

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ciencias Políticas y Sociales

Software Libre y abierto:
comunidades y redes de producción digital de
bienes comunes

por Tania E. Turner Sen

Tesis propuesta como cumplimiento
de los requisitos para la Maestría en

Estudios Políticos y Sociales

Dra. Norma Patricia Maldonado Reynoso

Directora de Tesis

Agosto 2012



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Tesis propuesta como cumplimiento de los requisitos para la Maestría en Estudios Políticos y Sociales, cuyo desarrollo fue financiado por el Consejo Nacional de Ciencia y Tecnología (Conacyt)

Por
Tania Elizabeth Turner Sen

Asesora: Dra. Norma Patricia Maldonado Reynoso
Dra. en Ciencias Políticas y Sociales

2012
Universidad Nacional Autónoma de México
Ciudad de México, México

*Para Inés e Isa
...otro mundo es posible
uno 'más alrevés volteado',
digo...*

Agradecimientos

Esta investigación es resultado del reconocimiento y apoyo que he recibido de todos los que han caminado junto a mí a lo largo de este proceso. Agradezco especialmente...

al Dr. Andrade, por apoyar mi ingreso y ayudarme a defender mi propuesta de investigación,

a la Dra. Maldonado, por su compromiso y apoyo a lo largo de todo el proceso,

a los Dres. Alejandro Méndez, Raúl Trejo, Enrique Quibrera y Gabriel Arango, cuya generosidad intelectual se ha convertido en mi ejemplo,

al Consejo Nacional de Ciencia y Tecnología (Conacyt), por el apoyo económico brindado,

a mis profesores y profesoras, quienes me invitaron a nuevos horizontes teóricos y perspectivas de análisis,

a mi familia, Francisco, Isabel e Inés, sin quienes nada de todo esto valdría la pena.

*Tania Turner.
Ciudad de México, agosto 2012*

Índice

<i>Introducción</i>	9
<i>Capítulo 1 El software. Trascendencia e impacto en el siglo XXI</i>	17
1.1 El contexto digital del siglo XXI: producción y uso de software.	19
1.2 Software privativo, abierto y libre.....	32
1.2.1 Definición de software.....	32
1.2.2 Diferencias entre software privativo, de fuente abierta y libre	35
1.2.3 Estudios sobre software libre.....	40
1.3 La emergencia y desarrollo del software.....	45
1.3.1 Antecedentes de la informática y la computación.....	46
1.3.2 El código fuente. Álgebra, lógica y lenguajes de programación	49
1.3.3 Máquinas, mecanismos y herramientas.....	50
1.3.4 De las super-computadoras al ordenador personal.....	52
1.3.5 Unix y el movimiento de software libre y abierto.....	56
1.3.6 Arpanet, Usenet, Internet. El espacio de la colaboración.....	68
1.4 Software libre y abierto en la actualidad (ejemplos y circulación en diferentes países).....	72
<i>Capítulo 2</i>	
<i>Tradicional y nuevos bienes comunes</i>	85
2.1 Globalización y nuevas tecnologías.....	86
2.2 Diferencia entre bienes públicos, privados y comunes.....	94
2.3 Breve historia de los bienes comunes.....	99
2.4 Nuevos bienes comunes.....	103
2.4.1 Nuevos bienes comunes de la información y el conocimiento	105
2.4.2 Aspectos económicos, políticos y culturales de los nuevos bienes comunes del conocimiento y la información.....	108
2.5 Software libre como bien común.....	123
<i>Capítulo 3 Dinámicas de interacción en redes y comunidades de producción de FLOSS</i>	131

3.1 Teoría del Esquema y el Conexionismo: motivaciones.....	136
3.2 Interaccionismo estructural y Análisis de Redes Sociales (ARS)	139
3.3 Organización en la producción de software libre y abierto	145
3.3.1 La comunidad hacker.....	148
3.3.2 Interacción entre desarrolladores y usuarios. La colaboración en línea.....	157
3.3.3 Proyectos, Programas y Repositorios/Forjas.....	162
<i>Capítulo 4 Las redes de producción de software libre y abierto.</i>	
<i>Interacción, motivaciones y perspectivas.....</i>	<i>171</i>
4.1 Guía metodológica.....	173
4.1.1 Software libre a la luz del análisis de redes sociales (ARS).	174
4.2 Dinámicas y procesos en las redes y comunidades de producción digital de software libre. Análisis comparativo de seis repositorios en internet.....	181
4.2.1 Características de los repositorios.....	185
4.2.2 Las redes de colaboración y sus dinámicas de intercambio en cada repositorio.....	198
4.3 Las comunidades y redes de producción digital de software libre y abierto desde la perspectiva de sus actores.....	213
4.3.1 Motivaciones de los actores.....	213
4.3.2 Respuestas de las comunidades y redes de FLOSS al mercado digital, en tanto defensa de la producción colectiva y de alianzas estratégicas.....	225
4.4 El futuro del software libre y abierto.....	232
<i>Conclusiones.....</i>	<i>237</i>

