos conocimientos y activos complementarios, el desarrollo de la tecnología y el acceso a los mercados complementarios, la co-creación y co-producción del valor, la resolución de los fallos del mercado que surgen en condiciones de racionalidad limitada, la configuración de la competencia con el objetivo de aumentar o disminuir la rivalidad inter-empresarial en el mercado, un acceso más rápido a nuevos mercados, legitimidad y reputación, garantías para la inversión, reducción de la incertidumbre de los requerimientos de recursos. Diversos autores señalan también que hay varios casos conocidos de “coopetencia abierta”, relativamente común en diferentes sectores intensivos en I+D como tecnologías web, *cloud computing*, móviles, de automoción y médicas, entre otras (Teixeira y Lin, 2014, Iansiti y Levien, 2004b, Clarysse *et al.*, 2014). Pero advierten que, si bien la cooperación entre competidores y el software de código abierto son fenómenos con un impacto reconocido sobre cómo se crea, explora y explota el valor en entornos de red, hay muy pocos estudios que aborden cómo las empresas rivales cooperan y compiten de forma simultánea en el ámbito del OSS (Germonprez *et al.*, 2013, Teixeira *et al.*, 2015).

El caso por ellos estudiado, la nube OpenStack, confirma la hipótesis básica de esta literatura de que las empresas “cooperan con sus competidores para ganar”

en un juego de autoservicio para finalmente obtener la ventaja (Brandenburger y Nalebuff, 2011, Chen y Miller, 2015, Hamel *et al.*, 1989).⁴ Y revela algunas peculiaridades fenomenológicas que los autores destacan para señalar la relevancia específica de la “coopetencia abierta” en el ámbito del FLOSS:

- El carácter inclusivo del ecosistema OS, en contraste con la literatura derivada de los joint-ventures y los consorcios de I+D en los que el acceso sólo se concede a unos pocos miembros seleccionados;

- La apertura y transparencia del ecosistema, que contrasta con la literatura sobre las relaciones interorganizacionales que hace hincapié en el mantenimiento de la puerta de entrada y en los derechos de propiedad intelectual;

- El hecho de que los actores cooperan y compiten simultáneamente tanto a nivel interindividual, como interempresarial e inter ecosistémico. Dentro de un escenario de código abierto, es una comunidad abierta (es decir, abierta a las contribuciones de todos) y en red (es decir, empresas relacionadas y desarrolladores independientes) la que dota de estos recursos externos. En este caso, las empresas competidoras cooperaron dentro del ecosistema socio-técnico de OpenStack buscando recursos externos complementarios (Bengtsson y Kock, 2014), para mitigar el riesgo (Gnyawali y Park, 2009) y aprender de otros.

Teixeira *et al.* (2016) también señalan que la noción metodológica de “construcción del ecosistema”, como una forma de hacer más explícitas las interdependencias en red de las empresas, ha ganado prominencia tanto en la investigación como en la práctica (Iansiti y Levien, 2004a, Moore, 1996)

2. Resultados empíricos

2.1. El mercado global del software en la actualidad

La industria del software exhibe un marcado crecimiento en los últimos años. Entre 2004 y 2017 su volumen de mercado se ha más que duplicado. Otro dato que ilustra el crecimiento de esta industria es que 5 empresas del sector se encuentran en el top 80 del ranking Global 2000 de la revista FORBES sobre las compañías más grandes del mundo: Microsoft puesto 16, Alphabet puesto 17, Amazon puesto 28, IBM puesto 60, Tencent puesto 74. Huelga recordar que 4 de ellas conforman nuestra unidad de análisis en cuanto a vínculos intensos con el FLOSS.

4- Ver detalles más abajo.

Una primera constatación empírica que surge al analizar la actividad de los principales actores de la industria del software, es que hay un segmento del mercado en que todos ellos realizan un gran esfuerzo: *cloud computing*.⁵ Según estimaciones de la empresa Oracle, actualmente el *cloud computing* es utilizado por el 40% de las empresas. Según la consultora *Statista*, se espera que el 70% de las firmas llegue a utilizarlo en los próximos años. E incluso otras estimaciones como la de Eyal Manor, vicepresidente de Engineering de Google Cloud, sostienen que alrededor de un 80 % de la carga de trabajo de las empresas en los servidores actuales aún no está en la nube. Es decir que las perspectivas de crecimiento del segmento son muy importantes.

El negocio del *cloud computing* está dominado por un pequeño número de actores, en este orden de liderazgo del segmento:

1) Amazon, pionero en servicios de *cloud computing* (que vende almacenamiento y servicios como por ejemplo Amazon EC2);

2) Microsoft, que comercializa estrategias de nube basadas en su infraestructura de computación en nube Azure (Microsoft Azure Sphere).

3) Google, que vende servicios en torno a su Motor de Cálculo (Google Compute);

Microsoft y Google no ofrecen productos de infraestructura en nube (como Amazon) sino servicios informáticos en torno a estas, de desarrollo y compatibilización (SaaS en *cloud*). Dichos servicios, o al menos los que efectivamente rentabilizan la actividad *cloud* de estas empresas, son los servicios empresariales orientados a empresas desarrolladoras de software, es decir, un círculo que alimenta al propio mercado del software y donde los usuarios principales de las herramientas son desarrolladores, con criterios especializados y exigencias acerca de las características y bondades de aquellas.

Este grado de control sobre el mercado de pocos proveedores de servicios de *cloud computing* condiciona las elecciones de las empresas clientes que, además de buscar el servicio de almacenamiento, buscan compatibilidad e interoperabilidad entre sistemas para evitar esfuerzos de compatibilización y disponer de una canasta de posibilidades para cualquier desarrollo que se presente (Armbrust *et al.*, 2010, Briscoe y Marinos, 2009) sin atarse a ningún proveedor en particular (lo que se conoce como entornos “agnósticos”).

5- El cloud computing, también conocido como servicios en la nube, informática en la nube, computación en la nube o simplemente la nube, es un paradigma que permite ofrecer servicios de computación a través de una red, que generalmente es el internet.

Sorprendentemente o no, las principales alternativas de producto a Amazon EC2, Google Compute y Microsoft Azure no son comerciales, sino cuatro proyectos de código abierto:

- 1) OpenStack.⁶
- 2) CloudStack, respaldado por Citrix y Apache Software Foundation.
- 3) Eucalyptus, un sistema compatible con los servicios de Amazon EC2 y respaldado por numerosas consultoras de TI.
- 4) OpenNebula, más presente en los mercados europeos, y respaldado por C12G, una empresa española.

A su vez, dos de las empresas líderes (Microsoft y Google) no sólo desarrollan sus propias nubes sobre FLOSS, sino que paralelamente contribuyen con recursos financieros y humanos a las comunidades de los proyectos comunitarios de nube arriba mencionados, que compiten con sus propios productos. Al mismo tiempo, el caso de otro gran actor del sector como IBM, que en la década de los '90 abandonó sus esfuerzos previos de desarrollo interno en servidores web para pasar a colaborar con un proyecto FLOSS competidor de quién dominaba el mercado en ese momento (Microsoft), refuerza la visibilidad de esta tendencia.

Como puede verse, el FLOSS juega hoy un papel central en un segmento del mercado de software tan importante como lo es el *cloud computing*, considerando que su presencia se verifica tanto entre los desarrollos *cloud* de empresas líderes del mercado de software, como en desarrollos comunitarios de nubes con amplia difusión y aceptación entre las empresas y desarrolladores a quienes se dirigen este tipo de infraestructuras y herramientas. De todos modos es necesario resaltar el hecho de que el actor que hasta hoy domina el segmento, Amazon, se mantiene dentro del ámbito privativo y alejado del FLOSS.

Otro hecho que ilustra la importancia del papel del FLOSS para las EMN del sector de software es la existencia de la *Open Invention Network* (OIN). Se trata de un grupo dedicado a proteger proyectos de código abierto como Linux a través de un programa de patentes defensivas que permite el uso de licencias cruzadas entre sus miembros sin pago de royalties. A su vez, esta comunidad apoya la libertad de acción en Linux como elemento clave del software de código abierto. Por esto es que, para pertenecer, los miembros acuerdan no hacer valer sus propias patentes contra Linux y sus sistemas y aplicaciones. A esta red se han integrado grandes compañías tecnológicas como Google, IBM, Microsoft, NEC,

6- Ver detalles más abajo.

Philips, Red Hat, Sony, SUSE o Toyota, Canonical, entre otras, en un número que ha crecido aceleradamente desde su formación en 2015 y hoy suma más de 2.600 miembros.

A lo largo de los apartados que siguen desarrollaremos en detalle los tipos de participación en FLOSS de cada una de las compañías que conforman nuestra muestra, que aquí sólo hemos adelantado someramente, lo que permitirá dimensionar con contundencia el peso del FLOSS en sus modelos de negocio y sus estrategias de competencia actuales.

2.2. Las empresas dominantes del mercado global del software con participación relevante en FLOSS

2.2.1. MICROSOFT: del FLOSS como un cáncer a la compra de GitHub ⁷

Microsoft Corporation, fundada en 1975, en la actualidad desarrolla, licencia y provee soporte de software para PCs, servidores y dispositivos electrónicos específicos, algunos de los cuales también manufactura. Sus productos de software más conocidos a nivel de usuarios domésticos son el sistema operativo Windows en sus distintas versiones, la *suite* ofimática Microsoft Office y los navegadores Internet Explorer y Edge. En hardware, los productos estrella son las consolas de videojuegos Xbox y las tablets Microsoft Surface. También produce hoy un amplio rango de otros productos de software menos popularmente conocidos para PC y servidores (por ejemplo en realidad mixta como HoloLens), así como sistemas de computación en la nube (Azure Sphere) y desarrollo de software (Visual Studio) que son utilizados predominantemente por desarrolladores y empresas de software.

Actualmente Microsoft es un operador dominante en el mercado de *suites* ofimáticas y de sistemas operativos para microcomputadoras compatibles con el entorno IBM, aunque en el campo de sistemas operativos –origen y bandera histórica de la compañía- ha perdido buena parte del mercado con la llegada de Android en 2008.

Sin embargo, la propia existencia de Android y los requerimientos para su uso en dispositivos de una diversidad de fabricantes (como Samsung, General Dynamics, Acer, Dell y varios otros) implican ingresos relevantes para Microsoft

7- Para el desarrollo de este apartado nos basamos en las siguientes fuentes documentales: Electrónicos online (2019), GenBeta (2019a, 2017, 2016), Inteldig (2018), My computer (2018, 2014), Xataca (2019a, 2018a, 2018b, 2016).

en concepto de unas 200 familias de patentes de sus softwares, que hasta hoy son necesarias para construir un teléfono Android.⁸ Aproximadamente el 70 por ciento de fabricantes Android pasan por la caja de Microsoft, lo que ha significado un ingreso estimado para la compañía de más de 2.000 millones de dólares anuales, circa 2014.

No obstante esta orientación histórica hacia el software propietario y los sistemas de licenciamiento y patentamiento, en el campo de las herramientas de desarrollo de software así como en el de la dinámica de las licencias es notorio el vínculo cada vez más estrecho de la empresa con el mundo del FLOSS. Partiendo de la memorable intervención del ex CEO Steve Ballmer en 2001 acerca del software libre como un “cáncer”, a partir de 2014 con la llegada del actual CEO Satya Nadella Microsoft ha ido virando no solo hacia un discurso empresarial “FLOSS friendly”, sino transformando efectiva y sensiblemente su modelo de negocio. La empresa ha ido reduciendo paulatinamente su énfasis en sistemas operativos clásicos, licenciamiento y hardware, para enfocarse en la provisión de SaaS por suscripción, en particular el desarrollo para computación en la nube y más recientemente en inteligencia artificial, movimientos que han revitalizado las acciones de la compañía en los últimos años.

La firma hoy conforma sus ingresos a partir de tres grandes divisiones de negocio:

- Intelligent Cloud: servidores y servicios en la nube como Azure, SQL, Windows Server y los servicios de Microsoft Consulting.
- More Personal Computing: recoge los ingresos del sistema operativo Windows, las tablets Surface, el negocio de Xbox y lo generado por el buscador Bing.
- Productivity and Business Processes: ventas tanto corporativas como de consumo de Microsoft Office o los productos de la gama Dynamics (orientados a empresas).

La evolución de los ingresos trimestrales de cada una de esas divisiones desde mediados de 2015 muestra un cambio de tendencia incipiente a favor de Cloud. Y, visto en detalle, el nivel de crecimiento de Cloud supera ampliamente el de las restantes divisiones.

En suma, el giro en el modelo de negocio pasa por generar infraestructuras

8- Son usadas en general por todos los smartphones pero también específicas para el sistema Android, y van desde la interfaz de usuario, gestión de pestañas, interacción con documentos, navegación por la Web, el sistema de archivos exFAT y Exchange ActiveSync.

y herramientas de desarrollo de servicios empresariales compatibles con Linux y eventualmente cualquier otro entorno de desarrollo (multiplataforma). Lo que implica la necesidad de captar y lograr conservar grandes volúmenes de desarrolladores con altas competencias y versatilidad, promovida por las condiciones de un mercado global del software cada vez más competitivo en términos de recursos humanos y cada vez más dinámico en volumen de desarrollos (en parte por la lógica de la generación constante de “aplicaciones”). Como consecuencia, dada la importante inclinación de los desarrolladores -profesionales y aficionados- hacia el FLOSS en todo el mundo, junto al avance de las firmas competidoras en modelos de negocio que hoy integran activamente el mundo FLOSS como Google e IBM, Microsoft ha pasado a congratularse con este mundo.

Las intervenciones históricas más notables de Microsoft en FLOSS han sido las siguientes:

- En 2016 la firma compró Xamarin, una empresa basada en FLOSS y muy valorada por la comunidad OS, que desarrolla una *suite* de herramientas con las que se puede programar código una vez y luego ser exportado a otros sistemas, con la peculiaridad de generar aplicaciones nativas para el resto de los sistemas. Esto es, software de desarrollo multiplataforma (compatible e interoperable con cualquier otro sistema).

- En el mismo año, Microsoft se unió a la Fundación Linux en calidad de miembro Platino. Y también patrocina iniciativas similares como Apache Software Foundation y Open Source Initiative.

El año 2018 fue el de consolidación de la integración de Microsoft al mundo del FLOSS:

- Con la liberación de unas 60.000 patentes, la empresa se integró como miembro de pleno derecho a la Open Invention Network (OIN). Microsoft no abrirá por el momento el código de Windows, de Office o de sus aplicaciones privativas más difundidas, pero sí algunas patentes relevantes como las utilizadas previamente para presionar precisamente a los fabricantes de Android.

- Dentro del ecosistema Azure Sphere, en 2018 año la empresa lanzó la primera distribución Linux de su historia: Azure Stack OS, un sistema operativo personalizado para IoT (*Internet of Things*). Gran parte de la computación en la nube de Microsoft se basa en máquinas virtuales, contando con soporte para Linux, Windows Server, SQL Server, Oracle, IBM o SAP, y un amplio

catálogo de configuraciones. En los orígenes de Azure, hace unos tres años, Linux representaba una de cada cuatro instancias en el sistema. En 2017, el 40% de máquinas virtuales de Azure ya funcionaban con Linux, mientras que a finales de 2018 las máquinas virtuales de Linux en Azure ya suponían prácticamente la mitad. Esta es la primera vez que Microsoft no solo desarrolla una versión pública de Linux sino que la convierte en la piedra angular de una oferta de productos: una familia de microcontroladores certificados Azure Sphere, personalizados y diseñados directamente por Microsoft; un circuito integrado programable bajo arquitectura ARM conectado a Internet, que combina procesadores de aplicaciones en tiempo real con conectividad y tecnología de seguridad por hardware; un servicio de seguridad basado en el Cloud de Microsoft que protegerá todos los dispositivos bajo esta plataforma.

- En suma, más de 4000 empleados de Microsoft en todo el mundo contribuyen en más de 2000 proyectos OS.

- Como corolario, Microsoft adquirió la plataforma de desarrollo y distribución de proyecto OS GitHub (creada por uno de los precursores del movimiento FLOSS, Linus Torvalds) por 7.500 millones de dólares, una de las operaciones económicas más costosas de su historia.⁹ Si bien Microsoft contribuía ya desde 2013 con soporte en Visual Studio y la respaldaba como su plataforma distribuida de control de código fuente, hoy en día casi todos los programadores de Microsoft trabajan con GitHub, marca que sigue manteniendo y operando de forma independiente.

2.2.2. GOOGLE. Inteligencia artificial y apuestas a pérdida inmediata = FLOSS¹⁰

Google, popularmente conocida a partir de su correo electrónico Gmail, la plataforma audiovisual Youtube o el sistema operativo más usado del mundo Android, es parte de su empresa matriz Alphabet, agencia de publicidad que como ya vimos es hoy una de las compañías más valiosas del mercado global.

Es sabido también que en términos generales el entorno Google es abierto, y en particular que el ecosistema Android es de tipo *open source*, a la vez que

9- Microsoft pagó 26.200 millones de dólares por LinkedIn en diciembre de 2016 y 8.500 millones de dólares por SKype en mayo de 2011. Otras adquisiciones como las de Nokia (7.200 millones de dólares) en septiembre de 2013 y aQuantive (6.333 millones de dólares) en agosto de 2007) se sitúan en ese ranking de operaciones de adquisición.

10- Para el desarrollo de este apartado nos basamos en las siguientes fuentes documentales: GenBeta (2019b), Google, Python, Xataca (2019b, 2017).

con el paso del tiempo Google ha construido toda una capa de personalización propietaria sobre la base de Android. La empresa es también miembro Platino de la Fundación Linux.

La mayoría de los ingresos de Alphabet provienen del servicio de publicidad Google AdWords, el núcleo de la compañía. Esta plataforma nace en los resultados de búsquedas del navegador de internet Google, donde los anunciantes llegan con un sistema de puja por las posiciones más altas, y cruza todos los servicios de la empresa: Gmail, YouTube, Maps, etc. El click realizado en estos resultados se paga a precios que varían en función del producto ofrecido y de la categoría en la que se encuentre. Junto a AdWords se encuentra la plataforma AdSense, que ofrece a cualquier página web insertar módulos de publicidad gestionados por Google cuyos ingresos (por impresión o por click) se reparten entre la empresa y el administrador de la web. Juntas, estas plataformas rentabilizan el conjunto de los productos de Alphabet, cuya división “otras apuestas” (es decir, no Google) incluye la venta de productos como los Chromebooks, suscripciones a almacenamiento en la nube, la plataforma Google Cloud, los ingresos obtenidos por Android, los Chromecast o la Play Store. La división aglutina también todos los proyectos experimentales que todavía forman parte de su banco de pruebas: Google Fiber (fibra y telefonía en el hogar), Nest (domótica), Verily (aplicaciones científicas para “mejorar la vida de las personas”), Calico (lucha contra el envejecimiento y sus enfermedades asociadas), autos eléctricos, y otros.

La división se encuentra habitualmente en pérdidas, dado que los proyectos no se convierten en empresas hasta que la matriz traza un plan de rentabilidad para ellas. No obstante, los millones de dólares ingresados por vías no publicitarias están registrando últimamente un crecimiento proporcionalmente mayor que el de la propia publicidad (un 36,5% de un año para otro entre 2017 y 2018). La tendencia indica una incipiente diversificación en el modelo de negocio de Alphabet.

El actual CEO de Google, Sundar Pichai, explica constantemente que el núcleo de producto de la compañía es el avance en inteligencia artificial, para publicidad y para el resto de las actividades. En este marco, una de las más fuertes apuestas actuales de la compañía son los servicios en la nube. Google Cloud Platform (GCP) se encuentra bastante distanciada de sus dos principales competidores (Amazon y Microsoft) debido a su entrada tardía en el negocio de IaaS, su limitado catálogo de productos, su enfoque histórico alejado del mundo empresarial y el no disponer de una infraestructura de *data centers* en tantas

regiones como AWS (de Amazon) o Azure. El uso típico de GCP por parte de sus clientes sigue siendo experimental o como segundo proveedor para ahorrar costos en algunos servicios secundarios. Pero, según las previsiones de la firma, el 88 % de las empresas en 2024 combinarán distintos proveedores, y es allí donde Google, con menos servicios que su competencia, ve la oportunidad de rentabilizar su experiencia diferencial con contenedores en la nube para convertirse directamente en el administrador de las diferentes infraestructuras, más que en la comercialización específica de máquinas virtuales. Google ha aplicado todos sus conocimientos para escalar y balancear grandes infraestructuras de sistemas ofreciendo herramientas lo más similares posibles a las que las empresas usan internamente. De este modo, sus productos principales destacan por el uso de *big data*, herramientas de analítica o *machine learning*.

En este camino, en 2015 Google lanzó Kubernetes, un sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores que fue originalmente diseñado por Google y donado a la Cloud Native Computing Foundation (parte de la Linux Foundation).¹¹ Soporta diferentes entornos para la ejecución de contenedores, y es hoy la plataforma en la que se basa toda la infraestructura agnóstica apoyada en contenedores fácilmente replicables y escalables.

Más recientemente Google lanzó Anthos (anteriormente denominado Cloud Services Platform), plataforma que permite compilar y administrar aplicaciones híbridas en diferentes entornos de nube. Funciona con Kubernetes y otras tecnologías de Google de código abierto, líderes en esta industria. De este modo Google ofrece en un único servicio la gestión, despliegue y monitorización de los distintas plataformas en la nube que el usuario quiera tener, incluso su propia infraestructura de hardware fuera de la nube (Google pretende con ello gestionar también, por supuesto, las aplicaciones en AWS y Azure, en un movimiento que se anticipa a sus competidores).

11- Los contenedores de aplicaciones (*docker*s) son entornos ligeros de tiempo de ejecución que proporcionan a las aplicaciones los archivos, las variables y las bibliotecas que necesitan para ejecutarse, maximizando de esta forma su portabilidad. Si bien las máquinas virtuales (VM) tradicionales permiten la virtualización de la infraestructura de computación, los contenedores habilitan la de las aplicaciones de software. A diferencia de las máquinas virtuales, los contenedores utilizan el sistema operativo (SO) de su host en lugar de proporcionar el suyo propio.

No es usual para los grandes competidores lanzar productos como Anthos que permiten a los usuarios ejecutar sus aplicaciones en otras plataformas, dado que las ganancias provienen de los costes del tiempo de computación usados y el almacenamiento, lo que significa que las empresas competidoras pueden apropiarse de dichas ganancias habiendo evitado directamente sus costos. Pero, según argumenta Google, este tipo de herramientas es lo que sus usuarios desarrolladores están demandando para afrontar el gran problema de tener una arquitectura híbrida con distintos productos que no se pueden comunicar o gestionar de forma conjunta. Aunque Google ceda las horas de computación o el espacio de almacenamiento a sus competidores, si la estrategia resulta exitosa podrá ser la que maneje realmente las necesidades de las compañías clientes para ser el proveedor del siguiente servicio que demanden o fomentar la migración utilizando un panel de gestión ya familiar.