Introducción

Hablar de tecnología, de su desarrollo, formas de producción y su utilidad es hablar de un modo de organización social. La tecnología implica el uso de herramientas para un objetivo específico, pero también refleja el sentido y el valor que una sociedad determinada le otorga a la creatividad, ingenio y trabajo de sus miembros.

En una sociedad como la capitalista, las formas de producción dominantes están orientadas a la acumulación de capital y tienen como principio la propiedad privada. Este principio organizativo regula la producción y el consumo de mercancías, incluida la tecnología.

En el contexto social actual, las tecnologías de información y comunicación (TIC) se presentan como productos representativos de procesos de cambio social y como herramientas para nuevas transformaciones. Desde los Estudios Políticos y Sociales, la reflexión sobre la forma en la que nos relacionamos con estas tecnologías emergentes se hace indispensable; sobre todo porque la relación entre la sociedad y la tecnología es una relación dialéctica que implica procesos sociales y nuevas configuraciones de los mecanismos de control y dominio. En esta relación sociedad-tecnología se generan

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

dependencias y autonomías, así como soluciones y nuevos problemas. Desde la perspectiva que se toma en este estudio, la tecnología es resultado de procesos de construcción social; sin embargo, una vez que emerge y se utiliza, sí puede incidir en el actuar de la sociedad, creando nuevas configuraciones sociales y nuevas necesidades.

Los procesos sociales que emergen del desarrollo y uso de las TIC y sus incidencias en el cambio técnico y cultural son necesarios y susceptibles de ser abordados en investigaciones diversas. La observación se plantea con respecto al establecimiento de nuevas formas de relacionarse socialmente, a partir de la comunicación inmediata, el intercambio constante, el exceso y uso de la información a gran escala. Estas herramientas de comunicación han permitido la generación de formas colectivas de colaboración para producir bienes y servicios, invitando a transformar de diversas maneras la división internacional del trabajo. En este estudio, se aborda una forma de producción colectiva que, gracias a la emergencia de las TIC, ha podido expandirse y fortalecerse en el mercado global de alta tecnología: la producción de *software* libre y abierto.

Las TIC funcionan sobre plataformas digitales, cuyo acceso está determinado por el tipo de programas informáticos que utilizamos y el conocimiento que tenemos de los mismos.

El potencial de estas plataformas digitales para la educación, el avance del conocimiento y la vinculación social quedará determinado por el significado y valor que les otorguemos. Esto es así porque las tecnologías no son neutrales, emergen como

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

estrategias y mecanismos que se insertan en intereses políticos, económicos y sociales de largo alcance. Por ejemplo, si es más fuerte la lógica del mercado, el acceso a las aplicaciones de las plataformas digitales podría restringirse a través de programas de código cerrado, cuyo costo se podría imponer a través de mecanismos de oferta y demanda. Por otro lado, si la lógica del bienestar social impera, el acceso a éstas tendría que ser universal y, para garantizarlo, el tipo de programas informáticos tendría que permitir el aprendizaje, conocimiento, modificación y mejora de su propio funcionamiento; es decir, las aplicaciones tendrían que ser *software* libre y abierto. Las comunidades y redes que lo producen, modifican y distribuyen son el objeto central de análisis de esta tesis.

Las comunidades y redes de producción de *software* libre y abierto, aunque no se puede decir que rechazan el sistema hegemónico, sí proponen una forma de organización de la producción alternativa a la producción mercantil privativa. El producto del trabajo colectivo de estas comunidades y redes se basa en la libertad de uso, modificación, mejora y distribución. Las reglas de intercambio no mercantil se construyen por los miembros dentro de las propias redes de *software* libre y abierto.

En la producción de *software* libre y abierto, los beneficios del desarrollo tecnológico, especialmente el avance del conocimiento y de las habilidades técnicas para productores y usuarios, pueden distribuirse de manera más solidaria, justa y equitativa. De esta manera, se protege el conocimiento acumulado como una forma de intercambio que no puede ser mercantilizado y, en su caso, se reparten las ganancias

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

económicas en una base más amplia de productores y usuarios, evitando la concentración de capital y de conocimiento técnico.

Existen diversos estudios que abordan la producción de *software* libre y abierto en relación a las implicaciones de su producción para la innovación, la explicación de las motivaciones de los actores desde la perspectiva teórica de la acción racional, la organización institucional de las comunidades, entre otros. Esta tesis retoma de muchos de estos estudios la categorización del *software* libre y abierto como un nuevo bien común, pero aborda el proceso de producción más allá de las reglas de las comunidades, describiendo la interacción en redes amplias de desarrolladores y proyectos.

El presente estudio también aborda el análisis de las motivaciones de los actores para la colaboración, pero busca proponer una forma de abordaje de la investigación que explique y describa las distintas formas de interacción en las redes de producción, en lugar de sólo explicar por qué los individuos cooperan. En el análisis no se plantea como problemática la acción colectiva; en este sentido, se aparta de los estudios desde la Teoría de la Acción Racional.

Los argumentos esbozados en este trabajo se enmarcan dentro de la perspectiva teórica de la construcción social de la tecnología, por lo que también se planteó como importante la narración histórica sobre el desarrollo de la computación y la informática. El conocimiento de los antecedentes de las TIC permite entender las formas de interacción colaborativa que han permanecido en la producción del *software* libre y abierto. Finalmente, se buscó describir la situación actual de este tipo de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

programas y aplicaciones y las perspectivas a futuro sobre la sustentabilidad de estas comunidades y redes.

Las preguntas rectoras de esta investigación son: ¿Cómo y por qué funcionan las redes de producción, difusión e intercambio de *software* libre, en tanto interacciones entre usuarios y desarrolladores? y ¿Cómo se inserta la producción de *software* libre y abierto en la dinámica del capitalismo del siglo XXI?

Para dar respuesta a estas inquietudes, la presente tesis se divide en cuatro capítulos más las conclusiones finales, a lo largo de los cuales se pretende describir y explicar el funcionamiento de las redes de *software* libre y abierto y sus dinámicas de producción e intercambio.

En el primer capítulo, denominado “El *software*. Trascendencia e impacto en el siglo XXI”, se explican las diferencias entre *software* privativo, abierto y libre, a través de la definición conceptual, pero también a través de la historia de su surgimiento. Se describen los acontecimientos históricos que dieron forma a este tipo de tecnologías, en tanto modos de producir e intercambiar conocimientos y habilidades técnicas. En este apartado, se presenta la revisión de literatura y se describe la situación actual del *software* libre y abierto.

El capítulo segundo, titulado “Tradicionales y nuevos bienes comunes”, aborda el contexto económico de la producción del *software* libre y abierto. En este apartado, se explican las diferencias entre bienes privados, públicos y comunes para comprender y fundamentar la inserción del *software* libre y

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

abierto en la categoría de nuevo bien común del conocimiento. La intención del análisis en esta tesis es que sea complementario a los estudios que se han hecho sobre el tema desde el neo-institucionalismo, aunque la perspectiva teórica no sea la misma. Por lo anterior, los trabajos sobre *software* libre como bien común de Charles M. Schweik, se exponen como complemento de la descripción del fenómeno en un sub-apartado.

En el tercer capítulo, “Dinámicas de interacción en redes y comunidades de producción de FLOSS”, se describe la organización de las comunidades y redes de producción y uso de *software* libre y abierto, a través de las infraestructuras virtuales, llamadas repositorios. En este apartado, se describen la Teoría del Esquema y el Conexionismo, así como el Interaccionismo Estructural y el Análisis de Redes Sociales, utilizadas para el análisis de la colaboración de la tesis. Asimismo se exponen y describen los perfiles de, y la interacción entre, desarrolladores y usuarios de *software* libre y abierto.

Finalmente, el cuarto capítulo, llamado “Las redes de producción de *software* libre y abierto. Interacción, motivaciones y perspectivas”, expone la guía metodológica para el análisis de los datos recabados. Los resultados se presentan en diferentes subapartados que abordarán las motivaciones de los actores, las redes de colaboración y el futuro del *software* libre y abierto. El análisis se expone a través de gráficas comparativas con base en estadística descriptiva, gráficos de redes y esquemas, resultado del manejo de datos obtenidos del sitio Flossmole.org y de la recopilación de documentos y realización de entrevistas a personajes clave del movimiento de *software* libre y abierto.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Las conclusiones que se presentan al final de este trabajo pretenden ofrecer una aproximación clara de la dinámica de las redes de producción de *software* libre y abierto en la infraestructura virtual, tanto de la interacción y forma de cooperación, como de los esquemas que subyacen a la acción colaborativa desde los actores. El análisis realizado ha permitido generar propuestas de abordaje para investigaciones futuras, tanto para extender el análisis de las redes en los repositorios, como para profundizar en la comprensión de las diversas formas de colaborar, debido a las diferencias en los esquemas cognitivos de los actores involucrados.