Al mismo tiempo, en este 2019 Google anunció la colaboración con un gran número de compañías relevantes del OS (como Confluent, DataStax, Elastic, InfluxData, MongoDB, Neo4J y Redis Labs) cuyas herramientas serán empleadas por los usuarios que lo deseen, además de recibir el soporte oficial de cada una de ellas.

En suma, Google aporta en más de 2000 proyectos de código abierto, y tiene a más de 2000 colaboradores interviniendo en comunidades FLOSS, que no es un número bajo considerando que sus contribuciones tienden a considerarse muy fuertes y relevantes técnicamente. Como estrategias complementarias para fortalecer su labor hacia los programadores, la empresa organiza maratones anuales como los Google Code-In, y otorga becas trimestrales de ayuda económica y mentores a estudiantes de informática (desde la creación de este programa en 2005 más de 14.762 estudiantes de 109 países han trabajado en 651 proyectos de código abierto bajo esta modalidad).

Por último, la empresa es también miembro precursor de la Open Invention Network, con su política de apertura de patentes.

2.2.3. IBM / RED HAT: liberar y adquirir líderes íntegramente FLOSS ¹²

El objeto de este apartado es la nueva empresa recientemente conformada a partir de la adquisición por parte de IBM de la mundialmente reconocida empresa de OS Red Hat.

12- Para el desarrollo de este apartado nos basamos en las siguientes fuentes documentales: Canal Ar (2019a, 2019b), IBM-Red Hat, Red Hat (2019).

International Business Machines Corporation (IBM) es una multinacional estadounidense de tecnología que fabrica y comercializa hardware y software para computadoras y ofrece servicios de infraestructura, alojamiento de Internet y consultoría informática. Desde sus orígenes como empresa orientada a las ventas (fundada en 1924) fue reorientándose hacia un perfil técnico, que terminó de consolidarse hacia la década de los 2000. En los últimos años el modelo de negocio de IBM y su cultura empresarial han virado específica y significativamente hacia la producción de software y servicios asociados, y en la actualidad la empresa es una de las principales defensoras, sponsors y agentes del movimiento de código abierto. IBM invierte miles de millones de dólares en servicios y software basados en GNU/Linux, a través del IBM Linux Technology Center (LTC) que incluye alrededor de 300 empleados de IBM trabajando en el núcleo Linux. Los desarrolladores del LTC contribuyen a varios proyectos de código abierto, como máquinas virtuales, Apache Hadoop, OpenStack, OpenPower Consortium, GNU toolchain y estándares de código abierto. IBM también ha lanzado diversos códigos bajo licencias abiertas, tales como el framework multiplataforma Eclipse, la licencia International Components for Unicode (ICU) y el sistema de gestión de bases de datos relacionales Apache Derby basado en el lenguaje de programación Java. Este último constituye uno de los proyectos de mayor envergadura, para el cual IBM liberó (500.000 líneas de código) de su programa Cloudscape a la Fundación Apache.

Pero el movimiento de mayor peso en la historia de IBM respecto al campo del FLOSS, ha sido hasta el momento la adquisición en julio de 2019 de la firma Red Hat (fundada en 1993), proveedora líder mundial de soluciones de TI empresariales de código abierto. Red Hat ofrece tecnologías de nube, Linux, gestión, middleware, mobile, almacenamiento, virtualización, soporte, capacitación y servicios de consultoría, todo íntegramente OS, a través de un modelo de suscripción. El modelo de negocio de Red Hat se basa principalmente en su capacidad para proporcionar una versión fácil de usar y accesible de Linux mediante la producción de productos empaquetados del sistema operativo, a la vez que proporciona servicios y atención al cliente que se adaptan a sus productos. La eficacia de este modelo de suscripción se basa, en gran medida, en dos factores interrelacionados: su reconocimiento como proveedor de productos y servicios FLOSS de confianza, así como la posición de institución legalmente reconocida, que puede ser considerada responsable por los productos y servicios que ofrece (Birkinbine, 2017). El modelo de negocios histórico de Red Hat dependía del

soporte técnico que desde la firma se le brinda al consumidor y el éxito de su estrategia ha estado en lograr comprender que su fuerte no es la comercialización y el desarrollo de software, sino proveer un servicio con valor agregado en el soporte de soluciones adaptadas a sus clientes (Munga, Fogwill y Williams, 2009).

Fundada en 2003, Red Hat surge con el propósito de lograr que el FLOSS se posicionase como un negocio comercialmente viable. El modelo de negocios que implementaron ha mostrado un crecimiento notable desde sus inicios, al punto de ubicarse como una de las 25 empresas tecnológicas con crecimiento más rápido de Estados Unidos. Este gran crecimiento se vio potenciado desde 1999 cuando la empresa comenzó a comerciar sus acciones en el mercado de valores, logrando alcanzar un valor de \$150 billones de dólares al poco tiempo. Además, en 2006 la empresa adquiere a JBoss, la segunda compañía más grande de FLOSS, lo que le permitió convertirse en un proveedor de servicios mucho más completo, logrando así clientes de gran magnitud como Amazon, DreamWorks, Morgan Stanley, entre otros (Ibídem).

La operación de adquisición de Red Hat por parte de IBM significó la compra de todas las acciones en circulación emitidas de Red Hat a U\$S190,00 por acción en efectivo, lo que representó un valor total de 34.000 millones de dólares.

Juntas, IBM y Red Hat ofrecerán una plataforma *multicloud* híbrida de próxima generación. La adquisición intenta posicionar a IBM como proveedor líder de nube híbrida. A su vez, provee a Red Hat de su histórico conocimiento industrial y liderazgo comercial en más de 175 países.

La plataforma, basada en tecnologías de código abierto como Linux y Kubernetes (de Google) permite a las empresas desplegar, ejecutar y gestionar de forma segura datos y aplicaciones tanto in situ como en nubes públicas múltiples y privadas. Respecto a la oportunidad de este nuevo negocio resulta importante e interesante revisar el posicionamiento público de las mismas empresas:

Las organizaciones están comenzando el nuevo capítulo de su reinvencción digital. Están modernizando la infraestructura y moviendo cargas de trabajo de misión crítica a nubes privadas y nubes múltiples de diferentes proveedores”, “Necesitan tecnología flexible y abierta para gestionar estos ambientes multinube híbridos. Y necesitan socios en los que puedan confiar para gestionar estos sistemas de forma segura. IBM y Red Hat son únicas para satisfacer estas demandas. Como el proveedor líder de nube híbrida, ayudaremos a los clientes a construir los pilares tecnológicos de su negocio para las próximas décadas.

Ginni Rometty, presidenta y consejera delegada de IBM (Red Hat, 2019).

Cuando hablamos a los clientes, sus desafíos están claros: necesitan moverse más rápido y diferenciarse a través de la tecnología. Quieren construir unas culturas más colaborativas, y necesitan soluciones que les den la flexibilidad para construir y desplegar cualquier app o carga de trabajo en cualquier lugar”, “Creemos que el código abierto se ha convertido en el estándar tecnológico de facto porque permite estas soluciones. La unión de fuerzas con IBM da a Red Hat la oportunidad de llevar más innovación en código abierto a una aún mayor variedad de organizaciones y nos ayudará a expandirnos para cubrir la necesidad de soluciones de nube híbrida que proporcionen una verdadera capacidad de elección y agilidad.

Jim Whitehurst, presidente y consejero delegado de Red Hat (Ibídem, resaltado nuestro).

Al mismo tiempo resulta importante revisar de qué manera se condicen estas previsiones sobre el comportamiento de las empresas clientes con lo que efectivamente la transformación del modelo de negocio ha reportado a IBM: los ingresos de IBM en la nube han pasado del 4% del total en 2013 al 25% hoy. Este crecimiento procede de una gran variedad de software y ofertas en modo servicio, servicios y hardware que permiten a IBM construir, mover y gestionar soluciones en la nube para los entornos públicos, privados o in situ de los clientes y asesorarlos. Los ingresos del negocio *cloud* de IBM durante los 12 meses transcurridos hasta marzo de 2019 crecieron hasta los 19.000 millones de dólares. La empresa prevé que la adquisición de Red Hat contribuya con aproximadamente 2 puntos de crecimiento de ingresos anual compuesto durante un periodo de cinco años.

A medida que las organizaciones buscan incrementar su velocidad de innovación para ser más competitivas, optan por el código abierto y un entorno en la nube distribuido para posibilitar una nueva ola de innovación digital que no era posible antes....

Frank Gens, vicepresidente senior y analista jefe de IDC (Ibídem, resaltado nuestro).

El modelo de negocio de Red Hat, pionero en sus orígenes, contribuyó a popularizar entre las empresas las tecnologías de código abierto -como Linux, Kubernetes, Ansible, Java, Ceph, etc. Y Linux es hoy la plataforma de desarrollo más utilizada en el mundo. Ello vuelve más clara la oportunidad y pertinencia de la adquisición, desde una óptica de crecimiento empresarial en el nuevo contexto del mercado global del software. En esta dirección también puede decirse que IBM opera de modo similar a Google en cuanto a la estrategia de “adelantarse” al mercado, en este caso de nube empresarial híbrida, en términos de entornos de gestión agnósticos.

Desde tiempos previos a la adquisición, tanto IBM como Red Hat ya revistaban como socias fundadoras de la Open Invention Network y colaboraban con la Fundación Linux (IBM en calidad de socio Platino y Red Hat de Plata).

Otro dato relevante a nuestros fines es que IBM preserva la independencia y neutralidad de Red Hat. Y Red Hat continuará construyendo y expandiendo sus alianzas, incluyendo aquellas con los principales proveedores de *cloud*, como Amazon Web Services, Microsoft Azure, Google Cloud y Alibaba.

2.2.4. ORACLE: de amenaza a ¿soporte? del FLOSS¹³

Oracle Corporation es una empresa con base en Estados Unidos, donde su principal actividad es el desarrollo de soluciones en la nube. Entre el 1 de octubre de 2017 y el 30 de septiembre de 2018 tuvo ingresos cercanos a los 40.000 millones de dólares. A su vez, se posicionó en el puesto 21 entre las empresas más innovadoras de 2018 en el ranking elaborado por la consultora Strategy&. ¹⁴

Su producto principal es la nube “Oracle Cloud” a partir de la cual presta múltiples servicios, ubicándose como uno de los grandes participantes del mercado de servicios en la nube para desarrolladores. En términos globales se encuentra en un segundo lote en conjunto con IBM, Alibaba y otras, después de Amazon, Microsoft y Google que dominan el segmento. Dando muestras de la estrategia de cooperación con otras empresas por parte de Oracle, Luis Meisler, vicepresidente ejecutivo de la compañía para América latina, sostenía en 2010:

13 - Las fuentes documentadas utilizadas para este apartado fueron: Computer world (2016, 2018), Computing (2004), Electronic online magazine (2019), El País (2009), FORBES, Fundación Linux, La Nación (2010), Muy Linux (2017), Openstack, Oracle, Xataca (2017b).

14- Este período corresponde al año fiscal 2018 en Estados Unidos.

Creo que nuestros competidores hoy son IBM, SAP y Microsoft. Pero a la vez IBM es socia de Oracle, y SAP tenía contratos con Sun para la parte de hardware que vamos a querer mantener. Promovemos el desarrollo del ecosistema, hacemos negocios con los que quieran trabajar con nosotros; si vendemos más nos conviene.

A su vez, Oracle lanzó en Octubre de 2017 una base de datos automatizada denominada “Oracle Autonomous Database” a partir de la cual comenzó a prestar servicios en abril de 2018. Tanto Oracle Cloud como Oracle Autonomous Database son de descarga libre y son acompañadas de una licencia de desarrollador, la cual permite hacer incrustaciones y distribuir el software. Este tipo de acceso al software no incluye soporte técnico ni telefónico por parte de Oracle.¹⁵

A su vez, Oracle se caracterizó por concretar una serie de adquisiciones en los primeros años del siglo XXI. Un caso icónico en este aspecto fue la adquisición de Sun Microsystems en 2009, operación que se realizó por más de 7000 millones de dólares. la operación fue muy resonante porque en el momento en que fue adquirida por Oracle, Sun Microsystems era un ícono del software libre, sumado a que con esta compra Oracle acercó su negocio a la nube y se estableció como jugador clave en el mundo Open Source. La compra fue recibida con cierto resquemor por parte de la comunidad FLOSS debido a que Oracle pasaba a monopolizar así gran parte del mercado de base de datos, contribuyendo aún más a esta desconfianza el hecho de que la compra fuera acompañada por declaraciones de ejecutivos de alto rango que planteaban que su política de adquisiciones se justificaba en poder adquirir una gran cantidad de productos que les permitieran disminuir los costos de compatibilidad e integración en sus entornos.¹⁶ Es decir, una valoración particular del OS por parte de la compañía no aparecía en el discurso público, lo que generó fundadas incertidumbres acerca de la continuidad de los proyectos abiertos que sostenía Sun Microsystems. Entre los más fuertes estaban la aplicación JAVA que en 2006 había adoptado una licencia de código abierto, la licencia GPLv2 y la base de datos MySQL. Alrededor de la última se generó un movimiento dentro de la comunidad de FLOSS identificado como

15- En Argentina el grupo Clarín utiliza la base de datos de Oracle para procesar la información existente en torno a la empresa (por ejemplo datos de las transacciones que hacen los suscriptores, o sus reclamos) y darle formato de negocios.

16- Declaraciones de Luiz Meisler, vicepresidente ejecutivo de Oracle para América latina al diario La Nación en mayo de 2010.

“save MySQL” que derivó en que Oracle tuviera que mantener una licencia dual para la misma. A su vez, luego de perder un litigio judicial con Google en el que Oracle reclamaba el pago de licencias por la utilización de JAVA, en 2017 la empresa propietaria terminó donando la aplicación a la Fundación Eclipse. La misma suerte corrió OpenOffice, cedido a la misma Fundación. Sin embargo otros proyectos fueron efectivamente desmantelados, como fue el caso de Open Solaris y SPARC. Open Solaris, fue un sistema operativo de software libre que dejó de desarrollarse en mayo de 2010 a partir de la compra de Sun Microsystems por parte de Oracle. SPARC, por su parte, fue una línea de procesadores de código abierto desarrollada por Sun Microsystems en 1987 y que tuvo distintas versiones hasta la última lanzada en 2004 en conjunto con la firma japonesa Fujitsu.

Como se ve, Oracle se ha presentado por momentos como un “enemigo” de la comunidad FLOSS. Sin embargo, su relación es ciertamente ambigua, teniendo en cuenta que es miembro Platinum de la fundación Linux, que aporta a OpenStack como “organización de apoyo”, y a su vez sostiene su propia comunidad de desarrolladores “Groundbreakers Community” en la cual hay sub-espacios específicos tanto para la *Oracle Cloud* como para la *Oracle Autonomous Database*.¹⁷ Oracle también organiza eventos para desarrolladores como por ejemplo el “Oracle developer Meetup” en el cual se tratan temas relacionados al FLOSS.

2.2.5. SAP: del software privativo de gestión al desarrollo abierto en la nube¹⁸

SAP es una empresa alemana fundada en 1972. En el último año (a julio de 2019) tuvo ingresos cercanos a los 30000 millones de dólares. Su actividad inicial, y la más importante, es el desarrollo y venta de módulos “enlatados” para gestión empresarial. En 2012 se estimaba que su producto era utilizado por entre el 70% y 80% de las grandes empresas del mundo.¹⁹ A su vez, SAP también compete -principalmente en a partir de fines de la década de los 2000- en el mercado de los negocios en la nube. En este sentido en 2010 compró la firma

17- El rótulo “organización de apoyo” abarca diversas maneras de apoyar la misión de OpenStack, ya sea que esté contribuyendo con código, creando un producto OpenStack o ayudando a construir la comunidad.

18- Este apartado fue construido en base a las siguientes fuentes documentales: Computer World (2010), El País (2012, 2012b), Forbes, Fundación Linux, SAP.

19- Declaraciones de Javier Sevillanos, de la Universidad Complutense de Madrid al diario El País de España. https://elpais.com/sociedad/2012/08/27/actualidad/1346089934_596891.html

de software empresarial, Ariba por más de 3500 millones de dólares, emulando la estrategia de Oracle, uno de sus principales competidores en el mercado de software para empresas. En el mismo año SAP también adquiere distintas firmas con el fin de proveerse de bases de datos propias (hasta ese momento usaba bases de otros actores como Microsoft, IBM y Oracle, entre otros). En el marco de estas adquisiciones, compró la firma FLOSS Sybase en alrededor de 5000 millones de dólares.

Al mismo tiempo se relaciona con el FLOSS por ser miembro de plata de la Fundación LINUX, y además “patrocinador corporativo” de OpenStack donde proporciona fondos para apoyar la misión de proteger, potenciar y promover esa nube. A su vez, es miembro de otros proyectos de FLOSS como son OpenAPI Initiative, World Wide Web Consortium y DENIC.

SAP también ha trabajado de manera conjunta con Oracle en el ámbito de las bases de datos; esto se observa en la versión de Oracle Exadata para usuarios de SAP, proyecto que surgió en 2010 y en la actualidad ya cuenta con la versión 6.0 de la misma. Esto se manifiesta también en que hay desarrolladores de SAP trabajando en la comunidad de desarrolladores de Oracle “Groundbreakers Community”. Este aspecto es sumamente interesante porque muestra cómo SAP, coopera con Oracle, firma con la cual en el segmento de software empresarial compitió tradicionalmente a través de adquisiciones.

2.2.6. TENCENT: el gigante chino del entretenimiento digital²⁰

Tencent es un proveedor líder de servicios de Internet en China, que ofrece algunos de los sitios web, aplicaciones y servicios más populares de ese país como QQ, Qzone, Tencent Cloud y el homólogo de Whatsapp, WeChat.

Es también la mayor compañía de videojuegos del mundo. Es dueña del accionariado de las productoras de videojuegos de liga mundial como Supercell y Epic, lo que la posiciona como dueña de los juegos Fortnite, Unreal y Clash of Clans. Es también dueña de Riot Games, es decir que es dueña del LOL. Y posee acciones y acuerdos comerciales con Ubisoft, Activision, Square Enix. Todo lo que implica que posee una gran porción de los juegos más usados del mundo.

En el mercado de China, supera el 50% de la cota del mercado de videojuegos en usuarios y facturación. La división de videojuegos de Tencent en 2018 generó 13.000 millones de euros.

20- En este apartado nos basamos en las siguientes fuentes documentales: Fundación Linux (2018), Xataka (2019).

El modelo general de negocios de la firma involucra también una plataforma de vídeo en streaming (homóloga de Netflix), una de música (homóloga de Spotify), la empresa también desarrolló la mayor plataforma online de literatura y cómic en China, y posee una división de realizaciones cinematográficas (Tencent Pictures).

Por todo lo dicho se la conoce en el mercado como el gigante chino del entretenimiento digital.

Para esta variedad de áreas de actuación Tencent está involucrada en el desarrollo de software a gran escala, y ha mostrado una creciente dedicación al software libre en los últimos años. Ha utilizado Linux extensamente durante muchos años, fue una de las primeras compañías en aprovechar el controlador abierto OpenDaylight SDN y fue uno de los fundadores del comité China Open SDN (COS). Tencent patrocina docenas de proyectos de código abierto en GitHub, y es miembro de la Cloud Native Computing Foundation, Hyperledger, LF Networking y la Open Networking Foundation. Además, es uno de los miembros fundadores de la recientemente creada LF Deep Learning Foundation, que se centra en apoyar diversos esfuerzos técnicos en inteligencia artificial en el campo FLOSS.

En junio de 2018 la Fundación Linux anunció el ingreso de la firma como miembro Platino, con lo cual uno de sus gerentes se unió a la Junta Directiva, y planteó en aquel momento: *El código abierto es el núcleo de la estrategia técnica de Tencent*” (Liu Xin, director general grupo de Internet móvil de Tencent (Fundación Linux, 2018).

La contribución ampliada de la firma a la comunidad Linux comenzó con un proyecto de microservicios de código abierto TARS, un proyecto de servicio de nombres de código abierto Tseer, y el proyecto de código abierto AI Angel especialmente orientado a *machine learning*.

Analistas del sector IT a nivel global señalan que si bien Tencent ha avanzado notoriamente en el mercado global del software, como se observa con claridad a partir de los datos brindados, y es líder absoluta en su enorme mercado, nacional, aun se encuentra lejos de competir con éxito con las líderes globales estudiadas.

2.3. El caso de una empresa líder íntegramente FLOSS: Canonical²¹

Canonical Ltd. es una compañía británica fundada en 2004. Su principal producto, y por el cual es mundialmente reconocida, es el sistema operativo de código abierto Ubuntu. Ubuntu es distribuido bajo licencia abierta y es el sistema operativo más popular dentro de las distribuciones Linux. Según declaró el fundador de Canonical, Mark Shuttleworth, en 2018 seis de los veinte principales bancos del mundo usaban Ubuntu OpenStack, por su seguridad y estabilidad. En 2016 Canonical estimaba que Ubuntu tenía más de 25 millones de usuarios.

El modelo de negocios de Canonical está íntegramente basado en la venta de servicios y soporte a partir de software de código abierto. Cuenta con más de 200 empleados a tiempo completo, y tiene ingresos anuales cercanos a los 30 millones de dólares.

Dentro del mercado de *cloud computing*, Ubuntu se posiciona a través de *Ubuntu Cloud*, nube a partir de la cual provee servicios como almacenamiento, potencia informática, infraestructura de red y software, a través de Internet, que se facturan por el tiempo utilizado. *Ubuntu Cloud* utiliza el software de código abierto OpenStack. En noviembre de 2015, según datos de Canonical, había más de dos millones de usuarios de Ubuntu en la nube a través de *Ubuntu Cloud*.

Por su propia constitución, Canonical está estrechamente vinculada al FLOSS y a su vez apuesta fuertemente al mismo en sentido amplio. Es miembro de plata de la Fundación Linux, organiza eventos para desarrolladores, participa a su vez de múltiples proyectos relacionados al FLOSS, entre los que se encuentran algunos muy importantes como OpenStack y Debian.

A su vez, desde la empresa plantean explícitamente sus convicciones respecto la importancia del software libre:

Creemos en el poder del software de código abierto; Ubuntu no podría existir sin su comunidad mundial de desarrolladores voluntarios. Estamos comprometidos a crearlo, refinarlo, certificarlo para la confiabilidad y promover su uso. Además de lanzar y ejecutar nuestros propios proyectos, aportamos personal, código y financiación a muchos más.

(Página web de Canonical)

21- Este apartado se basa en las siguientes fuentes documentales: Canonical, Forbes (2016), Muy Linux (2016), Muy computer (2015), Openstack, Ubuntu, Fundación Linux.

Por último, desde 2010 la empresa es miembro asociado de la Open Invention Network.