Además de las razones mencionadas al inicio de esta introducción, el abordaje del tema en esta tesis de maestría responde también a que creo que las formas colectivas de producción y uso del *software* libre y abierto son susceptibles de replicación en otros ámbitos para beneficio social. La comprensión y difusión de este tipo de acción colectiva ofrece alternativas para la acción social que superen y dejen atrás la tendencia al individualismo y la mercantilización del mundo.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Capítulo 1 El *software*. Trascendencia e impacto en el siglo XXI

En el siglo XXI, la dinámica entre sociedad y tecnología ha desarrollado de manera acelerada nuevas formas de relacionarnos a través de la información y la comunicación. La tecnología —que se entiende en este estudio no sólo en tanto los objetos físicos derivados de los avances científico-técnicos, si no también, y desde, las actividades humanas, técnicas e intelectuales, que dan forma a estos objetos— se construye socialmente, su forma es una expresión del modo de organización social.

Lo tecnológico es social [lo que implica no sólo lo sociológico, sino también] lo político, económico, psicológico —y, de hecho, histórico. (Bijker, Law, & Latour, 1992)

Las llamadas nuevas tecnologías de la información y la comunicación (TIC)¹ reflejan nuevas formas de mirar y

¹ Se entenderá por este término el “conjunto de procesos y productos derivados de las nuevas herramientas (*hardware* y *software*), soporte de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información.” (González Soto et al., 1996)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

aprehender el mundo, en las que la instantaneidad y el acceso se convierten en posibles herramientas de lo cotidiano, instrumentos para las relaciones sociales. Sin embargo, es importante hacer hincapié en que las tecnologías de hoy en día, son de determinadas formas, cumplen determinadas funciones y se utilizan de cierta manera, pero podrían ser, funcionar y utilizarse de modos distintos. Como sugieren John Law y Wiebe E. Bijker (1992):

Las tecnologías no emergen bajo el ímpetus de alguna lógica científica o tecnológica interna. No están poseídas de un momento inherente. Si emergen o cambian, es porque han sido presionadas a adquirir esa forma. La pregunta deviene: ¿por qué han tomado la forma que tienen actualmente? (p.3)

Armand Mattelart lo expresa en términos similares en su *Historia de la Sociedad de la Información* (2002):

En cada civilización, en efecto, cada área histórico-geográfica construye su modo de apropiación e integración de las técnicas, que da origen a configuraciones comunicacionales múltiples con sus respectivos niveles, ya sean económico, social, técnico o mental, y sus distintas escalas, local, regional, nacional o transnacional

Como tecnología de información y comunicación, el desarrollo y uso de *software* ha hecho posible formas de intercambio social y cultural y ha desarrollado el conocimiento

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

técnico y científico, facilitando los procesos de interacción virtual y real hasta cierto punto. La palabra '*software*' se refiere a aplicaciones y programas digitales que, través de órdenes codificadas, le dicen a una máquina específica lo que debe hacer; es decir, son las instrucciones empaquetadas que hacen funcionar de determinada manera a una computadora o un celular. El *software* se puede clasificar en libre, abierto o privativo, según muestre o no esas órdenes al usuario.

En este capítulo, definiré y explicaré los conceptos de *software* privativo, abierto y libre, describiendo las distintas maneras en las que se relacionan con sus desarrolladores y usuarios. Describiré los acontecimientos históricos que dieron forma a este tipo de tecnologías, en tanto modos de producir e intercambiar conocimientos y habilidades técnicas. Profundizaré en la narración y análisis de los sucesos históricos que llevaron a la distinción entre estos tres tipos de *software*, y cómo la diferenciación se manifiesta en la actualidad.

1.1 El contexto digital del siglo XXI: producción y uso de *software*

La tecnología que utilizamos es la expresión del modo en que vivimos, de la racionalidad de nuestras sociedades y su devenir histórico. Las tecnologías nuevas no emergen espontáneamente, son resultado de la investigación científica y de la evolución y el desarrollo de tecnologías existentes. Se puede decir que la sociedad da forma a la tecnología, pero el progreso técnico, una vez iniciado, se desarrolla casi con una lógica propia que incide

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

en la sociedad que le dio vida. Como dice Castells en el Prólogo del primer tomo de “La Era de la Información” (2005):

[...] la tecnología no determina la sociedad. Tampoco la sociedad dicta el curso del cambio tecnológico, ya que muchos factores, incluidos la invención e iniciativas personales, intervienen en el proceso del descubrimiento científico, la innovación tecnológica y las aplicaciones sociales, de modo que el resultado final depende de un complejo modelo de interacción [...] No obstante, tan pronto como se difundieron las nuevas tecnologías de la información y se las apropiaron diferentes países, distintas culturas, diversas organizaciones y metas heterogéneas, explotaron en toda clase de aplicaciones y usos, que retroalimentaron la innovación tecnológica, acelerando la velocidad y ampliando el alcance del cambio tecnológico, y diversificando sus fuentes (p.4)

De forma similar, Jacques Ellul (1964) menciona:

En un sentido, la técnica ciertamente progresa a través de pequeñas mejoras, que son resultado de esfuerzos humanos comunes, y que se acumulan indefinidamente hasta formar una masa de nuevas condiciones que permiten un paso decisivo hacia adelante. Pero es igualmente cierto que la técnica reduce agudamente el papel de la invención humana. (Ellul, 1964: 85-86)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Lo que dice Ellul en el libro “The Technological Society” (1964) es que, en efecto, la sociedad genera tecnología y empuja el desarrollo técnico, pero esa misma sociedad queda determinada por la tecnología que utiliza, al quedar inserta dentro de un paradigma tecnológico. La sociedad históricamente genera tecnología para cubrir sus necesidades, pero esta tecnología crea nuevas necesidades en la sociedad. La generación de tecnología y la forma que se le da a ésta responden a contextos políticos. Siguiendo a Langdon Winner:

Las sociedades crean nuevas instituciones, nuevos patrones de comportamiento, nuevas sensibilidades, nuevos contextos para el ejercicio del poder, siguiendo procesos de instrumentación superados paso a paso (Winner, 1987: 187).

Es decir, las tecnologías emergen creando nuevas necesidades, pero éstas nacen para y por un marco de poder existente; bajo arreglos técnicos y de diseño que preceden el uso de los propios objetos. Citando a Winner nuevamente:

Las cosas que llamamos 'tecnologías' son formas de construir orden en nuestro mundo. Muchos dispositivos y sistemas técnicos importantes en la vida diaria contienen posibilidades para ordenar de diversas maneras la actividad humana. Concientemente o no, con deliberación o sin advertirlo, las sociedades escogen estructuras para las tecnologías que influyen en cómo la gente trabajará, se comunicará, viajará, consumirá, a lo

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

largo de un largo período de tiempo. En el proceso en el que se toman las decisiones estructurantes, diferentes personas están situadas de modos diferentes y poseen grados desiguales de poder, así como niveles desiguales de conciencia. (MacKenzie & Wajcman, 1996: 30)

Como ejemplo, se puede mencionar la creación del automóvil. La sociedad tenía necesidad de transportarse de un lugar a otro, lo que dio paso a que se generara la idea de un medio de transporte más rápido y eficiente que las carretas de caballos. La invención del automóvil llevó a la necesidad social de reglamentar de un modo específico su uso, generando aparatos eléctricos como los semáforos; para protección, como los cinturones de seguridad, etcétera. Con esto, se estableció también una influencia, o modo de usar, de la tecnología emergente en la sociedad que la generó. Como lo expone Latour (1992: 247):

La Belleza de los artefactos es que toman en sí mismos los deseos y necesidades contradictorias de los humanos y no-humanos. La función de mi cinturón de seguridad supone amarrarme firmemente en caso de un accidente; por lo tanto, imponerme que respete el consejo 'NO CHOQUES CONTRA EL PARABRISAS', que en sí mismo es la traducción de la meta inalcanzable 'NO MANEJES DEMASIADO RÁPIDO' [...]

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Argumentos similares se pueden encontrar en Winner (1987: 22):

Si la experiencia de la sociedad moderna nos muestra algo, esto es que las tecnologías no son simples medios para las actividades humanas, sino también poderosas fuerzas que actúan para dar nueva forma a dicha actividad y a su significado.

O Iván Illich (2006: 336):

Cuando la velocidad de [los] vehículos rebasa cierto margen, la gente se convierte en prisionera del vehículo que la lleva cada día de la casa al trabajo [...] Mientras que unos pocos viajan en alfombra mágica entre puntos distantes, los otros, que son la mayoría, se tienen que desplazar con más y más rapidez por los mismos trayectos monótonos y deben consagrar cada vez más tiempo a estos desplazamientos.