3. Discusión

A partir del recorrido realizado sobre las grandes empresas globales de software y su vinculación con el FLOSS, es posible trazar un conjunto de conclusiones y reflexiones en torno a dos grandes dimensiones del mercado global del software actual que sistematizamos a continuación: A) la presencia constitutiva y co-predominante del FLOSS en el mercado; B) cambios en los procesos de competencia entre las empresas ligados a la extensiva importancia del FLOSS.

A) Presencia constitutiva y co-predominante del FLOSS en el mercado:

(i) Apuesta de los líderes por el negocio del *cloud* empresarial:

Como se ha visto, el segmento de *cloud computing* alberga un potencial de desarrollo exponencialmente mayor al actualmente materializado en el ámbito del software empresarial, lo que constituye al *cloud* como un nicho de apuesta para las empresas líderes que buscan ganar porciones mayores del mercado de software hoy orientado a empresas.

Vinculado a lo anterior, las estrategias de los actores líderes con fuerte presencia en FLOSS orientadas a *cloud* apuntan especialmente a la demanda de las empresas de desarrollo de software, que pagan por acceder a las infraestructuras y herramientas de desarrollo que su escala no les permite garantizar internamente (más allá de que usuarios domésticos también hagan amplio uso y aprovechamiento económico de las mismas herramientas), con preferencia por entornos agnósticos que habiliten la gestión centralizada de múltiples plataformas. Es decir que participar en el mercado del *cloud* es hoy intervenir en el negocio del desarrollo de software, más allá de que cada una de las empresas fuertes en este campo ofrezca también productos de software para consumidores finales y/o usuarios domésticos.

(ii) Presencia constitutiva del FLOSS en el ámbito del *cloud computing*:

El FLOSS atraviesa constitutivamente el ámbito del *cloud computing*. Entre las razones principales por las cuales el FLOSS atraviesa de tal modo este segmento figuran la calidad técnica de sus desarrollos en infraestructuras y herramientas, la estabilidad y seguridad del software, la necesidad de compatibilidad entre aplicaciones y empresas proveedoras, y la amplia disponibilidad de recursos humanos de alto nivel -desarrolladores- que se encuentra en el ámbito FLOSS.

Esta presencia constitutiva del FLOSS en el segmento *cloud* se pone de manifiesto en virtud de tres constataciones: la apuesta de las compañías competidoras de Amazon por desarrollar sus entornos íntegramente sobre herramientas OS (Microsoft, Google, IBM); las apuestas a *cloud* por parte de las grandes compañías íntegramente FLOSS (Canonical y Red Hat); y el éxito de determinados grandes proyectos *cloud* de carácter comunitario (OpenStack y GitHub principalmente).

(iii) Co-predominancia del FLOSS en la industria global de producción de software empresarial:

En virtud de 1 y 2, y sumando aquí un elemento central que resalta a lo largo del recorrido por las diversas empresas líderes como es el de las grandes adquisiciones de empresas OS, puede decirse que el FLOSS no solo forma parte de las estrategias actuales de negocios de la mayoría de las empresas líderes sino que ocupa un lugar preponderante en ellas. El FLOSS constituye hoy un elemento que ha adquirido predominancia en la industria global de la producción de software empresarial junto a los segmentos privativos de desarrollo dominados por Amazon.

Consideramos que un *locus* central de esta disputa focalizada en FLOSS es la disputa por los mejores desarrolladores, cuyos exponentes pueden ser identificados, evaluados y captados a través del sistema meritocrático de las comunidades OS y allí tienden a concentrarse. Son ellos los que utilizan los entornos de desarrollo, y es claro cómo las empresas líderes van tendiendo a disponer entornos “*developers friendly*” que motivan a trabajar sobre sus proyectos.

(iv) Consolidación de un modelo de producción híbrido en el ámbito FLOSS:

Como contracara de los procesos anteriores, el modo de desarrollo de los proyectos al interior de las comunidades FLOSS es ya predominantemente híbrido, habiendo virado de ser predominantemente individual y voluntario a ser sponsorado, sostenido o manejado por corporaciones del sector. Esto es, manteniéndose como un segmento de producción no monetizada pero intrínsecamente constitutivo del mercado global del software.

B) Cambios en los procesos de competencia entre las empresas ligados a la extensiva importancia del FLOSS.

En términos de organización industrial, eje de este capítulo, puede decirse que el conjunto de cambios identificados respecto a los nichos de actividad y vínculos

con el FLOSS está intrínsecamente vinculado a los procesos de competencia de las empresas productoras de software que lideran el mercado.

En cuanto a nichos de mercado, Microsoft y Google compiten entre sí por desplazar a Amazon del liderazgo occidental en *cloud computing* empresarial. Hasta el momento Microsoft viene ganando la pulseada bilateral a partir de su novedosa y agresiva estrategia de ingreso al FLOSS. IBM/Red Hat intenta competir con Microsoft y Google, desde un segundo lote. Las distribuciones Linux de Microsoft y Red Hat compiten con la de Canonical, que a su vez ofrece su propia nube en torno a Ubuntu. Oracle compete con IBM y SAP en nube y las tres cooperan entre sí en iniciativas sobre bases de datos. Tencent no compete hasta el momento en el mercado occidental en términos de productos, pero sí en términos de la propiedad de los productos más rentables en un segmento como el de videojuegos en auge mundial y que involucra intrínsecamente la producción de software; sin contar que domina el mercado chino del software y los desarrollos digitales en su amplísima dimensión.

Repasemos ahora las relaciones de cooperación. Las empresas estudiadas cooperan en torno al desarrollo de Linux (tanto al kernel como a las diversas distribuciones más populares como Debian o Ubuntu), el desarrollo de la nube abierta OpenStack, el repositorio estrella de proyectos abiertos GitHub, el proyecto de contenedores Kubernetes de Google, las máquinas virtuales de Microsoft que soportan los sistemas de IBM, Oracle, etc., la Open Invention Network en torno a patentes, y una larga lista de proyectos compartidos que ya hemos detallado. Es decir, las empresas competidoras aportan cuantiosos recursos financieros y humanos al desarrollo colaborativo de diversos proyectos abiertos vinculados a casi todas las dimensiones técnicas y legales del software actual para desarrollo y almacenamiento, cuyos beneficios comerciales puede eventualmente apropiarse cualquier otro actor de la trama. Incluso los proyectos OS de nube liderados por estas empresas compiten con la gran nube abierta OpenStack en la que las mismas empresas también contribuyen.

A partir de este cuadro de relaciones se observa con claridad cómo las empresas compiten pero cooperan para ganar, a través de la lógica de la *“coopetition”*. Hasta aquí ello podría no resultar particularmente novedoso en tanto -tal como señalan los autores citados- dicha lógica suele darse frecuentemente en distintas industrias en la actualidad. En la *“coopetition”* típica el *“cluster”* es el ganador, a costa de desplazar o no permitir la entrada de otros participantes del mercado.

Desde nuestro punto de vista, lo llamativo se encuentra en dos elementos específicos de esta lógica, principalmente:

- Por un lado, el hecho de que en los desarrollos FLOSS la apropiación de los beneficios por parte de competidores puede ocurrir en forma inmediata, es decir que a priori ni siquiera existen las ventanas temporales en las cuales puede desplegarse la lógica de conformación y reconfiguración de *clusters*.

- Por otro lado, es de alta relevancia el hecho de que las consecuencias técnicas y económicas del tipo de cooperación que las compañías establecen a través de proyectos abiertos no son cognoscibles, mensurables ni controlables *ex ante* del mismo modo que las apuestas comerciales tradicionales (privativas), en tanto una de las características distintivas de los proyectos de comunidades abiertas es que “toman vida propia” (ver Capítulo 3). Al mismo tiempo, el mercado en el que operan estas compañías y proyectos es particularmente dinámico hoy, con lo cual el éxito de las estrategias de colaboración a partir de estos ingentes esfuerzos es incierto.

Entonces, a diferencia de la “*coopetition*” tradicional, en la “*open-coopetition*” que estos actores despliegan intensivamente, visualizan un espacio de rentabilidad diferencial que los motiva a embarcarse y crecer en el ámbito FLOSS. Dicho diferencial, dadas las características de las tecnologías implicadas, solo se hace posible hoy si se garantiza la compatibilidad e interoperabilidad de los diversos sistemas involucrados (incluso con el entorno privativo de Amazon). Y dichas características son técnicamente posibles si se utilizan infraestructuras comunes y de gran escala, cuyos exponentes más avanzados hoy son de tipo OS. De manera que, para las empresas que necesitan que sus productos se comuniquen entre sí, cooperar es una estrategia de competencia.

Estas dinámicas que conjugan lo común y lo individual, a su vez, dan cuenta de la predominancia de la lógica de ecosistema que prevalece en el mercado global del software, y en el campo del FLOSS en particular. Una vez lanzadas las estrategias de *open coopetition*, las interdependencias de red características de todo ecosistema se van acentuando, y así la curva de adopción de FLOSS adquiere el carácter de lo que podemos denominar “efecto ecosistémico”.

Por último, en la misma dirección cabe resaltar que el potencial de rentabilidad de estos desarrollos abiertos, si bien está parcialmente determinado por las tecnologías ya existentes, se observa como particularmente traccionado por la oferta en tanto las empresas líderes se ocupan especialmente de trabajar en promover la adopción de *cloud* por parte de las empresas desarrolladoras de software (clientes

o potenciales clientes), como ponen de manifiesto principalmente los hallazgos sobre Google, IBM y Tencent. En este sentido, junto al avance de las tecnologías, la estructuración oligopólica de la industria que permite que los líderes vayan logrando generar su propio mercado para las tecnologías que van desarrollando a partir de FLOSS, constituye en nuestra opinión un factor crucial en la explicación de las actuales dinámicas del mercado de software.

Como señalamos desde el comienzo, el caso de Amazon presenta un potencial desafío a la línea argumental que aquí sostenemos, puesto que es actualmente el líder en ese mercado oligopólico de *cloud* empresarial y está muy lejos del FLOSS. No obstante, en función de las indagaciones realizadas en este estudio consideramos que el pilar del liderazgo por ahora vigente de esta compañía en *cloud* se vincula a su carácter de “*early comer*”, y en virtud de ello al volumen de sus activos tecnológicos y de mercado en términos de desarrollo de contenedores, experiencia en el ámbito del almacenamiento, cartera de clientes a partir de su amplia gama de servicios, alcance geográfico, etc. Lo que no garantiza que el liderazgo continúe manteniéndose, visto el tipo de *catch up* desplegado principalmente por Microsoft con Azure Sphere en muy poco tiempo, y en segunda medida por Google en sus apuestas -aún de final abierto- con Anthos, o la propia envergadura de OpenStack.

Para finalizar, consideramos que en torno a la fuerte participación de las EMN en las comunidades FLOSS, es importante reflexionar más allá del objeto del presente artículo:

- ¿Cómo y por qué han llegado las herramientas FLOSS a ser técnicamente superiores en términos de interoperabilidad, flexibilidad y estabilidad que las de software privativo? ¿Incide en ello el hecho de tratarse de producción colaborativa?

- ¿Es posible plantear, o en qué medida, que la innovación en FLOSS es técnicamente más potente que en software privativo, o que el FLOSS potencia la innovación?

- ¿Es la *open competition* posible debido a que el mercado del *cloud computing* se encuentra en proceso de expansión, permitiendo que las firmas vayan ganando mercado sin necesariamente desplazar a sus competidoras?

- Relacionado con lo anterior y respecto a los riesgos tecno-políticos que puede entrañar la profundización de la injerencia de grandes empresas privadas en el ámbito FLOSS: ¿hay posibilidades de un giro hacia una apropiabilidad privada de los beneficios económicos de los proyectos OS?

Fuentes documentales

- Canal Ar (Agosto 2019a). IBM transformó su software para que pueda ejecutarse sobre cualquier nube con Red Hat. Disponible en <https://www.canal-ar.com.ar/28096-IBM-transformo-su-software-para-que-pueda-ejecutarse-sobre-cualquier-nube-con-Red-Hat.html>
- Canal Ar (Julio 2019b). IBM cerró la compra de Red Hat por 34.000 millones de dólares. Disponible en <https://www.canal-ar.com.ar/28048-IBM-cerro-la-compra-de-Red-Hat-por-34-000-millones-de-dolares.html>
- Canonical: <https://canonical.com>
- Computer World (Junio 2010). Oracle, SAP working together on Exadata support. Disponible en: <https://www.computerworld.com/article/2518769/oracle--sap-working-together-on-exadata-support.html>
- Computer World (Abril 2018). Aterriza en el mercado el primer servicio de la nueva base de datos autónoma de Oracle. Disponible en: <https://www.computerworld.es/tecnologia/ateriza-en-el-mercado-el-primer-servicio-de-la-nueva-base-de-datos-autonoma-de-oracle>
- Computing (Junio 2004). Sun y Fujitsu convergen sus sistemas Sparc. Disponible en :<https://www.computing.es/infraestructuras/informes/1010604001801/sun-fujitsu-convergen-sistemas-sparc.1.html>
- Electrónicos online Magazine (Febrero 2019). Estas son las empresas más innovadoras del 2018 según un estudio global. Disponible en <https://www.electronicosonline.com/estas-son-las-empresas-mas-innovadoras-del-2018-segun-un-estudio-global/>
- El País (Abril 2009). Oracle adquiere Sun Microsystems por 5.710 millones. Disponible en https://elpais.com/tecnologia/2009/04/20/actualidad/1240216080_850215.html
- El País (Mayo 2012). SAP compra Ariba por 3.400 millones. disponible en:https://elpais.com/tecnologia/2012/05/23/actualidad/1337764756_822942.html
- El País (Agosto 2012b). “Cuando te casas con un SAP, es para toda la vida”. Disponible en: https://elpais.com/sociedad/2012/08/27/actualidad/1346089934_596891.html. Forbes: <https://forbes.com>
- Forbes (Enero 2016). 10 Reasons Why Ubuntu Is Killing It In The Cloud. Disponible en <https://www.forbes.com/sites/janakirammsv/2016/01/12/10-reasons-why-ubuntu-is-killing-it-in-the-cloud/#2bd787a11ae5>
- Fundación Linux: <https://www.linuxfoundation.org/>

- Fundación Linux (2018). Tencent Supports Open Source Community With Linux Foundation Platinum Membership. Disponible en <https://www.linuxfoundation.org/press-release/2018/06/tencent-supports-open-source-community-with-linux-foundation-platinum-membership/>
- GenBeta (Julio 2019a). Ocurrió lo inevitable: el uso de Linux ha superado al de Windows en Azure, la nube de Microsoft. Disponible en <https://www.genbeta.com/desarrollo/ocurrio-inevitable-uso-linux-ha-superado-al-windows-azure-nube-microsoft>
- GenBeta (Abril 2019b). Los cuatro motivos con los que Google Cloud quiere arrebatarse el control de la nube a AWS y Azure. Disponible en <https://www.genbeta.com/desarrollo/cuatro-motivos-que-google-cloud-quiere-arrebatarse-control-nube-a-aws-azure>
- GenBeta (Junio 2017). Microsoft adora el open source, y está usándolo para crear Windows. Disponible en <https://www.genbeta.com/windows/microsoft-adora-el-open-source-y-esta-usandolo-para-crear-windows>
- GenBeta (Noviembre 2016). Microsoft se une a la Linux Foundation en calidad de miembro Platino. Disponible en <https://www.genbeta.com/actualidad/microsoft-se-une-a-la-linux-foundation-en-calidad-de-miembro-platino>
- Google. Proyectos Open Source. En <https://opensource.google.com/projects/explore/featured>
- IBM – Red Hat: <https://www.ibm.com/cloud/redhat?lnk=ushpv1811&lnk2=learn>
- Inteldig (Noviembre 2018). Las compañías que más contribuyen a open source. Disponible en: <https://www.inteldig.com/2018/11/las-companias-mas-contribuyen-open-source/>
- La Nación (Mayo 2010). Por qué Oracle compró Sun Microsystems. disponible en: <https://www.lanacion.com.ar/tecnologia/por-que-oracle-compro-sun-microsystems-nid1261880>
- Muy Linux (Septiembre 2017). Oracle dona Java EE a la Fundación Eclipse. Disponible en <https://www.muylinux.com/2017/09/18/oracle-java-ee-fundacion-eclipse/>
- Muy Linux (Abril 2016). ¿Cuánta gente usa Ubuntu? Mucha más de la que te imaginas. Disponible en <https://www.muylinux.com/2016/04/07/cuanta-gente-usa-ubuntu/>
- Muy computer (Octubre 2018). Microsoft promete proteger Linux y el Open Source con su unión a OIN. Disponible en: <https://www.muycomputer.com/2018/10/11/microsoft-se-une-a-oin/>

- Muy computer (Abril 2018). Microsoft presenta la primera distribución Linux de su historia. Disponible en <https://www.muycomputer.com/2018/04/17/distribucion-linux-microsoft/>
- Muy computer (Junio 2014). El gobierno chino publica las patentes con las que Microsoft domina Android. Disponible en <https://www.muycomputer.com/2014/06/16/microsoft-domina-android/>
- Muy computer (Octubre 2015). Microsoft recomienda Ubuntu como el mejor Linux para la nube. Disponible en <https://www.muycomputer.com/2015/10/06/microsoft-recomienda-ubuntu/>
- Openstack: <https://www.openstack.org/>
- Oracle: <https://www.oracle.com/>
- Oracle Community: <https://community.oracle.com>
- Python. Becas Google. Disponible en <https://www.python.org.ar/noticias/12/>
- Red Hat (Julio 2019): <https://www.redhat.com/es/about/press-releases/ibm-cierra-la-adquisici%C3%B3n-hist%C3%B3rica-de-red-hat-por-34000-millones-de-d%C3%B3lares-define-el-futuro-de-la-nube-h%C3%ADbrida-abierta>
- SAP: <https://www.sap.com/>
- Ubuntu: <https://ubuntu.com/>
- Xataka (Junio 2019a). La otra guerra entre Microsoft, Google y Amazon: la batalla por controlar los servicios en la nube para desarrolladores. Disponible en <https://www.xataka.com/servicios/otra-guerra-microsoft-google-amazon-batalla-controlar-servicios-nube-para-desarrolladores>
- Xataka (Enero 2019b). Así es como gana dinero Alphabet: el absoluto rey de la publicidad tiene un gran problema con el resto de productos y servicios. Disponible en <https://www.xataka.com/empresas-y-economia/asi-como-gana-dinero-alphabet-publicidad-poco>
- Xataka (Febrero 2019c). Tencent, así es el 'desconocido' gigante chino que domina la industria del videojuego y es dueño de 'Fortnite' y 'LOL'. Disponible en <https://www.xataka.com/videojuegos/tencent-asi-desconocido-gigante-chino-que-domina-industria-videojuego-dueno-fortnite-lol>
- Xataka (Octubre 2018a). Así es como gana dinero Microsoft: el reto de compensar el descalabro en móviles. Disponible en <https://www.xataka.com/empresas-y-economia/asi-como-gana-dinero-microsoft-cada-vez-nube-para-compensar-descalabro-moviles>.

- Xataka (Octubre 2018b). Ya es oficial: Microsoft compra GitHub por 7.500 millones de dólares. Disponible en <https://www.xataka.com/aplicaciones/oficial-microsoft-compra-github-7-500-millones-dolares>
- Xataka (Octubre 2017). Android y la apuesta por el open source: cómo y quién lo hace posible. Disponible en <https://www.xatakandroid.com/sistema-operativo/android-es-open-source-como-y-quien-lo-hace-posible>
- Xataka (Septiembre 2017b). Así es como Oracle ha sometido a Sun Microsystems a una muerte larga y agónica. Disponible en <https://www.xataka.com/historia-tecnologica/asi-es-como-oracle-ha-sometido-a-sun-microsystems-a-una-muerte-larga-y-agonica>
- Xataka móvil (Febrero 2016). La compra de Xamarin relanza a Microsoft hacia la multiplataforma. Disponible en <https://www.xatakamovil.com/aplicaciones/la-compra-de-xamarin-relanza-a-microsoft-hacia-la-multiplataforma>

Capítulo 4. Las PyMEs de Software y el FLOSS ¹

Denise Gutiérrez Montecino y Florencia Manzo

El uso de Software Libre avanza a un gran ritmo, no solo lo hace en empresas multinacionales, como IBM, Asus, Google, HP, entre otras; sino que también se ha insertado velozmente en el mercado de las PyMEs. Todos los códigos de software desarrollados en las comunidades FLOSS pueden ser gratuitamente descargados por cualquier MiPyme del sector de software y nada impide que con un software de este tipo monte su oferta de servicios, o lo ofrezca customizado a medida, lo utilice para nutrir sus procesos de aprendizaje, desarrollo e innovación en sus propios proyectos, etc. (Colombo et al., 2014). Esta situación ha potenciado la incorporación del FLOSS en el negocio de las PyMEs de software.

La expansión de la actividad FLOSS ha generado sustanciales cambios ampliando las posibilidades y estrategias de negocios para este tipo de empresas, potenciando su desarrollo y abriendo un nuevo abanico de posibilidades de comercialización. Además, ha estimulado muy diversas respuestas en materia organizacional y estrategias de negocios (Dahlander y Magnusson, 2005, Harison y Koski, 2010).

Así, el cuarto capítulo del presente libro pretende abordar la importancia del FLOSS e ilustrar el impacto que generan las comunidades FLOSS en las Pequeñas y Medianas Empresas (PyMEs) de Software, a partir de siete estudios de casos de la Argentina. Para ello se cuenta con un trabajo de campo realizado entre 2014 y 2018 por el Equipo del Grupo de Economía de Innovación y Desarrollo (FCE-UNC) con diversos objetivos de investigación.

1- Este capítulo resume aspectos salientes de una serie de estudios de caso realizados en el Equipo de Investigación entre los años 2014 y 2017 con diversos objetivos de investigación por Carina Borrastero, Hernán Morero, Jorge Motta y Pablo Ortiz. Las autoras de este capítulo han realizado una presentación sintética de aquél trabajo de campo, adaptado a la finalidad del libro.

El primero de estos estudios surgió de la necesidad de mejorar la forma de medir innovación en la actividad de software en general. De esta forma, durante 2014 un estudio de caso en profundidad (ver sección 3.E) que fue comparado con un conjunto de casos de referencia, se realizó con el objeto de identificar las dimensiones relevantes para dar cuenta del proceso de innovación en estas empresas (Morero *et al.*, 2014). El análisis cualitativo fue de elevada profundidad, con un trabajo de campo con 8 entrevistas semi-estructuradas alrededor de 5 ejes de análisis a nivel de la empresa para indagar sobre las características del proceso de innovación en este sector, tomando una empresa dinámica que opera en un segmento ‘*top knowledge*’ del mercado. El estudio llega a una estilización sintética del proceso de innovación en la empresa que fue volcado en diversos documentos académicos (Morero *et al.*, 2015, Borrastero y Morero, 2014). El caso llamó la atención sobre la falta de adecuación del diseño de las encuestas tecnológicas en el sector de software para la captación de ciertas especificidades del FLOSS y su representación en las estadísticas.