Lo que describen los autores son procesos de construcción social de tecnología, en los que la invención responde a necesidades sociales específicas. Estas necesidades, a su vez, se ven transformadas por los propios productos generados por la ciencia. Sin embargo, no se debe perder de vista que la tecnología en desarrollo queda inserta en un modo de producción determinado y, por tanto, en formas de organización humanas pre-establecidas, que responden a formas de autoridad

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

y de organización social. Como dice Langdon Winner (1987: 127)

Aquéllos mejor situados para tomar ventaja del poder de una nueva tecnología son, a menudo, aquéllos previamente bien situados a fuerza de bienestar, posición social y posición institucional.

Los seres humanos no sólo desarrollan y transforman el mundo de la naturaleza, sino también sus propias capacidades y su propia naturaleza al transformar su mundo. El hombre produce en sociedad objetos materiales, pero también ideas, objetos artísticos, lenguaje. La actividad vital del hombre en este sentido amplio es fundamentalmente la continua recreación de su entorno material. Marx expone esta idea, en tanto un esfuerzo inserto en un modo de producción específico:

Lo que son [los hombres] coincide, por consiguiente, con su producción, tanto con lo que producen como en el modo cómo producen. Lo que los individuos son depende, por tanto, de las condiciones materiales de su producción. (Marx, 2000)

El modo de producción determina limitantes a la producción misma, en tanto condiciones materiales y formas de organización social, valores y representaciones del mundo. Un modo de producción contiene marcos de pensamiento determinados, que pueden cambiar, pero lo hacen lentamente y en circunstancias específicas. Con esto me refiero a la noción de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

paradigma; para este trabajo, específicamente, paradigma tecnológico.

La idea del paradigma tecnológico está basada en la propuesta teórica sobre los paradigmas científicos de Thomas S. Kuhn, que tiene dos sentidos relacionados entre sí. Por un lado, representa un ejemplo de una forma de solucionar científicamente un problema de manera exitosa y que, por tanto, se convierte en la forma de hacerlo para el futuro. Por el otro, paradigma es un concepto que expresa “toda una constelación de creencias, valores, técnicas, etcétera, compartida por los miembros de una comunidad dada.” (Marx, 2000)

El siglo XXI, con la herencia de los siglos anteriores, se enmarca en un paradigma tecnológico que sustenta, y es sustentado por, la forma en que vivimos y pensamos. Actualmente, es prácticamente imposible pensar nuestra vida sin luz eléctrica o agua entubada. En el momento en que nos encontramos ahora, el paradigma tecnológico enmarca, entre otras cosas, las tecnologías y medios masivos de comunicación e información. Se busca mayor comunicación en menos tiempo, superar las distancias a partir de la cercanía virtual, artefactos cada vez más poderosos y pequeños para mantener la conexión de personas, empresas y organizaciones a nivel mundial, así como las transacciones mercantiles y las operaciones financieras que engloban prácticamente ya a todos los países.

Los organismos nacionales e internacionales ven la adopción de nuevas tecnologías como un indicador importante del desarrollo de los países. El índice de desarrollo de tecnologías de la información y comunicación (TIC), desarrollado por la Unión

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Internacional de Telecomunicaciones (ITU, por sus siglas en inglés) en 2008, mide, a partir de la combinación de 11 indicadores, el nivel y evolución de las TIC a través del tiempo, el progreso en el desarrollo de éstas en países desarrollados y en desarrollo, la brecha digital, y el potencial de desarrollo de estas tecnologías. El objetivo final es medir las capacidades globales y el desarrollo de la sociedad de la información, tomando en cuenta 1) la infraestructura y el acceso, 2) la intensidad de uso y 3) las capacidades y habilidades de adopción de las TIC.

El acceso mide líneas telefónicas y suscripciones a líneas de telefonía celular por cada 100 habitantes, ancho de banda internacional por usuario de Internet, porcentaje de viviendas con computadora y porcentaje de viviendas con acceso a Internet. La intensidad de uso toma en cuenta el porcentaje de individuos que usan Internet, suscriptores de Internet de banda ancha y suscripciones activas de telefonía celular de banda ancha por cada 100 habitantes. Las capacidades y habilidades de adopción de las TIC se miden por las tasas de alfabetización y de inscripción bruta a educación media y superior.

Este último punto, relativo a las capacidades y habilidades de adopción de estas tecnologías, es particularmente importante para determinar las formas, impactos y posibilidades reales de la adopción de las tecnologías de información y comunicación, pues, citando a Winner nuevamente (p. 127):

La idea [de que el acceso a la información deviene en conocimiento y éste en poder, mejorando la democracia e igualando el poder social] es totalmente equivocada. Confunde el total

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

abastecimiento de la información con una habilidad disciplinada para obtener conocimiento y actuar de manera efectiva.