Ello maduró en el proyecto de investigación PICT 2015-2703 que culmina en este libro. Este proyecto propuso focalizar el estudio en el segmento FLOSS, y a través de estudios de caso y una evaluación de las encuestas anteriores en el sector de software, avanzar en diseñar un cuestionario que considere las particularidades productivas e innovativas de esta actividad. En este marco, entre fines de 2016 y 2017, se realizaron los otros 6 estudios de casos de empresas FLOSS que componen este capítulo. Estos estudios de caso se diseñaron con el objetivo de realizar el diseño de la “*III Encuesta sobre Innovación en el sector de Software de la Argentina 2019*” con la finalidad de aportar a dar visibilidad estadística a la participación de la producción de FLOSS en el sector de software y cuyos resultados parciales fueron presentados en el Capítulo 1.

En el proceso de diseño el equipo de investigación ha realizado publicaciones académicas realizando un análisis cualitativo con los casos con dos focos de estudio: i) Aportar una serie de reflexiones sobre las falencias de los Manuales de Innovación de la OCDE que guían el diseño de encuestas para captar las particularidades de las empresas FLOSS (Motta *et al.*, 2019, Morero *et al.*, 2017a); ii) Evaluar taxonomías de empresas de software según su uso y producción de FLOSS (Morero *et al.*, 2017b, Morero *et al.*, 2017c). Este capítulo se centra en explotar ese trabajo de campo del Equipo, pero con el objetivo de ilustrar la manera en que las pymes de software montan ofertas productivas alrededor del FLOSS, complementada con información de la encuesta ejecutada en 2018 para los casos disponibles.

El capítulo se encuentra dividido, principalmente, en dos secciones con una breve reflexión al final del mismo. En la primera sección, se procede a realizar una revisión bibliográfica respecto a los antecedentes de la importancia del FLOSS para las PyMEs de Software. Se presenta un caso en particular de PyMEs para contextualizar lo mencionado antes, el caso de EzSistems, una compañía noruega que logró crear una comunidad de FLOSS y mantenerla, que aún hoy en día permanece activa desarrollando el mismo modelo OSS. En la segunda sección, se exponen casos de PyMEs FLOSS argentinas: GCoop, XTech, ECIC Systems, Kunan, Machinalis, TECSO y Entornos Educativos.

1. El uso y producción de FLOSS en Pymes de software

Aunque la mayoría de los estudios de casos que existen sobre la temática generalmente abarcan a las grandes empresas, ha surgido recientemente una corriente de investigación alternativa (aún con escasa información) que se ha enfocado en las PyMEs. Para este tipo de empresas la innovación en Software Libre juega un rol crucial ya que son empresas que por lo general carecen de un conjunto completo de recursos y competencias internas para desarrollar, producir y comercializar de manera efectiva sus innovaciones (Colombo *et al.*, 2014).

A pesar de la poca atención que reciben, este tipo de empresas esta cada día ocupando un lugar más importante en la industria tanto a nivel internacional como en Argentina. En la unión europea, las PyMEs con menos de 250 empleados constituyen casi el 70% del sector de la informática y actividades relacionadas.

Se conocen, además, casos de PyMES exitosos, que han podido crear y mantener una comunidad FLOSS, como lo es el caso de eZSystems, un proveedor de software noruego fundado en 1999 (Hauge y Ziemer, 2009). A 2009 la empresa tenía alrededor de 60 empleados repartidos en oficinas en Noruega, Dinamarca, Alemania, Francia y Norteamérica. eZ se ha centrado casi desde su origen en proporcionar un PHPbasado en un sistema de gestión de contenido OSS (CMS por sus siglas en inglés), conocido como eZPublish, que es una aplicación que ofrece soluciones profesionales para el desarrollo de sitios web dinámicos, posible de ser implementado para proyectos de sitios web personales o sitios corporativos.² La empresa tiene una gran base de clientes en todo el mundo

2- El preprocesador de hipertexto PHP (PHP) es un lenguaje de script de servidor, utilizado para crear páginas web dinámicas. PHP es un OSS y es de descarga y uso gratuitos. PHP admite una amplia gama de bases de datos como MySQL, Oracle, Sybase, Solid, PostgreSQL, etc.

y su CMS se ha descargado más de 2,5 millones de veces desde su sitio web, a partir de febrero de 2009.

La empresa, que nace en 1999, comenzó desarrollando aplicaciones para corredores de bolsa y servicios de consultoría a negocios locales, lo que incluía administración de sistemas y desarrollo de aplicaciones y webs. La expansión de Internet le aportó una gran demanda para la creación de sitios web, para lo cual fueron desarrollando herramientas propias para hacer y publicar sitios web dinámicos, que comenzaron a reutilizar progresivamente. Por ello las empaquetaron en dos softwares, uno denominado eZPublish y otro eZTrade; que fueron casi inmediatamente liberados a la comunidad, dado que todos los empleados de la empresa apoyaban la ideología OS. Durante un período de tiempo la firma continuó en ambas actividades, pero paralelamente eZPublish atrajo la atención de la comunidad y ésta comenzó a demandar servicios de consultoría relacionados. Seguidamente, en el 2001, tomaron la decisión de especializarse solamente en eZPublish. Entre 2001 y 2005, las versiones que siguieron permitieron el desarrollo de complementos para el software base y la creación de librerías y componentes. Antes de esas versiones el software tenía una amplia comunidad de usuarios, pero después de ellas, atrajo una comunidad de desarrolladores.

Esta configuración del software habilitó a la “creación” de tres comunidades alrededor de los dos productos (Publish y Trade): una comunidad de usuarios, una comunidad de desarrolladores de complementos (plug-ins) y una comunidad de desarrolladores de componentes (que se refiere a librerías que sirven como de insumos compartidos entre distintos complementos y funciones, y que pueden servir a otras aplicaciones fuera de este software en sí).

La comunidad de usuarios creó una gran necesidad de servicios en materia de soporte, calidad, capacitación, instalación y hosting, facilitando la venta de esos servicios y como canal de marketing. La comunidad de desarrolladores de plug-ins han ido extendiendo la funcionalidad del producto de la firma, incrementando su valor para la firma, pero también habilitando a los desarrolladores de estos complementos a proveer servicios en el nicho especializado de sus complementos. Esta comunidad es como una manera en la que la firma puede tercerizar el desarrollo de plug-ins y de su mantenimiento. Y por último, la comunidad de desarrolladores de componentes contribuye código a las librerías de eZ que habrían desarrollado de todas maneras, independientemente de las necesidades de eZ, que sirve a la comunidad OS en general.

Respecto a su impulso para crear las comunidades, eZ ha invertido en infraestructura específica para las tres comunidades. Para la comunidad de plug-ins, eZ provee un portar de alojamiento al tiempo que organiza “días del desarrollador” y un evento anual de Publish. La infraestructura para esta comunidad incluye foros, listas de mail, seguidores de problemas y cuestiones (*issuetracker*), documentación y código fuente. Para la comunidad de componentes, ha provisto listas de correo y un seguidor de problemas (*issuetracker*) abierto. Una cuestión que señala el paper es que la provisión de esta infraestructura es, en realidad, una inversión “pequeña”, incluso para una pyme. Algo a destacar es que no la montó al mismo momento de la liberación del producto, sino a lo largo del tiempo, por lo que la inversión se puede realizar de modo progresivo.

Respecto a la manera de atraer a la comunidad y “gobernarla”, en el artículo se señala la importancia de partir de un producto atractivo, capaz de atraer potencialmente a una base de usuarios amplia. Luego, la tarea de aprobar contribuciones ha estado a cargo de la empresa, así como el hosting de las mismas, al tiempo que la empresa ha logrado transmitir una actitud positiva acerca del OS. En este caso de eZ, el atractivo para ser contratados por la empresa es muy menor y nunca propuesto por parte de ella. Por ello, la principal motivación de los miembros de la comunidad ha estado en la implementación de funcionalidades que están en sus propios intereses. Respecto al licenciamiento, la empresa ha usado una licencia dual, tipo GPL.

Por otro lado, siguiendo con la línea de casos de PyMEs que han logrado crear y desarrollar una comunidad FLOSS de manera exitosa indagamos en el trabajo presentado por Sowe y McNaughton (2012), donde llevan a cabo un análisis empírico de los datos de estudios de casos obtenidos de nueve Pymes basadas en TIC de Uganda, Etiopía, Sudáfrica, Kenia, Tanzania, Nigeria y Mozambique para mostrar cómo las PYME de estos países están aprovechando los beneficios inherentes a FLOSS para crear negocios sostenibles.

Los autores pudieron observar, en base a los estudios de caso realizados, que el FLOSS les ofrecía una opción alternativa de modelo de negocio para las empresas de TIC en África Subsahariana. Sin embargo, para estas empresas operar únicamente con soluciones y servicios FLOSS no es una opción comercial sostenible, pero sí lo era llevar a cabo sus actividades en forma de cuasi-negocio; lo que les permitía aprovechar las oportunidades de bajo costo y capacidad para personalizar las soluciones FLOSS, y al mismo tiempo vender y realizar el mantenimiento de software propietario. La principal motivación de las empresas

para participar en el negocio de software libre es impulsada por la reducción en el costo del desarrollo de software y el despliegue del mismo. Para la mayoría de las empresas, el software que necesitaban para iniciar un negocio (por ejemplo, crear sitios web, implementar y mantener un sistema de gestión de aprendizaje) estaba fácilmente disponible como FLOSS. Esto les ayudó a evitar problemas, como comprar y pagar un alto costo de licencias, asociado con el inicio de un negocio con software propietario.

Por ello es que el uso del FLOSS por parte de las PyMEs en la región ha aumentado significativamente y su desafío, según los autores, se ha vuelto ahora lograr mantener el nivel requerido de residentes y experiencia técnica para poder asentarse en una posición estable dentro de la comunidad FLOSS desarrollada.

Estos desafíos también sugieren oportunidades para intervenciones de políticas a nivel estatal o nacional que respaldarán y fomentarán la legitimidad y el uso de software libre, como un medio para estimular la industria de software indígena y crear empleos locales y oportunidades empresariales.

Además del análisis puntual de los casos en la región de África Subsahariana, en el mencionado artículo se mencionan una serie de datos que creo relevante destacar como que 85% de las empresas en Asia, el Pacífico, Europa y América del Norte están utilizando FLOSS, al menos el 80% de todas las soluciones de software comerciales incluyen componentes sustantivos de software libre y muchos municipios regionales en Europa están usando actualmente FLOSS (Gartner, 2008).

Los datos reflejan una situación cada vez más prometedora en lo que refiere al uso del Software Libre en el mundo para el caso de pequeñas y medianas empresas, aunque aún queda mucho por recorrer. Dichos datos son un poco más desalentadores para la región latinoamericana, aunque en Argentina el crecimiento del FLOSS no ha sido menor. A pesar de ello aún es muy escasa la información cuantitativa disponible lo que dificulta en gran manera el desarrollo de trabajos de investigadores especializados en el tema, por lo que el Capítulo 1 de este libro realiza un aporte en esta dirección.

Actualmente se ha evidenciado una tendencia reciente es la formación de emprendimientos con modelos de negocios basados en FLOSS en Argentina. 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, 2006b) 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). Por ello, cobra relevancia los resultados descriptivos presentados en el Capítulo 1 para la Argentina, que representan el diseño de un relevamiento específico a nivel de empresa en el sector del software en Argentina, a través de una investigación intensa en recopilación y evaluación de cuestionarios anteriores, y en estudios cualitativos, dada la escasez de datos nacionales en materia de FLOSS.

2. Estudios de casos de PyMEs FLOSS argentinas

En esta sección se presentan casos de PyMEs argentinas de Software que utilizan y/o producen Software Libre y de Código Abierto o *Free/Libre Open Source* Software (FLOSS). Los siete casos que se analizan son: GCoop, XTech, ECIC Systems, Kunan, Machinalis, TECSO y Entornos Educativos.

La información presentada en esta sección proviene de la combinación de diversas fuentes recopiladas por los miembros del equipo de investigación: entrevistas en profundidad con informantes clave, pertenecientes a distintas áreas organizacionales de las empresas presentadas; en segundo lugar, documentación de las empresas (sitios web oficiales de las firmas, blogs y foros de la firma, publicaciones en conferencias sobre los proyectos de la organización); y por último, la observación casual y directa dentro de los distintos establecimientos, lo que permite captar situaciones que no se manifiestan en los documentos ni surgen espontáneamente por parte de los entrevistados. Para el caso particular de Machinalis, se utilizó el Documento de Proyecto de Morero *et al.* (2014) en que se realizó un estudio de caso para dicha empresa en particular.

Tales entrevistas fueron semi estructuradas alrededor de un conjunto de ejes para abordar ciertas temáticas a nivel de empresa: Caracterización general de la empresa y situación actual; Actividades creativas FLOSS, vinculaciones y dinámica innovativa; y el Rol de colaboración con la Comunidad FLOSS y cuáles son las formas de participación en ella.

En la tabla siguiente se presenta el año en que se realizó la entrevista a la empresa junto con la ubicación que declaró.

Tabla 4.1. Caracterización PyMEs argentinas

<i>Nombre</i>	<i>Ubicación</i>	<i>Año</i>
GCoop	Bs. As.	2017
XTech	Bs. As.	2017
ECIC Systems	Córdoba	2016
Kunan	Córdoba	2016
Machinalis	Córdoba	2014
TECSO	Santa Fe y Bs. As.	2017
Entornos Educativos	Bs. As.	2017

Fuente: Elaboración Propia en base a entrevistas

El objetivo de esta sección, además de presentar las características principales de las empresas y sus actividades productivas, es relacionarlas con la Comunidades FLOSS, de qué manera interactúan con esta, el aporte que realizan y como se benefician de su participación.

A. GCoop

El primer caso que se presenta es la empresa de desarrollo de software libre llamada GCoop. Se encuentra ubicada en la Ciudad Autónoma de Buenos Aires (CABA), fundada en el año 2007. Posee aproximadamente 12 años de experiencia en el mercado, organizada como Cooperativa de Trabajo. La empresa se enfoca en otorgar soluciones según las necesidades de sus clientes a través de frameworks de desarrollo libre que tenga una comunidad grande.

Los integrantes de Gcoop poseen distintos niveles de participación en organizaciones sociales y de la economía social, tales como el cooperativismo, radios comunitarias, entre otras. Se organiza por equipos en que se distingue el equipo de desarrollos a medida, un equipo de desarrollo de portales web en Drupal, un equipo de implementaciones de ERP Tryton y un grupo de servicios tipo SaS que, a través de la plataforma “nube.coop”, otorga soluciones tecnológicas cooperativas y de comercio electrónico. Sus integrantes participan de comunidades de usuarios Linux como CAFELUG (Grupo de Usuarios de Software Libre de la Ciudad de Buenos Aires), agrupaciones de promoción del FLOSS y en Comunidades de FLOSS “técnicas”.

En particular, la empresa se especializa en la implementación de ERP (Enterprise Resource Planning o Sistemas de Planificación de Recursos Empresariales), es

decir, proporciona sistemas de información gerenciales con distintos módulos que integran y manejan negocios asociadas a operaciones desde la producción hasta la distribución de bienes y servicios. Para ello, la empresa utiliza una plataforma informativa general de alto nivel sobre la cual se desarrollan soluciones de negocios a través de sus módulos denominada Tryton (sistema de ERP de Python).

También desarrolla software a medidas en Django (Python), SuiteCRM y Drupal (sistema de gestión de contenidos o *Content Management System*) y desarrollos web con Drupal. Drupal (CMS) es un programa que permite crear una estructura de soporte para la producción y administración de contenido principalmente en páginas web. GCoop es miembro de la DrupalAssociation y la cooperativa está “certificada” como proveedor de servicios Drupal y como capacitador Drupal.

La empresa hace hincapié en que sus potenciales asociados tengan valores cooperativos adquiridos antes de otorgar algún tipo de formación técnica necesaria dentro de la cooperativa. Es decir, apuesta al trabajo asociado con otras cooperativas en el marco de una federación que las contiene (inter-cooperación). No sopesan al crecimiento de la empresa con el aumento del tamaño de la misma, ya que les traería más inconvenientes (mayor burocracia interna) que beneficios. Por ende, la firma tomó la decisión de dejar de crecer y apostar a la Federación Argentina de Cooperativas de Trabajo de Tecnología, Innovación y Conocimiento (FACTTIC) y así, poder crear un grupo inter-cooperativo de cooperativas de tecnología.

Entre sus productos desarrollados se distingue una implementación de SugarCRM para el Banco Credicop. También ha realizado implementación de TrytonERP para la gestión y venta en línea de la Editorial de la Universidad de Lanús, para el manejo de ventas y producción en empresas lácteas, mueblerías, para el Estado y la gestión de imprentas. Por último, ha realizado diversos desarrollos de portales web, federaciones y confederaciones de cooperativas, empresas de seguros, portales de periodismo deportivo, empresas cosméticas, entre muchos otros.

La liberación de código, software completo y módulos para proyectos FLOSS de la comunidad es una actividad frecuente de la empresa, aunque no posee ningún acuerdo formal respecto a la liberación con sus clientes. Uno de los problemas de la empresa, es que existen algunos desarrollos muy a medida perdiendo utilidad la liberación, pudiendo no ser útil para la comunidad. En cambio, sí realizan liberaciones cuando se trata de algo más universal, se planifica y asigna tiempo

como para liberarlo. De este modo, la actividad de liberación de software es parte del tiempo de trabajo.

A grandes rasgos, la empresa colabora en comunidades FLOSS y otras asociaciones sociales y cooperativas, con diversos módulos desarrollados para la comunidad, tanto individualmente como colectivamente con otras empresas FLOSS. Ha participado en proyectos I+D conjuntamente con otras empresas FLOSS y desarrollado proyectos internos y actividades destinadas a la experimentación con nuevas ideas y tecnologías, además de haber escritos algunos artículos, presentado a conferencias y charlas en congresos y eventos. A continuación, se detallan algunos de las actividades principales mencionadas.

En primer lugar, nos referiremos a las actividades realizadas por la empresa y sus formas de colaboración en las comunidades en las que participa. Gcoop participa de manera activa y “motorizada” en comunidades de usuarios Linux (CAFELUG), agrupaciones de promoción del FLOSS (Vía Libre, Solar, otras) y en Comunidades FLOSS “técnicas”. Su participación en comunidades FLOSS se distinguen entre aquellas que son globales y las de carácter nacional. Dentro de las globales se distingue a Drupal, SuiteCMR y Tryton mientras que las nacionales son Tryton-ar, Python -ar y Ubuuntu-ar. Las formas de colaboración van desde actividades que implique asistir a eventos y seminario hasta la realización de parches y liberaciones.

Según la información que se posee, la empresa figura como co-organizador de cuatro eventos regionales de Drupal, en que ha realizado un gran aporte al núcleo del sistema (Drupal 8), junto con otras colaboraciones. En particular, han aportado la liberación de la localización argentina, han aportado 3 módulos de logística para Drupal Commerce, y han colaborado en otros 8 proyectos. También desarrollaron dos módulos en Drupal de TodoPago para un software de un cliente, en que posteriormente, TodoPago lo terminó incorporando al sistema, generando que en 2017 sea partner con este.

En segundo lugar, se encuentran sus aportes a las comunidades Python y Tryton. Se destaca la integración de librerías en Python, además de la creación de la localización de Tryton Argentina y el desarrollo de 5 módulos de ERP para cuestiones de uso argentino. También participa de la comunidad de SuiteCRM, en que realizó la primera liberación, llamada SugarGraphviz para facilitar la programación de módulos, luego realizó la liberación de un módulo llamado Workflow.

En GCoop hay proyectos y actividades internas con el objeto de experimentar en tecnologías e ideas, muchos de los cuales han derivado en innovaciones sociales y dedicado tiempo laboral a proyectos de impacto social. En estos proyectos hay un interés propio en las metas sociales por parte de los miembros de la cooperativa, sin un fin comercial, por más que puedan implicar algún tipo de contraprestación económica.

En conclusión, GCoop participa principalmente en proyectos de Comunidad FLOSS en que coexiste con una multiplicidad de empresas. Su participación responde principalmente a un Modelo de Aporte de Código según la clasificación propuesta por Capra *et al.* (2011), realizando una personalización de software adaptándola a las necesidades y objetivos de los usuarios argentinos, o siguiendo la clasificación de Dahlander y Magnusson (2005), se la clasifica a la empresa dentro de un enfoque simbiótico, cuyos aportes le han permitido a la firma tener mayor control de los desarrollos de la comunidad, obteniendo un beneficio productivo y económico.

B. XTech

La empresa XTech se encuentra en la provincia de Buenos Aires, fundada en el año 1999, con 23 empleados aproximadamente. La empresa se enfoca en proyectos de servicios de infraestructura con una enorme diversidad de clientes y en resolver problemas muy diversos de esta índole. Además, brinda la tercerización de personal con conocimientos en software libre en infraestructura y en cloud (AWS – Amazon Web Services, Docker, OpenStack).

Gran parte de sus ventas proviene de productos y servicios vinculados al FLOSS cuyo destino son en su totalidad de carácter nacional. La empresa no cuenta con una organización funcional por áreas, es decir, los equipos se arman según el proyecto que surge. No realizan actividades de I+D ni mucho menos poseen un departamento destinado a este tipo de actividades.

XTech busca empresas que ya posean sistemas de Linux, tales como Garbarino, para la que prestó servicios, en que con contextos de alta demanda de servicios de impresión de todas las sucursales y frente a la aparición de una falla, el desperfecto no se repita y siga funcionando el resto de la infraestructura de impresión. También realizó una configuración de manejo de redes para la administración pública nacional con relación a la regional y local, entre otras.

Son más esporádicas las actividades de desarrollo de software, ya que no han tenido demanda de clientes en el área que se especializan. Han realizado

desarrollos relacionados con el manejo inteligente del aumento en la transmisión de información en redes, y han desarrollado algunos software a medida sobre manejo de padrones electorales y gestión de afilados, siempre utilizando *frameworks* y técnicas de software libre (Python y Django). Además, poseen una línea de provisión privativa, llamada Zextras, vendida con licencia que sirve para mejorar un software libre (Zimbra).

La liberación en XTech se hace en menor relación con la comunidad. Consiste en la publicación de repositorios de código completo y otorgando una licencia GPL. En general, no participan activamente colaborando con alguna comunidad, debido a que sus proyectos no lo requieren. No hay ninguna preocupación por parte de los clientes en estos casos sobre la liberación, por lo que no hay ningún tipo de acuerdo o contrato, liberándose aproximadamente el 50% de los desarrollos que han hecho.

Comparado con GCoop, XTech posee una participación menos activa en las comunidades FLOSS, en que la primera ponía un gran énfasis en sus actividades internas y proyectos en que participaba orientadas al desarrollo de SL. Siguiendo a Dahlander y Magnusson (2005), se puede considerar a XTech dentro del enfoque comensalista, ya que deben desarrollar una capacidad para adaptar sus estrategias para proporcionar lo que los clientes desean.