Es decir, es necesario aprender a traducir la información, tener la habilidad de procesarla. Aunque las variables medidas en el informe citado no parecen suficientes para determinar la habilidad para realmente lograr el acceso, implican un reconocimiento de que no sólo la información y el acceso a ésta es necesario para hablar de adopción eficiente de las tecnologías.

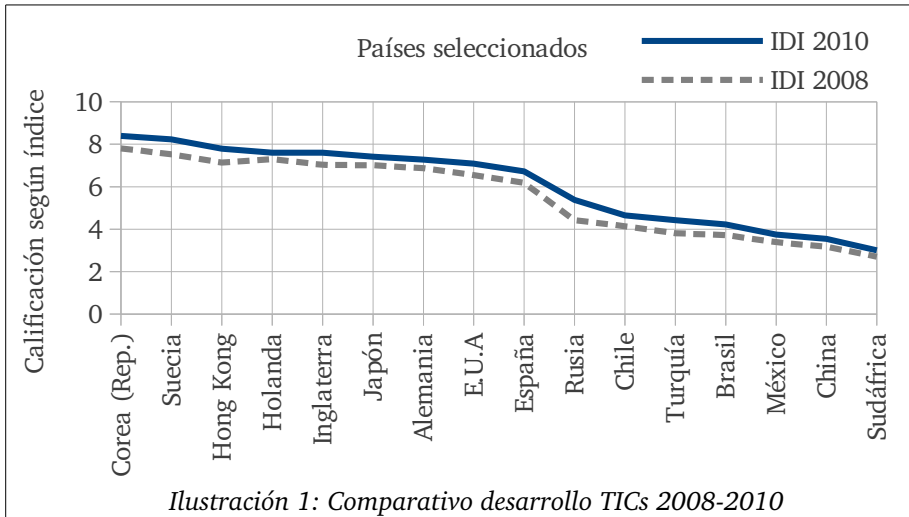
Cada uno de estos indicadores (acceso, uso y capacidades) tiene un valor en porcentaje (40, 40 y 20, respectivamente), y las variables de cada indicador son medidas según un valor de referencia. Los resultados obtenidos se transforman a una escala del 1 al 10 para obtener valores comparativos.²

El último informe de la Unión Internacional de Comunicaciones, publicado en 2011, presentó un análisis comparativo entre 2008 y 2010. Globalmente, el desarrollo en estos dos años de las tecnologías de la información y comunicación en el sentido descrito subió en promedio en .46, de 3.62 a 4.08, siendo el subíndice de uso el que más se incrementó (.62), y el de capacidades el que menos (.09). Prácticamente todos los países se han insertado en mayor o menor grado en el desarrollo y uso de las TIC. Sin embargo, las diferencias entre países son enormes, pues se muestra que el país con la menor calificación (Chad) en el desarrollo de TIC

² Para mayor información sobre la construcción del índice ver (ITU, 2011)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

tuvo apenas un valor en el índice de .8, mientras que el mejor calificado, la República de Corea, logró una calificación de 8.4 en la escala.



Fuente: ITU, 2011. *Measuring the Information Society*, p.13
(<http://www.itu.int/ITU-D/ict/publications/idi/2011/>)

De acuerdo al gráfico mostrado, México se ubica en el lugar número 75 de 152 países, con una calificación en 2010 de 3.75. Presento a continuación, las calificaciones obtenidas en los subíndices, que expresan el desarrollo de cada uno de los apartados que se toman en cuenta para calificar el avance en TIC por país:

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

INDICADOR	2010		2008	
	LUGAR	CALIFICACIÓN	LUGAR	CALIFICACIÓN
Acceso	81	3.94 [*]	78	3.50
Uso	69	1.86 [*]	68	1.15
Habilidades	71	7.14 [*]	74	6.99

*Cuadro 1: Comparativo de indicadores para México 2008-2010.
Elaboración propia en base al documento de ITU, 2011. Measuring the Information Society*

En el comparativo del reporte de la ITU, sólo del continente americano, México se encuentra en el lugar 12. La región en su conjunto, siguiendo el reporte mencionado bajó su puntuación con respecto al 2008, convirtiéndose en “la región menos dinámica del período con el mayor número de países con un cambio negativo en su calificación” (p.48). En el subíndice de uso, México logró un incremento en las suscripciones de banda ancha móvil e internautas. La penetración de la banda ancha móvil en el país creció del 2% al 8%, y el número de usuarios de Internet del 22 al 31%. Lo que demuestran estos datos es que el desarrollo de las TIC a nivel mundial es desigual, pero que sigue una evolución hacia una mayor adopción y uso de este tipo de tecnologías. Algunos países aumentaron la velocidad de acceso,

* La puntuación más alta en cuanto a acceso la obtuvo Hong Kong, China y fue de 9.06

* La puntuación más alta en cuanto a uso la obtuvo la República de Corea y fue de 7.85

* La puntuación más alta en cuanto a habilidades la obtuvo Finlandia y fue de 9.89

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

otros de uso –como es el caso de México-, y otros, aún pocos, de capacidades y habilidades de los usuarios para la adopción efectiva, en cuanto a conocimiento y destreza.