C. ECIC Systems

En la provincia de Córdoba se encuentran tres casos. El primero se refiere a ECICSystems, una empresa pequeña de software. La empresa otorgaba soluciones GPL orientado a la provisión de servicios a administradores de redes, sobre la base de software libre. Para ello, generaron desarrollos propios de software de servers y ofrecían servicios sobre la base de ese producto. Unos de sus objetivos era la optimización de recursos de terceros, para generar una interfaz entre los distintos equipos de red más eficiente.

La empresa se enfocaba exclusivamente en el mercado interno, cuyos principales clientes eran empresas de servicios (gastronomía, hotelería y urgencias mecánicas del automotor) y sector industrial. Aunque poseían una amplia cartera de clientes, sus ventas se concentraban en pocos clientes. Asimismo, la mayoría de sus ventas se basaban en servicios de software libre y, en menor nivel, al desarrollo de software a medida y servicios de productos asociados.

La empresa dedicaba tiempo a actividades de I+D internas, en que adquirirían nuevos software para desarrollar nuevos productos y servicios además de los de

uso interno, como también software para el análisis según los requerimientos de los clientes. En general, son innovaciones para la empresa, aunque en nuevos productos y servicios con mejoras significativas a nivel de mercado interno.

Desarrollaron lo que hoy se conoce como PITS Router, nombre comercial de un software para la producción de una plataforma de control centralizado de tráfico y seguridad en redes. Su participación entre 2012 y 2013 del Primer Cluster de Software Libre en el país, creado ad hoc con sede en Paraná-Entre Ríos y financiado por el FONTAR, junto a empresas de Entre Ríos, Córdoba y la Universidad Nacional de Entre Ríos, además de permitirle aportar a la promoción y visibilización de la producción OS como un tipo de producción profesionalizada, seria y sistemática, le facilitó completar el desarrollo del mencionado software.

A pesar de estos avances, el cluster nunca funcionó como tal y el proyecto en sí mismo tampoco incentivó la interacción entre los participantes del cluster. Sumado a que había muy poca gente que hiciera *networking* en esos temas en la comunidad. El software en cuestión, que tiene licencia GPL, es libre pero no está liberado.

La vinculación con la comunidad *Open Source* se daba básicamente por dos vías. Una de ellas indirecta, a través de las actividades de uno de los socios de la empresa que es Embajador de Fedora. Si bien no forma parte de la política de la empresa por falta de tiempo para hacerlo, dicho socio solía dedicar algún tiempo a participar en acciones IRC con la comunidad. La segunda vía de vinculación con la comunidad FLOSS es más directa, y proviene de la colaboración que reciben de dicha comunidad para la confección de documentación de modo gratuito. Solicitan a la comunidad que se las haga y hay gente que les colabora. Si bien, el foco de su negocio giraba alrededor del software libre, no se encuentra asociado a CADESOL.

Para poder actuar en su mercado objetivo de manera competitiva han debido desarrollar una plataforma de control centralizado de tráfico y seguridad en redes, que le permite brindar a sus clientes un servicio confiable y eficiente. Su negocio se basa en las capacidades de uso y reconfiguración según las necesidades del cliente de dicho software que la empresa ha logrado durante el periodo de desarrollo del mismo. Por lo tanto, se diferencia y destaca de sus competidores en la especialización de un segmento específico del mercado y el alto desarrollo de calificaciones y de experiencia que dicha especialización ha permitido e incentivado. El financiamiento de estas actividades de innovación provino fundamentalmente de recursos internos y también del Fontar.

D. Kunan

La empresa de Kunan, también ubicada en la Provincia de Córdoba, se especializa en tres áreas productivas, una de ellas basada en FLOSS. En 2015 contaba con una planta de 14 ocupados (la mayor parte de ellos desarrolladores), que ascendía a 20 personas en 2014 y que ha retomado este nivel hacia 2016.

El producto FLOSS en el que se especializa la empresa es un CRM (*Customer Relationship Management*), en particular CRM Suite, que es un derivado de Sugar CRM. Sugar CRM es una aplicación para la administración de la relación con los clientes (un CRM) basado en LAMP (Linux-Apache-MySQL-PHP), diseñada para facilitar la gestión de ventas, oportunidades y contactos de negocios. Un CRM es básicamente un software para la gestión relacional de una organización. Suite CRM es una bifurcación (un *fork*) de Sugar CRM, que se popularizó cuando Sugar decidió discontinuar el desarrollo de sus versiones open source comunitaria.

El área CMR reporta alrededor del 30% de los ingresos de la empresa y está compuesta por tres trabajadores a tiempo completo. Luego hay dos áreas más: DBA remoto y soluciones mobile. El área de DBA remoto provee servicios de mantenimiento, monitoreo y administración de bases de datos Oracle. Cuenta en la actualidad con 7 empleados y da cuenta de alrededor del 60% de la actividad de la empresa. El área *mobile* es más incipiente, ofrece soluciones Android y está compuesta por dos trabajadores. La empresa cuenta con actividades de I+D de manera transversal a la empresa, es decir, que en cierta forma todos los trabajadores participan y no hay un departamento definido de tal actividad.

La empresa contaba con un nivel de ventas de alrededor de 400.000 dólares al año, prácticamente concentradas en el mercado interno (en 2015 las exportaciones representaron sólo el 4% de las ventas); y presenta un nivel de innovación alto. Ha introducido varios tipos de innovaciones durante el período 2013-2015: nuevos productos, productos con mejoras significativas, nuevas soluciones integrales y servicios con mejoras significativas. Todo ello revistiendo novedad para el mercado doméstico, mientras que además ha introducido mejoras organizacionales nuevas para la empresa.

También como parte de las actividades del área OS, ha estado la aplicación y adaptación de Sugar para uso interno en la empresa para todos los procesos de gestión interna y de desarrollo. Esto implicó la implementación de CRM a Kunan, en una primera instancia en la fase comercial; y en un segundo momento, utilizando módulos existentes adaptados, en el resto de las operaciones dentro

de la empresa y para el sistema de calidad para las auditorías y evaluación y certificación de las normas ISO, incorporando otros procesos, como los administrativos desarrollándose módulos específicos. En Kunan liberan sólo el 20% de sus desarrollos en OS, más por una limitación de tiempo, que por una política particular de negocios.

La empresa participó como coordinador corporativo de Libertya, un ERP Open Source. Esta no contaba con una plataforma de desarrollo colaborativo que permitiese probar los aportes en versiones paralelas de prueba, donde se vea en cuando impacta una modificación, si corrige efectivamente las fallas o no. En ese sentido, esta falta de una herramienta de interacción colaborativa para el desarrollo, en cierto modo ralentizó el ritmo de evolución del producto respecto de otro tipo de proyectos FLOSS internacionales.

Otra de las actividades de promoción del FLOSS que han realizado, y podemos mencionar como una actividad de tipo social, consistió en una capacitación que realizaron en una escuela secundaria en un contexto social de relativa marginalidad social. A través de un sistema de pasantías con el IPEM 338 en barrio Marqués Anexo de Córdoba, tomaron un alumno en la empresa que capacitaron primeramente en actividades de la empresa, y luego se trabajó en una implementación del CRM de Sugar las necesidades del sistema de la escuela.

Se diseñó un sistema de gestión administrativa para los alumnos, los profesores y las materias, haciendo las modificaciones necesarias a los módulos de Sugar, y el pasante fue el nexo entre la empresa y la escuela. Trasladó la capacitación a los alumnos del colegio, hizo las instalaciones del CRM en las *netbooks* provistas por el Estado, y los alumnos quedaron a cargo del mantenimiento del sistema. Más allá del interés social del proyecto, que involucraba además la propuesta de réplica de la experiencia con otras empresas del Cluster Cordoba Technology que no tuvo una buena recepción, la intención estaba en reforzar la difusión de las ventajas del software libre en lo estudiantes secundarios.

Otras de las actividades educativas en FLOSS que realiza Kunan están en el ámbito universitario con la finalidad de trasladar a este ámbito, las tecnologías que se utilizan de este tipo a nivel empresarial, que no persiguen un beneficio directo tangible para la firma, sino como acciones de promoción y sostenimiento de la actividad del FLOSS.

El abanico de actividades de vinculación se completa con los *partnerships* con otras empresas y la participación en asociaciones empresariales. Con respecto a esto último se destaca la participación de Kunan en la Cámara Argentina de

Empresas de Software Libre (CADESOL), donde es miembro activo. También es miembro de las asociaciones empresariales locales, como el Cluster Technology Cordoba, que agrupa a la mayoría de las PyMEs de software de la región.

Kunan es un caso mayoritario entre las PyMEs FLOSS, es un modelo de negocios híbrido, combinando áreas de negocio que funcionan con la lógica típicamente privativa con áreas de negocio que han logrado montarse a través de una estrategia FLOSS. El negocio se apoyó en elección estratégica de un proyecto de software FLOSS (CRM) sobre el cual montar una oferta de servicios determinado, que además tenga potencial competitivo.

Kunan, igual que la empresa GCoop, posee un interés propio en las metas sociales. Ambos se han enfocado en las innovaciones sociales así como también han dedicado tiempo a proyectos de impacto social.

E. Machinalis

En la Ciudad de Córdoba también se encuentra Machinalis, una empresa que surge en el año 2009 como S.R.L, a partir del cierre de una empresa anterior en el 2008 (Except). Posee aproximadamente alrededor de 60 empleados con un alto nivel de calificación formal. La empresa se dedicaba al desarrollo de software libre mayormente Python (aunque no se especializaba en eso) y servicios en las áreas de Inteligencia Artificial, Data Processing, Big Data o Data Science y Servicios de Desarrollo Complejo. Se distinguían tres áreas: área operativa de la empresa, área comercial y de RRHH y área de calidad y administración de finanzas.

Recientemente (2018), Mercado Libre (MeLi) anunció la adquisición de la empresa para el desarrollo de Inteligencia Artificial (IA) de Mercado Libre. Esta área es de suma importancia en una empresa de *e-commerce* como MeLi, para el desarrollo estratégico que le permita ser diferenciador e innovador.

Respecto a sus principales clientes, en el año 2014 la empresa exportaba la totalidad de su producción, siendo su destino principal los EEUU. Para concretar las ventas externas, en términos generales en la empresa se utilizaban las siguientes modalidades: se buscan clientes o *partners*, y las vinculaciones comerciales que surgen en ocasiones están amparadas en contratos académicos o tecnológicos que generan oportunidades de negocios.

Siguiendo el documento de trabajo realizado a Machinalis (Morero, Borrastero y Ortiz, 2014), se detalla mejor las actividades realizadas por la empresa. La firma se dedicaba principalmente al desarrollo de software a medida, empleando y desarrollando herramientas al interior de las áreas de conocimiento mencionadas.

Una de sus particularidades es que en la empresa se realizaban dos tipos de proyectos tecnológicos:

- Proyectos basados en software libre para clientes (*clientled projects*): se trata de desarrollos a medida típicamente comercial, orientada a la satisfacción de las necesidades de un cliente, producidos principalmente a partir de herramientas de desarrollo de software libre (SL).

- Proyectos “internos” o “netamente innovadores” (*internally led projects*): estos son apuestas internas innovadoras que tienen por objetivos generar procesos de aprendizaje en la organización, desarrollar herramientas internas, mantener la satisfacción de los integrantes de la empresa y posicionarla empresa en el mercado como proveedora de productos/servicios innovadores en sus áreas de experiencia. Se trata de proyectos no orientados a clientes ni a objetivos estrictamente comerciales, desarrollo de software netamente SL que es liberado y compartido en la comunidad tecnológica, no vendido.

A finales de 2013, el 65% de las actividades de la empresa constituían desarrollo de software a medida y el restante 35% actividades de Factory. En 2014 esa distribución cambió, pasando el desarrollo de software a medida a representar el 100% de las ventas de la empresa. Además, al momento de realizar la entrevista, la empresa no contaba con un departamento destinado a la investigación y desarrollo, ya que sostenían que para empresas del tamaño de Machinalis no era necesario, sino que aspiraban a tener una cultura de I+D e innovación en toda la firma (“que atravesen a toda la firma”).

Si bien la empresa producía software a medida como base de su actividad productiva, también se dedicaba al desarrollo de partes de software o aplicaciones (módulos, librerías, componentes), que forman parte de los proyectos innovadores, de los cuales se generaban piezas de software que son liberadas. También realizaban servicios “no monetizados”, es decir, servicios que cumplen con algunas características de los productos (usabilidad y escalabilidad) pero constituyen servicios especiales por ser desarrollos a medida. Los servicios se encontraban asociados a los productos que vendían o compartían con la comunidad tecnológica, y por las tecnologías de desarrollo que utilizaban.

Tanto estas últimas como buena parte del software específico que producían son de tipo software libre y en los casos de venta de software, la generación y utilización de herramientas libres para desarrollarlo eran concebidas por la empresa como “servicios de desarrollo” asociados al *know how* que la empresa poseía para producir a partir de ellas, más allá de los productos específicos que se

generaban: principalmente el lenguaje de programación Python y el *framework* de desarrollo web Django.

En cuanto a la incidencia específica que tiene el software libre en la actividad productiva de la empresa, su uso es una de las principales singularidades de la firma. La empresa posee interacciones con la comunidad con objetivos directamente vinculados al proceso de desarrollo e innovación de los proyectos, tales como el testeo y validación de *releases*, así como un reclutamiento de personal adecuado para cada proyecto o equipo, entre otros.

Más allá de la visión filosófica de la empresa respecto al SL (misión y visión de la empresa basada en esta), lo defienden por ofrecer potenciales herramientas para lograr desarrollos de calidad, *networking* a nivel mundial y posicionamiento de la firma en su nicho de oferta.

F. TECSO

La cooperativa de trabajo TECSO, fundada en el año 2002, se encuentra ubicada en la provincia de Santa Fe (Rosario) con dos centros de desarrollo y un centro en la provincia de Buenos Aires, con más de 150 empleados. La cooperativa brinda servicios de diseño, desarrollo e implementación de Software y Servicios Informáticos para grandes empresas y organismo públicos.

Más en detalle, TECSO se especializa en software FLOSS a medida orientados principalmente al sector público, software *factory* desarrollando módulos para otras empresas y, brinda servicios como consultoría (análisis funcional, gestión de proyectos, reingeniería de procesos de negocios) y servicios de soporte (resolución de incidentes de diversas complejidades, monitoreo, implementación de alertas y herramientas de control sobre los sistemas finales).

Sus clientes se pueden agrupar en cuatro áreas verticales: gobierno, seguros, industria y servicios (bancos, finanzas, salud). Algunos de sus clientes del ámbito privado son: Citibank, Toyota, Volkswagen, Banco Credicoop C.L., Banco Comafi, Banco General de Inversiones (Mex), Interbanking, Río Uruguay Seguros, Petrobras, IBM, Compañía Mega (Repsol), entre otros. En el ámbito público se destacan: Municipalidad de Rosario, Santa Fe, Paraná, Morón, Nogoyá, Villaguay, Entidad Binacional Yacyretá, SEMTUR Rosario, MIXTA Rosario, LIFIA-UNLP, INP-Gobierno de Chile y otros.

Los dos productos FLOSS más relevantes son el Sistema Integral de Administración Tributaria (SIAT) y la Gestión de Actas Electrónicas Móviles (GAEM). El primero consiste en un sistema web para gestionar actividades de

cálculo, emisión, recaudación y control de impuestos, que fue desarrollado en un primer momento para el Gobierno de la Ciudad de Rosario (2009), liberado públicamente al año siguiente. El segundo es un sistema para ejecutar actas de infracciones de tránsito a través de dispositivos móviles desarrollado en el año 2013 que, al igual que el primero, fue liberado al público bajo licencia.

Después de la liberación pública de estos sistemas, muchos municipios los implementaron (Santa Fe, Nogoya, Córdoba, Morón, Paraná, Villaguay, etc.), generando demanda de servicios de implementación y soporte, en algunos casos directamente a Tecso. En cierta medida, la liberación ha servido a la Cooperativa como una vía de promoción de la firma, permitiendo nuevos negocios.

Por lo tanto, sus productos se concentran en soluciones de ERP (siglas en inglés, Planificación de Recursos Empresariales) y desarrollo de sistemas de software completo para el sector público junto con su posterior liberación. También ofrece el servicio de implementación de algunas soluciones privativas extranjeras -de IBM- al sector industrial (particularmente en torno a la automatización) y Microsoft (un CRM, MS Dynamics).

En cuanto a las actividades de innovación, Tecso realiza principalmente tres: I+D interna, liberación de productos y desarrollo de productos OS. Atribuyen especial valor a la innovación de tipo organizacional que supone el hecho de funcionar como cooperativa, vinculando los aspectos técnicos de la producción a los aspectos organizacionales y económicos de la distribución de los beneficios.

Por otro lado, la cooperativa dedica esfuerzos a realizar un proyecto social cada año. Los miembros de la cooperativa votan por una contribución social, que no es una asignación de trabajo a un determinado desarrollo de software, sino algún tipo de donación a un grupo social. Los proyectos incluyen desarrollos de llave en mano, fábricas de software, mantenimiento correctivo y evolutivo de sistemas y desarrollo de productos.

En la actualidad, cada año definen un plan anual de capacitación (técnica), en concordancia con el plan estratégico general de la empresa. Adicionalmente tienen una política de capacitación sobre el modelo organizacional (cooperativo), como parte de lo que llaman “proceso de inducción” de los aspirantes a formar parte de la cooperativa.

En cuanto a la participación de TECSO con la Comunidad FLOSS, colaboran con las comunidades Tryton y Odo, competidoras en el área de ERP de software libre, aunque la cooperativa participa más en Tryton por tener una comunidad más grande que Odo. La colaboración se realiza durante el tiempo de trabajo,

mientras un proyecto en ERP esté activo, pero la empresa no colabora con la liberación de código o módulos en estas comunidades.

Los vínculos FLOSS incluyen también la participación en FACTTIC (Federación Argentina de Cooperativas de Trabajo de Tecnología, Innovación y Conocimiento), con otras cooperativas de trabajo FLOSS y TI. Este espacio es considerado por las empresas como una comunidad, en que muchos procesos de colaboración con otras empresas de FLOSS surgen de FACTTIC.

Respecto a su vinculación con las instituciones, se han vinculado con el Centro de Innovación Tecnológica, Empresarial y Social (CITES), la incubadora que el Grupo Sancor Seguros tiene en Sunchales (Santa Fe), para dos proyectos puntuales: uno relacionado con drones y una plataforma de procesamiento de imágenes más orientada a agro, seguridad y la parte industrial.

Si bien poseen como filosofía la promoción del software libre y la prefieren, no son “limitantes a”. Es decir, suelen vincularse con productos propietarios que son representados y comercializados para después realizar servicios sobre esos productos. Por lo tanto, se podría decir que poseen un modelo de negocios híbrido, en que participan de comunidades FLOSS como Tryton y Odoó así como también comercializan con productos propietarios.

G. Entornos Educativos

La empresa Entornos Educativos se ubica, desde el año 2006, en la Ciudad Autónoma de Buenos Aires en la Provincia de Buenos Aires, con aproximadamente 14 empleados. Es la primera empresa en América Latina en ofrecer servicios asociados a la plataforma Moodle y el único *partner* oficial en el cono sur.

En particular, la empresa se especializa en implementar plataformas de educación y capacitación a distancia Moodle, que pueden ser parametrizadas o desarrollos a medida. Sus clientes son referidos al ámbito educativo tales como universidades, escuelas y empresas que poseen una dispersión geográfica que los limita a la hora de dictar cursos presenciales. Los clientes mencionados son todos de origen nacional, por lo que el destino de las ventas de la empresa es totalmente nacional y, además, el 100% de las ventas proviene de productos y servicios vinculados al FLOSS.

Las áreas de la empresa son: administradores de sistemas, desarrolladores, consultoras, contenidistas y administración gerencial. También realizan actividades de investigación y desarrollo como también de innovación, debido a que el FLOSS que utilizan carece de funcionalidades que requieren para su empresa,

como por ejemplo, implementar una versión en español del software dado el carácter nacional de sus clientes. Todos sus proyectos internos son orientados inicialmente a un cliente, aunque no han realizado proyectos de co-desarrollo con otras empresas de software. Por lo general, no han liderado proyectos, sino que aportan en mejoras y módulos al software general.

Asimismo el personal contratado debe tener formación en FLOSS y, dadas las características del Moodle, la liberación es un proceso habitual. Aproximadamente se libera el 80% de los desarrollos, en que no existe ningún tipo de acuerdo para la liberación de los mismos.

En lo que se refiere a la Comunidad FLOSS, Entornos es *partner* de Moodle (estatus diferencial de la Comunidad), por lo que paga un canon del 10% de sus ventas a cambio de ser publicado en la web de Moodle, otorgándole la posibilidad de captar clientes. Las formas de colaboración de la empresa con la Comunidad incluyen aportes de códigos, corregir errores y dar talleres de difusión de Moodle. Estos se realizan dentro del horario de trabajo, aunque existen trabajadores “moodlers” que siguen colaborando en su tiempo personal (“gratis”) porque son parte de la comunidad.

3. Una síntesis de los casos de pymes estudiados

Luego de presentado los casos, se puede concluir que las empresas que forman parte de la comunidad FLOSS se benefician de su conocimiento libre, aunque algunas se benefician más que otras, así como también algunas realizan más aportes a la comunidad y tienen mayor participación respecto a otras. En la tabla a continuación se presenta una síntesis de las actividades que desarrolla cada empresa y algunos de los principales clientes que poseían al momento de las entrevistas. Además, se evidencia el año de inicio declarado de sus actividades.

Tabla 4.2. Caracterización productiva de las PyMEs FLOSS argentinas

<i>Empresa</i>	<i>Año de Inicio</i>	<i>Principales Actividades</i>	<i>Principales Clientes</i>
A. GCoop	2007	Desarrollos web, Desarrollos a medida, Gestión de Relación con el Cliente (CRM), Planificación de Recursos Empresariales (ERP), Capacitaciones.	Banco Credicoop, Dirección de Bibliotecas y Promoción de la Lectura de la Pcia. de Bs As, Fecootra-CNCT, Red Cooperativa de Comunicaciones (RCC).
B. XTech	1999	Consultoría de Infraestructura IT, Call Center de IT, Reclutamientos de talentos IT.	Garbarino, Administración Pública Nacional.
C. ECIC Systems	1986	Desarrollos de SL, Desarrollos a medida y servicios de productos asociados.	Empresas de servicios (gastronomía, hotelería y urgencias mecánicas del automotor) y sector industrial.
D. Kunan	2004	CRM, DBA remoto, Big Data Vertica y Aplicaciones móviles.	Disytel, Grupo Casino-Libertad, Renxo, Celta, URG urgencias, ProCórdoba.
E. Machinalis	2009	Desarrollo de SL y servicios en las áreas de Inteligencia Artificial, Data Processing, Big Data o Data Science y Servicios de Desarrollo Complejo.	Exportaciones a Estados Unidos.
F.TECSO	2002	Servicios de diseño, desarrollo e implementación de Software y Servicios Informáticos.	Citibank, Toyota, Volkswagen, Banco Credicoop, Banco Comafi, Banco General de Inversiones, Interbanking, Río Uruguay Seguros, Petrobras, IBM, Compañía Mega (Repsol), Municipalidad de Rosario, Santa Fe, Paraná, Morón, Nogoyá, Villaguay, entre otros.
G. Entornos Educativos	2006	Implementación de plataformas de educación y capacitación a distancia Moodle.	Universidades, escuelas y empresas que poseen una dispersión geográfica limitante.

En general, la mayoría de las empresas entrevistadas tiene como principal actividad el desarrollo de software libres ya sean “a medida” o no, así como también actividades orientadas a la gestión de relación con el cliente (CRM). Existen casos más aislados en lo que refiere a su actividad principal, como el de Entornos Educativos, que se dedica a la implementación de plataformas de educación y capacitaciones a través de Moodle, siendo la única empresa que se dedica a esta actividad dentro del grupo de empresas entrevistadas.

En lo que respecta a los clientes, presentan una gama heterogénea. Se distinguen clientes tanto nacionales como internacionales, del ámbito público y privado. También se observan clientes con distintas actividades productivas, tales como: financieras, bancarias, de servicios, de producción, venta de bienes, entre otras.

Las empresas también se pueden clasificar según el uso y valorización que posean con el FLOSS. Es decir, que tipo de vínculo posee o cómo se relaciona con la comunidad FLOSS y las liberaciones que realiza (o no) a la misma, como también la importancia que poseen las actividades de innovación en el contexto en que se encuentran realizando sus actividades productivas. En la tabla a continuación se presenta tal información de manera sintetizada.

Tabla 4.3. Las PyMEs argentinas y el FLOSS

<i>Empresa</i>	<i>¿Realiza actividades de Innovación?</i>	<i>¿Realiza liberaciones a la Comunidad?</i>	<i>Comunidades FLOSS donde participa</i>
A. GCoop	Sí	Sí	Drupal, Suite CRM, Tryton ERP, Django
B. XTech	No	Sí, en menor medida	Django (Python), Zymbra
C. ECIC Systems	Sí	Sí	Fedora
D. Kunan	Sí	Sí, en menor medida	Suite CRM, Libertya
E. Machinalis	Sí	Sí	Django (Python)
F. TECSO	Sí	Sí	Tryton, Odoo
G. Entornos Educativos	Sí	Sí	Moodle

Mientras uno de los casos (XTech) tiene una colaboración menor con la comunidad, donde los integrantes de la empresa se sirven de las capacidades y recursos que necesitan de la comunidad FLOSS para la prestación de sus servicios de infraestructura, sin que medie un tipo de colaboración activa y retributiva con la comunidad; cuatro de los casos analizados pueden ser clasificados como empresas comensalistas³ en su tipo de vinculación (Tecso, Kunan, Entornos Educativos y ECIC System). Todas éstas empresas procuran respetar y obedecer las normas y valores de la comunidad, incluso de obtener aceptación de su uso de los recursos comunitarios con fines comerciales. Suelen destinar personal para trabajar en la comunidad, pero mantienen su involucramiento en la comunidad con bajas posibilidades de influenciarla, haciendo aportes más bien incrementales y manteniendo la comunidad una dirección mayormente indiferente a la participación de los integrantes estas empresas.

Sin embargo, estas empresas mantienen intensidades muy distintas en su colaboración, con posibilidades de aprendizaje muy distintas. En el caso de ECIC, la intensidad es baja (la participación en la comunidad es ocasional y acotada al tiempo personal y recursos de los trabajadores, sin destinar tiempo de trabajo para liberar desarrollos). Por el contrario, en el caso de Kunan, la intensidad de la colaboración es alta: mantienen una participación sostenida y durante el tiempo laboral, asignando tiempo a la liberación de desarrollos. En el caso de Entornos Educativos, además de que la participación en la comunidad Moodle es intensa y alta (asignan tiempo laboral a participar, liberan aportes moderados por la

3- Esto responde a la clasificación de Dahlander y Magnusson (2005). Estos autores proveen una tipología de empresas según su relacionamiento con la comunidad en tres clases: i) Simbiótica; ii) Comensalista; y iii) Parasitaria. En la relación simbiótica la empresa codesarrolla con la comunidad y por su cuenta, beneficiándose mutuamente; incluso creando y apoyando una comunidad directamente desde la empresa o proveyendo una infraestructura para estimular la interacción, además de contribuir con código, reportar bugs, etc. La empresa tiene una alta posibilidad de influir sobre la comunidad, aunque su estatus se apoya en el respeto de las normas y valores de ella. Por su parte, una empresa de tipo comensalista se beneficia de la comunidad sin dañarla ni beneficiarla; obteniendo inputs útiles para su negocio desde ella, procurando respetar sus normas y valores, obedecer sus licencias y obtener aceptación del uso con fines comerciales. Ocasionalmente destinan personal para trabajar en la comunidad, pero con un mínimo involucramiento en la comunidad, con bajas posibilidades de influenciarla, mayormente indiferente. Por último, un tipo hipotético es la empresa parasitaria, que se focaliza en su propio beneficio, sin tener en cuenta el impacto de sus acciones en la comunidad. En este sentido, buscan código y recursos útiles sin compartir las normas, valores y reglas de la comunidad y prácticamente no tienen posibilidad de influir sobre los desarrollos de la comunidad.

comunidad y la colaboración es activa y permanente); son *partners* locales del líder del proyecto.

En el caso de Tecso, la intensidad de colaboración es intermedia. Si bien como parte de las actividades laborales está la asignación de tiempo a la colaboración en la comunidad y a la liberación de desarrollos, la participación está condicionada a la vigencia de algún proyecto que así lo requiera, período donde ésta es activa. Además, las liberaciones se han realizado a través de repositorios públicos (*forkeando* software completo), sin invertir en el aporte de mejoras con referato de las comunidades FLOSS de referencia.

Luego, hay dos casos que pueden ser considerados como empresas que mantienen una colaboración simbiótica con sus comunidades FLOSS: Machinalis y Gcoop. El hecho de que Machinalis tenga una vinculación de este tipo se refleja en que sus miembros tienen una autoridad reconocida en la Comunidad Python Argentina y sus colaboraciones son valoradas por la comunidad Python internacional. La comunidad FLOSS de referencia no es indiferente a sus aportes, y los miembros de la empresa tienen una muy alta posibilidad y capacidad de influir sobre la organización y direcciones de la comunidad Python Argentina. La comunidad es vista como una extensión de la base de conocimiento de la empresa y gran parte de su estatus se apoya en el respeto de las normas y valores de ella; pero sobre todo en su carácter proactivo.

Igualmente el caso de GCoop muestra una relación de tipo simbiótica con sus comunidades FLOSS. Presenta una participación activa en tres comunidades FLOSS globales y en tres nacionales, contribuyendo con código al núcleo de algunos de estos FLOSS globales y realizando muchos aportes; han liderado proyectos propios; la colaboración se realiza con alta intensidad (durante el horario laboral, de modo permanente e invirtiendo en la liberación de desarrollos); cuenta con experiencias de co desarrollo en I+D colaborativa con otras cooperativa FLOSS; y muestran una participación muy activa en todo tipo de conferencias y eventos FLOSS. Son motorizadores de la comunidad FLOSS y sus participaciones afectan en alguna medida de la dirección de las mismas.

Este hecho de que el FLOSS haya sido adoptado por todo tipo de industrias y empresas, sin importar su naturaleza, da cuenta de cómo ha evolucionado la industria del software. En los últimos años, las empresas tradicionales se han dado cuenta de que cualquier tecnología que hayan desarrollado para ejecutar su infraestructura de software no es lo que realmente ofrece valor a sus clientes, por lo que tiene sentido colaborar en esta área, incluso si eso implicaba colaborar con

sus competidores. A lo largo de este capítulo se ha podido apreciar la manera en que pequeñas empresas de software montan ofertas productivas desarrollando vinculaciones tanto de tipo comensalista, como de corte simbiótica con las comunidades; lo que ilustra el gran potencial que tienen las comunidades FLOSS para la competitividad de las pymes del sector.

Conclusiones y Reflexiones Finales: La presencia del FLOSS en la industria del software y las limitaciones del *mainstream* en economía para comprenderlo

Desde distintas aristas, hemos ilustrado que la presencia empresarial dentro de la actividad del FLOSS se ha tornado un fenómeno notorio. Y esta presencia se trata de un hecho que está respaldado tanto por estadísticas globales de la industria del software, como por estadísticas locales del sector en la Argentina y estudios teórico-conceptuales sobre el rol económico-productivo del OS. El FLOSS se muestra como un aspecto característico del funcionamiento de la competencia, la producción y la innovación en el sector del software.

Con respecto a la producción, en los últimos años: 1) ha crecido sustancialmente la participación del FLOSS en la oferta productiva de las empresas del sector, y 2) sin abandonar el desarrollo de proyectos internos, las empresas – incluida la mayoría de las líderes a nivel internacional –adoptaron formas de desarrollo predominantemente comunitarias en muchas de sus divisiones de negocios, lo que implicó una profunda transformación en sus formas de generar y gestionar sus desarrollos tecnológicos.

En relación al primer punto, el libro muestra que muchas empresas que sólo producían software propietario se fueron sumando al mundo del FLOSS (Microsoft, IBM, Oracle, para citar algunas de las principales). En esa dirección, el Capítulo 1 aporta una dimensión cuantitativa del notable grado de penetración que en la actualidad el FLOSS tiene en el mercado mundial y en el de Argentina en particular. Utilizando información proveniente de una encuesta a más de 100 empresas que operan en el mercado argentino de software se señala que más del 41% de estas empresas realizan desarrollo de productos FLOSS o prestan servicios a partir de un FLOSS, tanto si éste ha sido producido internamente como si se trata de un software desarrollado por terceros. A su vez, en el Capítulo 3 se describen las diversas maneras a través de las cuáles las grandes empresas multinacionales del sector participan en proyectos FLOSS. Estas formas van

desde la liberación de código y programas desarrollados *in-house*, pasando por el financiamiento y/o la dedicación de equipos de trabajadores de la empresa a programar en proyectos colectivos, hasta la adquisición de empresas OS y plataformas de desarrollo colaborativo y repositorios de proyectos.

Simultáneamente, ya en relación al segundo punto, la participación de las empresas en comunidades de desarrolladores, liderándolas o en calidad de colaboradores o financistas, se ha ido tornado un fenómeno generalizado. Esto se visualiza tanto en el continuo y sostenido crecimiento cuantitativo y cualitativo de las Comunidades, como en el lugar que ha adquirido el FLOSS en los planes de negocio de las empresas, en las estrategias de producción y uso de software y en las maneras en que los actores se involucran para su producción. La dimensión en que la producción empresarial se ha integrado con el mundo del FLOSS es enorme.

En el Capítulo 2 se muestra la manera en que distintos tipos de comunidades alrededor de proyectos FLOSS, sean comerciales o comunitarios, tanto gestionados por una única firma o un grupo de ellas, generan la oportunidad de montar esquemas de negocios tanto a PyMes como a grandes empresas de software. En el capítulo 4 hemos podido apreciar casos de pymes nacionales de software que han desarrollado parte de su oferta productiva a partir de comunidades FLOSS muy diversas como SugarCRM (Kunan), Python (Tecso, Gcoop y Machinalis), Drupal (Gcoop), Odoos (Tecso) o Moodle (Entornos Educativos), pymes que lo hacen a partir de un uso más laxo del OS (tales como XTech), y pymes que han establecido su oferta productiva nutrida de su participación en proyectos como Fedora, comunidad que interactúa con las estrategias de negocios de grandes empresas, como IBM/Red Hat.

Con respecto a la **innovación**, las comunidades FLOSS aparecen como un enorme *pool* de I+D colectivo. Este es un *pool* común para la innovación en diversas firmas, relativamente abierto, pues su aprovechamiento depende de los niveles de destrezas y competencias de los trabajadores de estas organizaciones, de la frecuencia con la que se colabora con estas comunidades, del grado de compromiso con las mismas y del respeto e interacción con sus normas y valores.

Raramente una sola organización podría lograr internamente desarrollos innovativos de la misma dimensión y complejidad de los que logran miles y miles de programadores trabajando incesante y descentralizadamente en un ecosistema como el que conforman los proyectos FLOSS. El ritmo, velocidad y calidad que adquiere la innovación en estas formas colaborativas de producción donde existen derechos colectivos de propiedad, se impone como superior en el mercado comparativamente a los esquemas privativos, especialmente para desarrollos de elevada complejidad.

Entre los elementos novedosos de esta forma de innovación “privada-colectiva” que caracteriza los desarrollos a través de comunidades abiertas hay tres que merecen ser destacados. El primero es que los esfuerzos innovativos no son directamente retribuidos en términos monetarios, aunque generan costos monetarios y no monetarios. Algunos colaboradores, aquellos que son contratados por empresas para dicho fin, reciben un pago monetario de la empresa contratante, pero no de la comunidad a la que aportaron sus conocimientos. El segundo elemento que interesa destacar es la ausencia de derechos de propiedad intelectual que favorezcan la apropiación privada de los beneficios de la innovación o bloqueen el uso de los programas a terceros interesados (hayan o no hayan aportado esfuerzos innovativos o recursos al desarrollo). Y el tercero se refiere a las variadas formas de apropiación de los beneficios de parte de los diferentes participantes. En el Capítulo 3 se mencionan los principales beneficios que obtienen las empresas de su participación en proyectos comunitarios: distribución de los costos de desarrollo, reducción de la incertidumbre, identificación de recursos humanos de altísima calificación, acceso a nuevos conocimientos y activos complementarios, compatibilidad de los desarrollos con los principales sistemas y aplicaciones de la empresa, resolución de fallos de mercado, un acceso más rápido a nuevos mercados, legitimidad y reputación, garantías para la inversión, etc. También en ese capítulo se muestra cómo en el nicho tecnológico más dinámico del sector, el de *cloud computing* empresarial, el FLOSS al garantizar la compatibilidad e interoperabilidad entre los diversos sistemas y aplicaciones de las empresas competidoras permite la generación de externalidades de red y conduce a la obtención de rendimientos crecientes dinámicos, que acaban beneficiando tanto a productores como a los usuarios de dichos sistemas.

Con respecto a la **competencia**, a la intensa rivalidad interempresarial propia de una actividad en fuerte expansión, el FLOSS le sumó una extendida y vigorosa cooperación. El cooperar ha pasado a ser una estrategia de competencia. La “coopetencia abierta” u “open coopeition”, características del FLOSS, se diferencia de las tradicionales formas de “coopetencia” en el sentido de que no limita la participación a los integrantes de un *joint venture*, un clúster o una alianza -que terminarían conformando un espacio de cooperación cerrado aunque fuera de carácter multi-organizacional- sino que es un espacio de cooperación constitutivamente abierto a todos los que quieran participar con posibilidad tanto de acceso inmediato a los productos en competencia, como de apropiación inmediata de los beneficios que ellos pueden generar.

El análisis que se efectúa en el Capítulo 3 sobre lo que ocurre actualmente en el segmento de servicios de infraestructura en base a *cloud computing* ilustra sobre distintas características que puede asumir esta coopetencia abierta. Por un lado, encontramos empresas competidoras que cooperan en un mismo desarrollo de

infraestructura de nube para grandes volúmenes de datos. Por ejemplo, Microsoft, IBM, SAP y muchas otras colaboran en el proyecto OpenStack. Por otro lado, esas mismas empresas simultáneamente colaboran en algunos de los otros proyectos FLOSS “competidores”, tales como CloudStack, Eucalyptus y OpenNebula. Pero incluso más, las empresas competidoras más grandes desarrollan herramientas que son usadas o aportan al desarrollo de las nubes propias de sus competidoras, como en se observa en la relación entre Google y Microsoft, y algunos otros ejemplos del tipo. Es decir, las empresas competidoras colaboran aportando recursos, códigos y/u horas de programación a más de un proyecto de la misma clase y en más de un proyecto de distinta clase pero que redundan en beneficios para los servicios ofrecidos por la competencia.

Esta situación no es exclusiva del caso analizado. A lo largo del libro se presentan otros casos tanto de grandes empresas como pymes que colaboran en proyectos FLOSS “rivales” en la misma área de negocios. Así por ejemplo, en el Capítulo 4, dedicado al análisis del mundo FLOSS en la pymes, se describe el caso de Tecso, empresa que en el área de desarrollos de software ERP colabora simultáneamente en dos proyectos “competidores” (Tryton y Odoo).

El apoyo, la participación y la colaboración en estos proyectos de desarrollo colectivo de FLOSS de algunos grandes actores del sector se han tornado una manera de competir en áreas tecnológicas específicas. Al respecto, es interesante destacar que en algunos casos el entusiasmo con el que algunos de los líderes tecnológicos del sector han adherido al desarrollo de iniciativas FLOSS parece estar relacionado con el lugar que ocupan en el área en cuestión. En estas situaciones, empresas que históricamente desarrollaron software propietario y quedan rezagadas en la lucha competitiva son las más proclives a apostar por el desarrollo del FLOSS, como una vía para reducir la brecha respecto de la o las líderes.

Uno de estos casos ocurre en el segmento de desarrollo de software para la provisión de servicios de infraestructura *cloud computing*. El liderazgo lo posee una firma privativa, que no colabora explícitamente en los proyectos FLOSS competidores, que es el caso de Amazon con Amazon EC2. Pero el liderazgo de Amazon ha llevado a antiguos enemigos acérrimos del FLOSS, como Microsoft a revertir totalmente su aversión al OS, y pasar a cooperar en un proyecto de código abierto como OpenStack, a lanzar su primera distribución Linux (Azure Stack OS), liberar patentes y adquirir repositorios OS; en su búsqueda por desplazar a Amazon del liderazgo occidental en *cloud computing* empresarial.

Sin embargo, esta no es la única experiencia ilustrativa de este tipo. Mientras Microsoft durante la década de los 1990's lideraba el segmento de servidores web, IBM abandonó sus esfuerzos de desarrollo de software *in-house* de este tipo para volcar esos esfuerzos a colaborar en esta línea tecnológica con el proyecto

FLOSS Apache. Así, IBM pasó a liberar y donar casi 500.000 líneas de código de su programa Cloudscape a la Fundación Apache (producto que fue renombrado como Apache Derby). Apache ha mantenido siempre una cuota de mercado pareja con Microsoft en el segmento de servidores y, por momentos, dominante. De esta manera, la participación de grandes empresas en algunos de estos proyectos también puede verse como una modalidad para contribuir al debilitamiento de la posición tecnológica de quien domina un segmento determinado.

Todo esto indica que el FLOSS se ha configurado como espacio de cooperación y competencia entre las empresas, que el espacio de las comunidades desarrolladoras de software es hoy concurrente al mercado. Concurrente en los distintos sentidos que tiene esta palabra. Por un lado, en el sentido de juntarse en un mismo lugar o tiempo. Las comunidades pueden ser vistas como el lugar en que las empresas se juntan y se vinculan entre sí. En segundo lugar, en el sentido de coincidir en alguien o en algo. En estos espacios hay coincidencias en algunos objetivos, como por ejemplo el aprendizaje y la generación y desarrollo de nuevos softwares. En tercer lugar, en el sentido de contribuir con una cantidad para determinado fin. Las comunidades para su financiamiento suelen contar con la ayuda de empresas aportantes y contribuyentes. Y finalmente, en el sentido más restringido de competir, en dos planos. En un plano infraestructural, o estrictamente mercantil, las empresas se apoyan en las comunidades para competir entre sí. Y en un plano superestructural, o de la gobernanza, la lógica empresarial entra en cierta tensión con la comunitaria: en estos espacios, especialmente en aquellos cuya gobernanza no está en manos empresariales, hay una competencia o “tensión” entre la lógica de las comunidades, centrada en los beneficios colectivos y la lógica del mercado, centrada en los beneficios individuales. Tal como se señala en el Capítulo 2, la gobernanza de las comunidades establece límites a la capacidad de apropiación y direccionamiento del trabajo de las comunidades de parte de empresas particulares. Visualizar la existencia de estas tensiones es central para entender no solo el funcionamiento de las comunidades sino también para comprender la lógica empresarial en el mundo del software en la actualidad.

En definitiva, el FLOSS favorece el desarrollo de procesos de aprendizaje (estimulados por la interacción con cientos o miles de otros agentes), permite el desarrollo de proyectos complejos, aumenta el grado de competencia (muchos competidores pueden acceder a la misma o a una relativamente similar tecnología), y acelera el ritmo de innovación (por el trabajo colaborativo y la aceleración del aprendizaje, pero también por la mayor competencia en un mercado en vertiginosa y permanente transformación, en el que las posiciones logradas se pueden perder muy rápidamente). Además de estas “virtudes”, también se debe destacar que el FLOSS tiene un rol crucial en la apertura de nuevas oportunidades de negocios a

micro, pequeñas y medianas empresas, permitiéndoles, al menos en algunas áreas, comerciar productos y servicios sin rezago tecnológico, a pesar de sus limitadas capacidades de inversión en actividades de investigación y desarrollo.

A pesar de la magnitud que ha alcanzado el FLOSS, tal como se señala en el Capítulo 1, y de su creciente participación en el mercado del software esta actividad está invisibilizada en las estadísticas de producción, valor agregado y empleo. Tampoco los economistas se han preocupado demasiado por analizar las transformaciones que el fenómeno del OS y sus comunidades han introducido en las formas de producir, innovar y competir en el sector del software. Nuestra hipótesis es que este deficiente tratamiento que ha recibido el FLOSS de parte de los economistas tiene mucho que ver con ciertas particularidades de la teoría económica hegemónica o *mainstream*. Básicamente hay cuatro elementos o razones que deben ser citadas al respecto

En primer lugar, el simplista tratamiento que se hace de la empresa. Ésta es analizada como si fuera una “caja negra”, sin que interese nada de lo que pasa en su interior. Utilizando la noción de función de producción, que se supone dada e igual para todas o al menos para cada subgrupo de las empresas del mercado, se puede conocer cuánto va a producir cada empresa dada la dotación de recursos productivos que dispone. Pero no hay lugar para el aprendizaje y la acumulación de conocimientos al interior de las empresas. Es por ello que no hay lugar para el análisis y la consideración de los procesos de aprendizaje colectivo y de innovación que tienen lugar en las comunidades.

En segundo lugar, su incompreensión del fenómeno de la actividad bajo código abierto en general. Esto es debido a la excesiva importancia que le da el *mainstream* a la apropiabilidad privada, y en particular a los instrumentos de propiedad intelectual, como incentivo para la existencia de innovaciones. Por el contrario, a lo largo del libro se visualiza que en muchos casos es precisamente “la ausencia de propiedad privada” sobre los desarrollos que se generan la condición que alienta, permite y acelera la generación de innovaciones. El aferramiento dogmático del *mainstream* a la importancia de los derechos de propiedad privada, como una precondition para la existencia de innovación y beneficios económicos en economías de mercado, ha diseminado una visión económica del OS como un fenómeno anómalo, que requiere una explicación excepcional. Consideramos que ha quedado suficientemente mostrado en este libro que si algo dista de constituir el FLOSS en el sector del software es un caso excepcional: progresivamente se convierte en la norma constitutiva de producir software.

En tercer lugar, la incompleta y limitada conceptualización que posee el *mainstream* de los procesos de innovación constituye una barrera a la comprensión del OS, puesto que no cuenta con un instrumental que sea adecuado para estudiar

tanto las condiciones que hacen posible la emergencia de innovaciones, como las formas de apropiación de las novedades tecnológicas que vayan más allá de los mecanismos de propiedad intelectual y el mercado.

Por último, el *mainstream* otorga una consideración casi nula a la temática de la cooperación inter-empresarial, más allá de una forma de colusión. Ello ha derivado en la invisibilidad, en términos analíticos, de las Comunidades FLOSS como un espacio económico de cooperación relevante para la producción de software y la innovación. Como las vinculaciones que establece entre distintos agentes ocurren en un espacio diferente al del mercado, sus lógicas, normas y valores están por fuera de lo que el *mainstream* considera “economía”. Sin embargo, las comunidades son espacios no sólo de actividad económica y de inducción de la actividad innovadora, sino también de valorización para las empresas. La realidad económica, como tantas otras veces, pasa indiferente al costado de las funciones de producción, curvas de costos y problemas de optimización restringida de la economía ortodoxa.

En este libro hemos procurado reunir una serie de instrumentos analíticos, que provienen de distintas disciplinas, como la sociología, las ciencias políticas y las ciencias informáticas; para poder llevar adelante la investigación académica de la producción de software de código abierto y *open source*, desde un punto de vista económico. La investigación en nuestro equipo de investigación tiende a sugerirnos que el FLOSS y sus lógicas, van adquiriendo una relevancia creciente en la actividad del software en general. Ello también abre a reflexionar más ampliamente sobre la medida en que las formas de cooperación y competencia en el terreno de la producción abierta y colaborativa que caracterizan al FLOSS, y su impacto en el ritmo y calidad de la innovación; puedan reproducirse en otros sectores y mercados. Sus prácticas, que guardan semejanzas con las de las comunidades científicas, se han ido expandiendo a otra serie de actividades productivas, que van desde la fabricación de semillas a la de medicamentos, pasando por la investigación genética y la producción electrónica en plataformas de hardware de código abierto; todas actividades económicas y productivas en que la economía ortodoxa tiene muy malas condiciones para poder comprender.

Bibliografía

- AALTONEN, T. y JOKINEN, J. Influence in the Linux kernel community. IFIP International Conference on Open Source Systems, 2007. Springer, 203-208.
- AKSULU, A. y WADE, M. (2010) "A comprehensive review and synthesis of open source research." *Journal of the Association for Information Systems*, 11, 11, p. 576.
- AMANT, K. y STILL, B. (2007) *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives: Technological, Economic, and Social Perspectives*. Hershey - New York: IGI Global.
- ARMBRUST, M., FOX, A., GRIFFITH, R., JOSEPH, A. D., KATZ, R., KONWINSKI, A., LEE, G., PATTERSON, D., RABKIN, A. y STOICA, I. (2010) "A view of Cloud Computing, Communications of the ACM." *vol*, 53, p. 5058.
- BALDWIN, C. Y. y CLARK, K. B. (2006) "The architecture of participation: Does code architecture mitigate free riding in the open source development model?" *Management Science*, 52, 7, p. 1116-1127.
- BENGTSSON, M. y KOCK, S. (2014) "Coopetition—Quo vadis? Past accomplishments and future challenges." *Industrial marketing management*, 43, 2, p. 180-188.
- BENKLER, Y. (2017). "Open access and information commons", en: F. Parisi (ed.) *The Oxford Handbook of Law and Economics*. Cambridge: Oxford University Press. pp. 256.
- BENKLER, Y., SHAW, A. y HILL, B. M. (2015) "Peer production: A form of collective intelligence." *Handbook of collective intelligence*, 175, p.
- BERDOU, E. Insiders and outsiders: paid contributors and the dynamics of cooperation in community led F/OS projects. IFIP International Conference on Open Source Systems, 2006. Springer, 201-208.
- _____ (2010) *Organization in open source communities: At the crossroads of the gift and market economies*. Routledge.
- BITZER, J. y SCHRÖDER, P. J. (2006) *The economics of open source software development*. Elsevier Science Inc.
- BOIDI, S. y GIARDINI, P. (2006). Implantación de Software Libre y Recambio de Tecnología Informática en la Municipalidad de Rosario. *35º Jornadas Argentinas de Informática e Investigación Operativa*. Mendoza.
- BONACCORSI, A., GIANNANGELI, S. y ROSSI, C. (2006) "Entry strategies under competing standards: Hybrid business models in the open source

- software industry.” *Management Science*, 52, 7, p. 1085-1098.
- BONACCORSI, A. y ROSSI, C. (2003) “Why Open Source software can succeed.” *Research policy*, 32, 7, p. 1243-1258.
- _____ (2006) “Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business.” *Knowledge, Technology & Policy*, 18, 4, p. 40-64.
- BONACCORSI, A. y ROSSI LAMASTRA, C. (2004) “Altruistic individuals, selfish firms? The structure of motivation in Open Source software.” *First Monday*, 9, 1-5, January 2004, p. 1-6.
- BORDELEAU, F., MEIRELLES, P. y SILLITTI, A. Fifteen Years of Open Source Software Evolution. IFIP International Conference on Open Source Systems, 2019. Springer, 61-67.
- BORRASTERO, C. y MORERO, H. A. (2014). Inteligencia Artificial desde la Periferia: un caso de Córdoba (Argentina). *I Congreso de Investigación Cualitativa en Ciencias Sociales / I Post Congreso International Congress of Qualitative Inquiry*. Córdoba, Argentina.
- BRANDENBURGER, A. M. y NALEBUFF, B. J. (2011) *Co-opetition*. Crown Business.
- BRINK, D., ROOS, L., WELLER, J. y VAN BELLE, J. Critical Success Factors for Migrating to OSS-on-the-Desktop: Common Themes across Three South African Case Studies. IFIP International Conference on Open Source Systems, 2006. Springer, 123-134.
- BRISCOE, G. y MARINOS, A. Digital ecosystems in the clouds: towards community cloud computing. 2009 3rd IEEE international conference on digital ecosystems and technologies, 2009. IEEE, 103-108.
- CAPILUPPI, A. y MICHLMAYR, M. From the cathedral to the bazaar: An empirical study of the lifecycle of volunteer community projects. IFIP International Conference on Open Source Systems, 2007. Springer, 31-44.
- CAPILUPPI, A., STOL, K.-J. y BOLDYREFE, C. Exploring the role of commercial stakeholders in open source software evolution. IFIP International Conference on Open Source Systems, 2012. Springer, 178-200.
- CAPRA, E., FRANCALANCI, C., MERLO, F. y LAMASTRA, C. R. A survey on firms’ participation in open source community projects. IFIP International Conference on Open Source Systems, 2009. Springer, 225-236.
- CAPRA, E., FRANCALANCI, C., MERLO, F. y ROSSI-LAMASTRA, C. (2011) “Firms’ involvement in Open Source projects: A trade-off between

- software structural quality and popularity.” *Journal of Systems and Software*, 84, 1, p. 144-161.
- CASTELLO, R., BOLLO, D., ARÓNICA, S., GAUNA, E., PERETTI, F. y ROCHA VARGAS, M. (2009). Modelos de negocio en Software Libre. 38 *JAIIO - Jornadas Argentinas de Informática - SADIO*. Mar del Plata: SADIO.
- CENATIC (2009). Informe sobre el estado del arte del Software de Fuentes Abiertas en la empresa española. 2009.: Observatorio Nacional del Software de Fuentes Abiertas (ONSFA), Centro de Excelencia de Software de Fuentes Abiertas, Red.es / Ministerio de Industria, Energía y Turismo (MINETUR), Gobierno de España.
- _____ (2011). El Software Libre en el Sector Español de Servicios Informáticos. Informe de resultados de la Encuesta sobre el Software de Fuentes Abiertas en el sector SI español (ESFA-SI) 2010-2011. Observatorio Nacional del Software de Fuentes Abiertas (ONSFA), Centro de Excelencia de Software de Fuentes Abiertas, Red.es / Ministerio de Industria, Energía y Turismo (MINETUR), Gobierno de España.
- CLARYSSE, B., WRIGHT, M., BRUNEEL, J. y MAHAJAN, A. (2014) “Creating value in ecosystems: Crossing the chasm between knowledge and business ecosystems.” *Research Policy*, 43, 7, p. 1164-1176.
- COLOMBO, M. G., PIVA, E. y ROSSI-LAMASTRA, C. (2013) “Authorising Employees to Collaborate with Communities During Working Hours: When is it Valuable for Firms?” *Long Range Planning*, 46, 3, p. 236-257.
- _____ (2014) “Open innovation and within-industry diversification in small and medium enterprises: The case of open source software firms.” *Research policy*, 43, 5, p. 891-902.
- CONKLIN, M. S. Beyond low-hanging fruit: Seeking the next generation in floss data mining. IFIP International Conference on Open Source Systems, 2006. Springer, 47-56.
- CROWSTON, K. y HOWISON, J. (2005) “The social structure of free and open source software development.” *First Monday*, 10, 2, p.
- _____ (2006) “Assessing the health of open source communities.” *Computer*, 39, 5, p. 89-91.
- CROWSTON, K. y SHAMSHURIN, I. (2017) “Core-periphery communication and the success of free/libre open source software projects.” *Journal of Internet Services and Applications*, 8, 1, p. 10.

- CHEN, M. J. y MILLER, D. (2015) "Reconceptualizing competitive dynamics: A multidimensional framework." *Strategic management journal*, 36, 5, p. 758-775.
- D'AGOSTINO, S. (2010). SL para la Gestión por Procesos y Resultados en la DGR de Misiones. *39º Jornadas Argentinas de Informática e Investigación Operativa*. Catamarca.
- DAHLANDER, L. y MAGNUSSON, M. (2008) "How do Firms Make Use of Open Source Communities?" *Long Range Planning*, 41, 6, p. 629-649.
- DAHLANDER, L. y MAGNUSSON, M. G. (2005) "Relationships between open source software companies and communities: Observations from Nordic firms." *Research Policy*, 34, 4, p. 481-493.
- DAHLANDER, L. y WALLIN, M. W. (2006) "A man on the inside: Unlocking communities as complementary assets." *Research policy*, 35, 8, p. 1243-1259.
- DALLE, J.-M. y JULLIEN, N. (2000) "Windows vs. Linux: some explorations into the economics of Free Software." *Advances in Complex Systems*, 3, 01n04, p. 399-416.
- _____ (2003) "Libre'software: turning fads into institutions?" *Research policy*, 32, 1, p. 1-11.
- DAVID, P. A. y RULLANI, F. (2008) "Dynamics of innovation in an "open source" collaboration environment: lurking, laboring, and launching FLOSS projects on SourceForge." *Industrial and Corporate Change*, 17, 4, p. 647-710.
- DE LAAT, P. B. (2007) "Governance of open source software: state of the art." *Journal of Management & Governance*, 11, 2, p. 165-177.
- DESHPANDE, A. y RIEHLE, D. The total growth of open source. IFIP International Conference on Open Source Systems, 2008. Springer, 197-209.
- DI TULLIO, D. y STAPLES, D. S. (2013) "The governance and control of open source software projects." *Journal of Management Information Systems*, 30, 3, p. 49-80.
- DUDLEY-SPONAUGLE, A., HONG, S. y WANG, Y. (2007). "The social and economical impact of OSS in developing countries", en: Kirk Amant & B. Still (eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. IGI Global. pp. 102-114.
- DUGHERA, L., YANSEN, G. y ZUKERFELD, M. (2012) *Gente con códigos. La heterogeneidad de los procesos productivos de software*. Buenos Aires: Universidad Maimónides.

- DURAND, D., VUATTOUX, J.-L. y DITSCHIED, P.-J. OSS in 2012: a long-term sustainable alternative for corporate IT. IFIP International Conference on Open Source Systems, 2012. Springer, 371-376.
- FORREST, D., JENSEN, C., MOHAN, N. y DAVIDSON, J. Exploring the role of outside organizations in Free/Open Source Software projects. IFIP International Conference on Open Source Systems, 2012. Springer, 201-215.
- GARTNER (2008) *User Survey Analysis: Open-Source Software, Worldwide*.
- GERMONPREZ, M., ALLEN, J. P., WARNER, B., HILL, J. y MCCLEMENTS, G. (2013) "Open source communities of competitors." *interactions*, 20, 6, p. 54-59.
- GIARDINI, P. y AGUZZI, V. (2011). Adopción y Liberación de Software Libre en la Municipalidad de Rosario, Argentina. *40º Jornadas Argentinas de Informática e Investigación Operativa*. Córdoba.
- GNYAWALI, D. R. y PARK, B. J. (2009) "Co-opetition and technological innovation in small and medium-sized enterprises: A multilevel conceptual model." *Journal of Small Business Management*, 47, 3, p. 308-330.
- GODFREY, M. y TU, Q. Growth, evolution, and structural change in open source software. Proceedings of the 4th international workshop on principles of software evolution, 2001. ACM, 103-106.
- GONELLA, C. y SCHELSKE, S. (2010). SL para la Gestión por Procesos y Resultados en la DGR de Misiones. *39º Jornadas Argentinas de Informática e Investigación Operativa*. Catamarca.
- GRANOVETTER, M. (1992) "Economic institutions as social constructions: a framework for analysis." *Acta sociologica*, 35, 1, p. 3-11.
- HAMEL, G., DOZ, Y. L. y PRAHALAD, C. K. (1989) "Collaborate with your competitors and win." *Harvard business review*, 67, 1, p. 133-139.
- HARDY, J.-L. y BOURGOI, M. Exploring the potential of OSS in Air Traffic Management IFIP International Conference on Open Source Systems, 2006. Springer, 123-134.
- HARISON, E. y KOSKI, H. (2010) "Applying open innovation in business strategies: Evidence from Finnish software firms." *Research policy*, 39, 3, p. 351-359.
- HARS, A. y OU, S. (2001) "Working for free? Motivations of participating in open source projects." *International Journal of Electronic Commerce*, 6, p. 25-39.

- HAUGE, Ø., SØRENSEN, C.-F. y CONRADI, R. Adoption of open source in the software industry. IFIP International Conference on Open Source Systems, 2008. Springer, 211-221.
- HAUGE, Ø. y ZIEMER, S. Providing Commercial Open Source Software: Lessons Learned. IFIP International Conference on Open Source Systems, 2009. Springer, 70-82.
- HECKER, F. (1999) "Setting up shop: The business of open-source software." *IEEE software*, 16, 1, p. Latest version revised on June 2000. Revision 0.8 available on <http://hecker.org/writings/setting-up-shop>.
- HEINZ, F. (2006) "¿ Competitividad Informática, o Competitividad Social?" *BORELLO, J., ROBERT, V. y YOGUEL, G.(2006) La informática en la Argentina. Desafíos a la especialización ya la competitividad. Prometeo-UNGS, Buenos Aires*, p.
- HENKEL, J. (2004) "Open source software from commercial firms—tools, complements, and collective invention." *Zeitschrift für Betriebswirtschaft*, 4, p. 1-23.
- HENKEL, J. (2009) "Champions of revealing - The role of open source developers in commercial firms." *Industrial and Corporate Change*, 18, 3, p. 435-471.
- HIPPEL, E. V. y KROGH, G. V. (2003) "Open source software and the "private-collective" innovation model: Issues for organization science." *Organization science*, 14, 2, p. 209-223.
- IANSITI, M. y LEVIEN, R. (2004a) *The keystone advantage: what the new dynamics of business ecosystems mean for strategy, innovation, and sustainability*. Harvard Business Press.
- _____ (2004b) "Strategy as ecology." *Harvard business review*, 82, 3, p. 68-81.
- JAAKSI, A. Experiences on product development with open source software. IFIP International Conference on Open Source Systems, 2007. Springer, 85-96.
- JENSEN, C. y SCACCHI, W. Governance in open source software development projects: A comparative multi-level analysis. IFIP International Conference on Open Source Systems, 2010. Springer, 130-142.
- JULLIEN, N. y ZIMMERMANN, J.-B. (2011) "Floss firms, users and communities: a viable match?" *Journal of Innovation Economics & Management*, 1, p. 31-53.

- KAMEI, Y., MATSUMOTO, S., MAESHIMA, H., ONISHI, Y., OHIRA, M. y MATSUMOTO, K.-I. Analysis of coordination between developers and users in the apache community. IFIP International Conference on Open Source Systems, 2008. Springer, 81-92.
- KIM, S., YOO, J. y LEE, M. Step-by-step strategies and case studies for embedded software companies to adapt to the foss ecosystem. IFIP International Conference on Open Source Systems, 2012. Springer, 48-60.
- KOCH, S. Evolution of open source software systems—a large-scale investigation. Proceedings of the 1st International Conference on Open Source Systems, 2005. 148-153.
- KOCH, S. y GONZALEZ-BARAHONA, J. M. (2005) “Open Source software engineering-The state of research.” *First Monday*, p.
- KOGUT, B. y METIU, A. (2001) “Open-source software development and distributed innovation.” *Oxford Review of Economic Policy*, 17, 2, p. 248-264.
- KUECHLER, V., GILBERTSON, C. y JENSEN, C. Gender differences in early free and open source software joining process. IFIP International Conference on Open Source Systems, 2012. Springer, 78-93.
- LAKHANI, K., WOLF, B., BATES, J. y DIBONA, C. (2002) “The boston consulting group hacker survey.” *Boston, The Boston Consulting Group*, p.
- LAKHANI, K. R. y VON HIPPEL, E. (2003) “How open source software works: “free” user-to-user assistance.” *Research Policy*, 32, 6, p. 923-943.
- LEE, G. K. y COLE, R. E. (2003) “From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development.” *Organization science*, 14, 6, p. 633-649.
- LEE, S., BAEK, H. y JAHNG, J. (2017) “Governance strategies for open collaboration: Focusing on resource allocation in open source software development organizations.” *International Journal of Information Management*, 37, 5, p. 431-437.
- LERNER, J. y SCHANKERMAN, M. (2013) *The comingled code: Open source and economic development*. London, UK.
- LERNER, J. y TIROLE, J. (2000). The simple economics of open source. National Bureau of Economic Research, Working Paper 7600.
- LINDMAN, J. Shared assumption concerning technical determination in apache web server developer community. IFIP International Conference on Open Source Systems, 2007. Springer, 283-288.

- LINDMAN, J. y HAMMOUDA, I. Investigating Relationships Between FLOSS Foundations and FLOSS Projects. IFIP International Conference on Open Source Systems, 2017. Springer, Cham, 14-22.
- LINDMAN, J., JUUTILAINEN, J.-P. y ROSSI, M. Beyond the business model: Incentives for organizations to publish software source code. IFIP International Conference on Open Source Systems, 2009. Springer, 47-56.
- LINDMAN, J., ROSSI, M. y MARTTIIN, P. Applying open source development practices inside a company. IFIP International Conference on Open Source Systems, 2008. Springer, 381-387.
- MACCORMACK, A., RUSNAK, J. y BALDWIN, C. Y. (2006) "Exploring the structure of complex software designs: An empirical study of open source and proprietary code." *Management Science*, 52, 7, p. 1015-1030.
- MÄENPÄÄ, H., KILAMO, T., MIKKONEN, T. y MÄNNISTÖ, T. Designing for participation: three models for developer involvement in hybrid Oss projects. IFIP International Conference on Open Source Systems, 2017. Springer, 23-33.
- MARKUS, M. L. (2007) "The governance of free/open source software projects: monolithic, multidimensional, or configurational?" *Journal of Management & Governance*, 11, 2, p. 151-163.
- MARTINEZ-ROMO, J., ROBLES, G., GONZALEZ-BARAHONA, J. M. y ORTUÑO-PEREZ, M. Using social network analysis techniques to study collaboration between a FLOSS community and a company. IFIP International Conference on Open Source Systems, 2008. Springer, 171-186.
- MELIAN, C. y MÄHRING, M. Lost and gained in translation: Adoption of open source software development at Hewlett-Packard. IFIP International Conference on Open Source Systems, 2008. Springer, 93-104.
- MERTON, R. K. (1977) *La sociología de la ciencia: investigaciones teóricas y empíricas*. Madrid: Alianza Editorial.
- MONCAUT, N. y ROBERT, V. (2016). Determinantes del uso y desarrollo de software libre en Argentina. *XXI Red Pymes Mercosur*. Tandil.
- MOORE, J. F. (1996) *The death of competition: leadership and strategy in the age of business ecosystems*. HarperBusiness New York.
- MORERO, H. A. y BORRASTERO, C. (2015). Modificaciones en la organización del trabajo en empresas productoras de software abierto y formas de apropiación. *Jornadas Vincular Córdoba*. Córdoba, Argentina.

- MORERO, H. A., BORRASTERO, C. y MOTTA, J. (2015) “Procesos de innovación en la producción de software en argentina. Un estudio de caso.” *Revista de Estudios Empresariales*, 2015, 2, p. 24-48.
- _____ (2017a). La medición de la innovación en empresas de software libre y open source (Free/Libre Open Source – FLOS). *XXII Reunión Anual de la Red Pymes MERCOSUR*. Montevideo, Uruguay.
- MORERO, H. A., BORRASTERO, C. y ORTIZ, P. (2014). Innovación en el Sector de Software. El caso de Machinalis. Documento de Proyecto. Reporte de Estudio de Caso. .
- MORERO, H. A., MOTTA, J., ORTIZ, P. y VÉLEZ, J. G. (2017b). Pequeñas empresas de software libre (FLOSS) en la Argentina. *XVII Congreso Latino-Iberoamericano de Gestión Tecnológica ALTEC*. Ciudad de México.
- MORERO, H. A., MOTTA, J., VÉLEZ, J. G. y ORTIZ, P. (2017c). Taxonomías de empresas de software libre y open source. Sus limitaciones a partir de casos de pymes argentinas. *XXII Reunión Anual de la Red Pymes MERCOSUR*. Montevideo, Uruguay.
- MORGAN, L. y FINNEGAN, P. (2007) “How perceptions of open source software influence adoption: an exploratory study.” *NUI Galway*, p.
- MOTTA, J. J., MORERO, H. A. y BORRASTERO, C. (2019) “Measuring Non Monetary Innovation In Software: a case study in floss firms from Argentina.” *International Journal of Innovation: IJI Journal*, 7, 1, p. 135-154.
- MOWERY, D. C. (1999) “The computer software industry.” *Sources of industrial leadership: Studies of seven industries*, p. 133-168.
- MUNGA, N., FOGWILL, T. y WILLIAMS, Q. The adoption of open source software in business models: a Red Hat and IBM case study. Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists, 2009. ACM, 112-121.
- MUSTONEN, M. (2003) “Copyleft—the economics of Linux and other open source software.” *Information Economics and Policy*, 15, 1, p. 99-121.
- NAKAKOJI, K., YAMAMOTO, Y., NISHINAKA, Y., KISHIDA, K. y YE, Y. Evolution patterns of open-source software systems and communities. Proceedings of the international workshop on Principles of software evolution, 2002. ACM, 76-85.
- NIELSEN, J. (2006). The 90-9-1 rule for participation inequality in social media and online communities.

- NIKULA, U. y JANTUNEN, S. Quantifying the interest in open source system: case south-east Finland. Proceedings of the 1st International Conference on Open Source Systems (Scotto, M. and Succi, G. Eds.), 2005. 192-95.
- NOLL, J., SEICHTER, D. y BEECHAM, S. A qualitative method for mining open source software repositories. IFIP International Conference on Open Source Systems, 2012. Springer, 256-261.
- NYMAN, L. y LINDMAN, J. (2013) “Code forking, governance, and sustainability in open source software.” *Technology Innovation Management Review*, 3, 1, p.
- O’MAHONY, S. y FERRARO, F. (2007) “The emergence of governance in an open source community.” *Academy of Management Journal*, 50, 5, p. 1079-1106.
- O’MAHONY, S. (2003) “Guarding the commons: how community managed software projects protect their work.” *Research Policy*, 32, 7, p. 1179-1198.
- PETRINJA, E. y SUCCI, G. Two evolution indicators for FOSS projects. IFIP International Conference on Open Source Systems, 2012. Springer, 216-232.
- POO-CAAMAÑO, G., SINGER, L., KNAUSS, E. y GERMAN, D. M. Herding cats: a case study of release management in an open collaboration ecosystem. IFIP International Conference on Open Source Systems, 2016. Springer, 147-162.
- RAYMOND, E. (1999) *The cathedral and the bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*. US: O’Reilly.
- REINGART, M. (2012). PyAfipWs: facilitando, extendiendo y liberando los Servicios Web de AFIP (Factura Electrónica y otros). *41º Jornadas Argentinas de Informática e Investigación Operativa*. La Plata.
- RIBEIRO, A., MEIRELLES, P., LAGO, N. y KON, F. Ranking source code static analysis warnings for continuous monitoring of FLOSS repositories. IFIP International Conference on Open Source Systems, 2018. Springer, 90-101.
- RIEHLE, D. y BERSCHNEIDER, S. A model of open source developer foundations. IFIP International Conference on Open Source Systems, 2012. Springer.
- RITALA, P. y HURMELINNA-LAUKKANEN, P. (2013) “Incremental and radical innovation in coepetition—The role of absorptive capacity and appropriability.” *Journal of Product Innovation Management*, 30, 1, p. 154-169.

- ROBERT, V. (2006a). "Límites y efectos de la difusión de software libre en un país en desarrollo. El caso de la Argentina.", en: J. Borello, V. Robert & G. Yoguel (eds.), *La informática en la Argentina*. Buenos Aires: Prometeo-UNGS. pp. 205-228.
- _____ (2006b) *Restricciones en la difusión de tecnologías abiertas. La difusión de software libre en la Argentina*. Buenos Aires: Tesis de Maestría, UNGS.
- ROBLES, G., AMOR, J. J., GONZALEZ-BARAHONA, J. M. y HERRAIZ, I. Evolution and growth in large libre software projects. Eighth International Workshop on Principles of Software Evolution (IWPSE'05), 2005. IEEE, 165-174.
- ROBLES, G., DUENAS, S. y GONZALEZ-BARAHONA, J. M. Corporate involvement of libre software: Study of presence in Debian code over time. IFIP International Conference on Open Source Systems, 2007. Springer, 121-132.
- ROBLES, G., REINA, L. A., GONZÁLEZ-BARAHONA, J. M. y DOMÍNGUEZ, S. D. Women in free/libre/open source software: The situation in the 2010s. IFIP International Conference on Open Source Systems, 2016. Springer, 163-173.
- ROSSI, M. A. (2006). "Decoding the free/open source software puzzle: A survey of theoretical and empirical contributions", en: Jürgen Bitzer & P. Schöder (eds.), *The Economics of open source software development*. Elsevier. pp. 15-55.
- RUBBINI, N. (2009). Medición del Uso de FLOSS en las Empresas Argentinas. Una Tarea Pendiente. *38 JAIIO - Jornadas Argentinas de Informática - SADIO*. Mar del Plata: SADIO.
- _____ (2010). Una Propuesta Para Medir el Uso de FLOSS en las Empresas en Argentina. *39 JAIIO - Jornadas Argentinas de Informática - SADIO*. Buenos Aires: SADIO.
- RULLANI, F. y HAEFLIGER, S. (2013) "The periphery on stage: The intra-organizational dynamics in online communities of creation." *Research Policy*, 42, 4, p. 941-953.
- SAMUELSON, P. (2006) "IBM's pragmatic embrace of open source." *Communications of the ACM*, 49, p. 15.
- SCACCHI, W. y JENSEN, C. (2008). Governance in open source software development projects: Towards a model for network-centric edge organizations. CALIFORNIA UNIV IRVINE INSTITUTE FOR SOFTWARE RESEARCH.

- SCHAARSCHMIDT, M., WALSH, G. y VON KORTZFLEISCH, H. F. (2015) “How do firms influence open source software communities? A framework and empirical analysis of different governance modes.” *Information and Organization*, 25, 2, p. 99-114.
- SCHMIDT, K. M. y SCHNITZER, M. (2002) “Public Subsidies for Open Source-Some Economic Policy Issues of the Software Market.” *Harv. JL & Tech.*, 16, p. 473.
- SEPPÄNEN, M., HELANDER, N. y MÄKINEN, S. (2007). “Business Models in Open Source Software Value Creation”, en: Kirk Amant & B. Still (eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. NY, US: Information Science Reference. pp. 282.
- SHAH, S. K. (2006) “Motivation, governance, and the viability of hybrid forms in open source software development.” *Management Science*, 52, 7, p. 1000-1014.
- SIRVENTE, A. (2006). Estudio Económico en Repartición Nacional Licencias vs Software Libre. *35º Jornadas Argentinas de Informática e Investigación Operativa*. Mendoza.
- SKIDMORE, D. (2007). “FLOSS legal and engineering terms and a license taxonomy”, *Handbook of research on Open Source software: Technological, economic, and social perspectives*. IGI Global. pp. 394-410.
- SOWE, S. K., ANGELIS, L., STAMELOS, I. y MANOLOPOULOS, Y. Using repository of repositories (RoRs) to study the growth of F/OSS projects: A meta-analysis research approach. IFIP International Conference on Open Source Systems, 2007. Springer, 147-160.
- SOWE, S. K. y MCNAUGHTON, M. Using Multiple Case Studies to Analyse Open Source Software Business Sustainability in Sub-Saharan Africa. IFIP International Conference on Open Source Systems, 2012. Springer, 160-177.
- STALLMAN, R. (1983) “El manifiesto GNU.” *El manifiesto de GNU*, p.
- STAM, W. y VAN WENDEL DE JOODE, R. (2007). “Analyzing Firm Participation in Open Source Communities”, en: Kirk Amant & B. Still (eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. NY, US: Information Science Reference. pp. 282.
- STARING, K. y TITLESTAD, O. Networks of Open Source Health Care Action IFIP International Conference on Open Source Systems, 2006. Springer, 123-134.

- STEINMUELLER, E. (2004). "The European software sectoral system of innovation", en: F. Malerba (ed.) *Sectoral systems of innovation. Concepts, Issues and Analyses of Six Major Sectors in Europe*, Cambridge University Press, United Kingdom. Cambridge: Cambridge University Press. pp.
- SUCCI, G., PAULSON, J. y EBERLEIN, A. Preliminary results from an empirical study on the growth of open source and commercial software products. EDSER-3 workshop, 2001. 14-15.
- TEIXEIRA, J. Open-source technologies realizing social networks: A multiple descriptive case-study. IFIP International Conference on Open Source Systems, 2012. Springer, 250-255.
- _____. Release early, release often and release on time. an empirical case study of release management. IFIP International Conference on Open Source Systems, 2017. Springer, Cham, 167-181.
- TEIXEIRA, J. y LIN, T. Collaboration in the open-source arena: the webkit case. Proceedings of the 52nd ACM conference on Computers and people research, 2014. ACM, 121-129.
- TEIXEIRA, J., MIAN, S. y HYTTI, U. Cooperation among competitors in the open-source arena: The case of OpenStack. As accepted for presentation at the 2016 International Conference on Information Systems (ICIS 2016), held in Dublin, Ireland, 2016.
- TEIXEIRA, J., ROBLES, G. y GONZÁLEZ-BARAHONA, J. M. (2015) "Lessons learned from applying social network analysis on an industrial Free/Libre/Open Source Software ecosystem." *Journal of Internet Services and Applications*, 6, 1, p. 14.
- UNU MERIT y BERLECON RESEARCH (2002) *FLOSS FINAL REPORT*. The Netherlands: European Commission.
- VAN REIJSWOUD, V. y MULO, E. (2007). "Evaluating the potential of free and open source software in the developing world", en: Kirk Amant & B. Still (eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. IGI Global. pp. 79-92.
- VAN ZWANENBERG, P., FRESSOLI, M., ARZA, V., SMITH, A. y MARIN, A. (2017). Open and Collaborative Developments. *STEPS Working Paper 98*. Brighton: STEPS Centre.
- VARIAN, H. R. y SHAPIRO, C. (2003) "Linux adoption in the public sector: An economic analysis." *Manuscript. University of California, Berkeley*.

- VEN, K., VAN NUFFEL, D. y VERELST, J. The introduction of OpenOffice.org in the Brussels Public Administration. IFIP International Conference on Open Source Systems, 2006. Springer, 123-134.
- VEN, K. y VERELST, J. The organizational adoption of open source server software by Belgian organizations. IFIP International Conference on Open Source Systems, 2006. Springer, 111-122.
- VON KROGH, G., SPAETH, S. y LAKHANI, K. R. (2003) "Community, joining, and specialization in open source software innovation: a case study." *Research policy*, 32, 7, p. 1217-1241.
- WEISS, M., MOROIU, G. y ZHAO, P. Evolution of open source communities. IFIP International Conference on Open Source Systems, 2006. Springer, 21-32.
- WEST, J. (2003) "How open is open enough?: Melding proprietary and open source platform strategies." *Research policy*, 32, 7, p. 1259-1285.
- WEST, J. y O'MAHONY, S. (2008) "The role of participation architecture in growing sponsored open source communities." *Industry and innovation*, 15, 2, p. 145-168.
- WHEELER, D. A. (2017). The free-libre/open source software (FLOSS) license slide. .
- WIGGINS, A. y CROWSTON, K. Reclassifying success and tragedy in FLOSS projects. IFIP International Conference on Open Source Systems, 2010. Springer, 294-307.
- ZANOTTI, A. (2014) "Comunidades de software libre en Argentina: motivaciones, participación, militancia." *Perspectivas de la Comunicación- ISSN 0718-4867*, 7, 2, p. 55-74.
- _____ (2015) "El software libre y su difusión en Argentina: mercado, Estado, sociedad." *Poliantea*, 11, 21, p. 147-166.
- ZUKERFELD, M. (2012). "Las tecnologías digitales y el software: tendencias históricas y situación actual", en: Lucila Dughera, Guillermina Yansen & Mariano Zukerfeld (eds.), *Gente con códigos. La heterogeneidad de los procesos productivos de software*. Buenos Aires: Universidad Maimónides.
- AKSULU, A. y WADE, M. (2010) "A comprehensive review and synthesis of open source research." *Journal of the Association for Information Systems*, 11, 11, p. 576.
- BALDWIN, C. Y. y CLARK, K. B. (2006) "The architecture of participation:

- Does code architecture mitigate free riding in the open source development model?" *Management Science*, 52, 7, p. 1116-1127.
- BERDOU, E. (2010) *Organization in open source communities: At the crossroads of the gift and market economies*. Routledge.
- CAPRA, E., FRANCALANCI, C., MERLO, F. y LAMASTRA, C. R. A survey on firms' participation in open source community projects. IFIP International Conference on Open Source Systems, 2009. Springer, 225-236.
- CENATIC (2009). Informe sobre el estado del arte del Software de Fuentes Abiertas en la empresa española. 2009.: Observatorio Nacional del Software de Fuentes Abiertas (ONSFA), Centro de Excelencia de Software de Fuentes Abiertas, Red.es / Ministerio de Industria, Energía y Turismo (MINETUR), Gobierno de España.
- _____ (2011). El Software Libre en el Sector Español de Servicios Informáticos. Informe de resultados de la Encuesta sobre el Software de Fuentes Abiertas en el sector SI español (ESFA-SI) 2010-2011. Observatorio Nacional del Software de Fuentes Abiertas (ONSFA), Centro de Excelencia de Software de Fuentes Abiertas, Red.es / Ministerio de Industria, Energía y Turismo (MINETUR), Gobierno de España.
- DAHLANDER, L. y MAGNUSSON, M. G. (2005) "Relationships between open source software companies and communities: Observations from Nordic firms." *Research Policy*, 34, 4, p. 481-493.
- DALLE, J.-M. y JULLIEN, N. (2000) "Windows vs. Linux: some explorations into the economics of Free Software." *Advances in Complex Systems*, 3, 01n04, p. 399-416.
- _____ (2003) "Libre'software: turning fads into institutions?" *Research policy*, 32, 1, p. 1-11.
- DURAND, D., VUATTOUX, J.-L. y DITSCHIED, P.-J. OSS in 2012: a long-term sustainable alternative for corporate IT. IFIP International Conference on Open Source Systems, 2012. Springer, 371-376.
- GODFREY, M. y TU, Q. Growth, evolution, and structural change in open source software. Proceedings of the 4th international workshop on principles of software evolution, 2001. ACM, 103-106.
- GRANOVETTER, M. (1992) "Economic institutions as social constructions: a framework for analysis." *Acta sociologica*, 35, 1, p. 3-11.

- HAUGE, Ø., SØRENSEN, C.-F. y CONRADI, R. Adoption of open source in the software industry. IFIP International Conference on Open Source Systems, 2008. Springer, 211-221.
- HECKER, F. (1999) "Setting up shop: The business of open-source software." *IEEE software*, 16, 1, p. Latest version revised on June 2000. Revision 0.8 available on <http://hecker.org/writings/setting-up-shop>.
- KOCH, S. Evolution of open source software systems—a large-scale investigation. Proceedings of the 1st International Conference on Open Source Systems, 2005. 148-153.
- KOCH, S. y GONZALEZ-BARAHONA, J. M. (2005) "Open Source software engineering-The state of research." *First Monday*, p.
- LERNER, J. y SCHANKERMAN, M. (2013) *The comingled code: Open source and economic development*. London, UK.
- MACCORMACK, A., RUSNAK, J. y BALDWIN, C. Y. (2006) "Exploring the structure of complex software designs: An empirical study of open source and proprietary code." *Management Science*, 52, 7, p. 1015-1030.
- MERTON, R. K. (1977) *La sociología de la ciencia: investigaciones teóricas y empíricas*. Madrid: Alianza Editorial.
- MORERO, H. A., MOTTA, J., ORTIZ, P. y VÉLEZ, J. G. (2017a). Pequeñas empresas de software libre (FLOSS) en la Argentina. *XVII Congreso Latino-Iberoamericano de Gestión Tecnológica ALTEC*. Ciudad de México.
- MORERO, H. A., MOTTA, J., VÉLEZ, J. G. y ORTIZ, P. (2017b). Taxonomías de empresas de software libre y open source. Sus limitaciones a partir de casos de pymes argentinas. *XXII Reunión Anual de la Red Pymes MERCOSUR*. Montevideo, Uruguay.
- MOWERY, D. C. (1999) "The computer software industry." *Sources of industrial leadership: Studies of seven industries*, p. 133-168.
- MUSTONEN, M. (2003) "Copyleft—the economics of Linux and other open source software." *Information Economics and Policy*, 15, 1, p. 99-121.
- RAYMOND, E. (1999) *The cathedral and the bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*. US: O'Reilly.
- ROBLES, G., AMOR, J. J., GONZALEZ-BARAHONA, J. M. y HERRAIZ, I. Evolution and growth in large libre software projects. Eighth International Workshop on Principles of Software Evolution (IWPSE'05), 2005. IEEE, 165-174.

- ROSSI, M. A. (2006). “Decoding the free/open source software puzzle: A survey of theoretical and empirical contributions”, en: Jürgen Bitzer & P. Schöder (eds.), *The Economics of open source software development*. Elsevier. pp. 15-55.
- SEPPÄNEN, M., HELANDER, N. y MÄKINEN, S. (2007). “Business Models in Open Source Software Value Creation”, en: Kirk Amant & B. Still (eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. NY, US: Information Science Reference. pp. 282.
- SKIDMORE, D. (2007). “FLOSS legal and engineering terms and a license taxonomy”, *Handbook of research on Open Source software: Technological, economic, and social perspectives*. IGI Global. pp. 394-410.
- STEINMUELLER, E. (2004). “The European software sectoral system of innovation”, en: F. Malerba (ed.) *Sectoral systems of innovation. Concepts, Issues and Analyses of Six Major Sectors in Europe*, Cambridge University Press, United Kingdom. Cambridge: Cambridge University Press. pp.
- SUCCI, G., PAULSON, J. y EBERLEIN, A. Preliminary results from an empirical study on the growth of open source and commercial software products. EDSER-3 workshop, 2001. 14-15.
- UNU MERIT y BERLECON RESEARCH (2002) *FLOSS FINAL REPORT*. The Netherlands: European Commission.
- WHEELER, D. A. (2017). The free-libre/open source software (FLOSS) license slide. .

Datos de autores y autoras

Carina Borrastero es Doctora en Ciencias Sociales por la Universidad de Buenos Aires y Magister en Ciencia, Tecnología y Sociedad por la Universidad Nacional de Quilmes (Argentina). Es Investigadora Asistente del Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET) de Argentina, y Profesora Investigadora en la Facultad de Ciencias Económicas de la Universidad Nacional de Córdoba con asignación en la cátedra de Economía Industrial y el Centro de Investigaciones en Ciencias Económicas (CICE) de la misma institución. Sus intereses de investigación se concentran en temas Sociología Económica y Economía del Desarrollo. E-mail: cariborrastero@gmail.com

Denise Gutiérrez Montecino es ayudante del Grupo de Economía de la Innovación y el Desarrollo (FCE-UNC) y estudiante avanzada de la Licenciatura en Economía (FCE-UNC). E-mail: gutierrez.denise.5@gmail.com

Ignacio Juncos es Licenciado en Economía por la Universidad Nacional de Córdoba (UNC). Es Profesor Ayudante B en la cátedra de Estadística II en la Facultad de Ciencias Económicas (UNC). Es miembro de la Sociedad de Economía Crítica (SEC) y actualmente trabaja como analista económico en la Dirección General de Estadística y Censos de la Provincia de Córdoba. Sus intereses de investigación se concentran en los estudios de financiarización y su impacto en el sector productivo, especialmente su manifestación en economías periféricas. Email: ignacijuncos94@gmail.com

Florencia Manzo es ayudante del Grupo de Economía de la Innovación y el Desarrollo (FCE-UNC) y estudiante avanzada de la Licenciatura en Economía (FCE-UNC). E-mail: florenciamanzo1@gmail.com

Hernán Alejandro Morero es Doctor en Ciencias Económicas por la Universidad Nacional de Córdoba (UNC). Es Investigador Adjunto del CONICET en el Centro de Investigaciones y Estudios sobre Cultura y Sociedad

(CIECS, CONICET y UNC) y Profesor Adjunto en la cátedra de Economía Industrial en la Facultad de Ciencias Económicas (UNC). Es miembro del Comité Científico de LALICS y actualmente es editor de la Revista Pymes, Innovación y Desarrollo de la Red Pymes Mercosur. Sus intereses de investigación se concentran en las áreas de economía de la innovación y procesos de aprendizaje a nivel de la firma en diversos sectores industriales y de servicios, y en políticas industriales y estrategias tecnológicas en economías periféricas. E-mail: hernanmorero@eco.uncor.edu

Jorge Motta es Doctor por la Universidad Autónoma de Barcelona y Magíster en Economía por la Universidad Estatal de Campinas (UNICAMP). Es Director de la Carrera de la Licenciatura en Economía y Profesor Titular en la Facultad de Ciencias Económicas de la UNC. Sus investigaciones se concentran en las áreas de economía industrial y desarrollo económico, con especial énfasis en estudios sectoriales, pequeñas y medianas empresas y economía de la innovación. E-mail: jorgejmotta@gmail.com

Juan Gabriel Vélez es Licenciado en Economía por la Universidad Nacional de Córdoba (UNC). Becario doctoral de la Secretaría de Ciencia y Tecnología de la UNC, en el Instituto de Economía y Finanzas de la Facultad de Ciencias Económicas. Docente investigador en el Departamento de Sociedad, Estado y Gobierno de la Universidad Nacional de Rafaela. E-mail: juangavelezz@gmail.com.

Agustín Zanotti es Doctor en Estudios Sociales de América Latina por la Universidad Nacional de Córdoba. Docente e investigador de la Facultad de Ciencias Sociales, en el Centro de Investigaciones y Estudios sobre Cultura y Sociedad, Universidad Nacional de Córdoba y CONICET. Docente e investigador del Instituto AP Ciencias Sociales, Universidad Nacional de Villa María. E-mail: agustinzanotti@gmail.com

Este libro procura constituir un aporte para el abordaje académico del Software Libre y de Código Abierto o Free/Libre Open Source Software (FLOSS) desde una perspectiva económico-productiva, actualmente ausente en la literatura en castellano. Para ello, en primer lugar se reúnen los aspectos conceptuales elementales y un estado del arte para un abordaje de este tipo junto a una serie de estadísticas sobre la importancia cuantitativa de la actividad del FLOSS en el sector de software de la Argentina y el mundo. En segundo lugar, aportamos nociones teóricas que permiten considerar la organización de la producción de software en comunidades y el rol de las Fundaciones FLOSS, que ilustramos a partir de cuatro casos de comunidades de desarrollo: Fedora, Gnome, Python y LibreOffice. Por último, estudiamos cómo las empresas de software articulan sus estrategias de competencia alrededor de estas comunidades y sus proyectos. Allí vemos que tanto grandes corporaciones de software, como pymes, mantienen esquemas de vinculación con la comunidad FLOSS y participan en numerosísimos proyectos colectivos de desarrollo de software, cooperando y compitiendo entre sí. A lo largo del libro puede apreciarse la manera que el FLOSS ha atravesado y alterado a la industria del software por completo. Ello torna necesario desarrollar un adecuado instrumental analítico para su consideración en la investigación académica en economía.