Cabe mencionar que el índice se construye cuantitativamente, lo que implica que puede dar cuenta de suscripciones, acceso, conexiones a Internet y comunicación celular, pero no califica la forma en que se aprovecha o no ese acceso. Más allá de tener habilidades para hacer un uso de la tecnología, también es importante determinar si existen posibilidades de explotar estas habilidades a pesar de las diferencias económicas.

El desarrollo de las TIC a nivel global es desigual; sin embargo, en lo general, se están incrementando principalmente el acceso y el uso en todos los países. Este incremento está transformando la forma de vida de todas las sociedades del globo. Estas tecnologías, devenidas de las necesidades sociales de comunicación e información constante, cada vez son más utilizadas por más personas en el mundo. Esto implica que los flujos de comunicación ahora son más dinámicos y, con ello, la necesidad de mantenerse informado y conectado emerge en el imaginario social. Hoy por hoy, las nuevas tecnologías están cada vez más presentes, se insertan en nuestra cotidianeidad y en nuestros esquemas cognitivos como parte fundamental para la socialización y, de manera incremental, para el aprendizaje y los procesos de trabajo. No debe olvidarse que esta tecnología ha emanado de una construcción social, de un devenir histórico, pues, como dice Ivan Illich (2006: 141)

Un mito moderno desea hacernos creer que el sentido de impotencia con el que vive la mayoría de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

los hombres de hoy es consecuencia de la tecnología que sólo puede crear sistemas enormes. Pero no es la tecnología la que hace enormes sistemas, instrumentos inmensamente poderosos, canales de comunicación unidireccionales. Todo lo contrario: si la tecnología estuviera adecuadamente controlada, podría capacitar a cada hombre para entender mejor su medio ambiente, moldearlo con sus propias manos, y permitirle la intercomunicación total a un grado nunca antes alcanzado.

En el contexto digital del siglo XXI, las TIC se están convirtiendo en parte fundamental de la organización social. Estas tecnologías funcionan gracias a programas y aplicaciones que permiten la conectividad y el acceso y uso de la información y el conocimiento. Cómo nos acercamos a estos programas, aplicaciones y sistemas implica tener control o no sobre la tecnología que utilizamos y sobre nuestra propia información.

Una era rica en información electrónica puede lograr maravillosas oportunidades sociales a cambio de meter a la libertad, quizás sin advertirlo, en un congelador (Winner, 1987: 136)

La diferencia entre saber, más allá de lo que podemos ver, lo que ocurre en nuestro ordenador y no saberlo está en la posibilidad de tener acceso o no a todas las instrucciones que están detrás de las interfaces de los programas que utilizamos. Probablemente, no tendremos el conocimiento para, y/o la voluntad de, leer y entender los códigos que le indican a la

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

máquina lo que debe hacer, pero el asunto es si tenemos la libertad o no de aprender ese lenguaje porque podemos verlo.

1.2 *Software* privativo, abierto y libre

1.2.1 Definición de *software*

Al hablar de *software*, como se ha mencionado anteriormente, se hace referencia a aplicaciones de cómputo; más precisamente, se habla de códigos, de lenguaje de programación, órdenes escritas de forma lógica que hacen funcionar de determinada manera lo que vemos en el monitor de la computadora o la pantalla del teléfono celular.

software, según las definiciones más formales, se entiende por el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación (Estándar 729 del Instituto de Ingenieros Eléctricos y Electrónicos). Así *software* refiere, no sólo a las aplicaciones, tomando en cuenta su código fuente, archivo binario e interfaz gráfica, sino también a los sistemas operativos. En sí, entendemos por *software* como aquellos programas de cómputo que hacen funcionar el hardware –es decir, la estructura física, material, como el monitor y el teclado- de la computadora. Para simplificar la definición, se puede decir que *software*, es el conjunto de algoritmos que guían a los circuitos electrónicos para que hagan o dejen de hacer determinado proceso.

Mientras que no siempre se considera la

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

*programación de computadoras como un esfuerzo científico, de hecho, el desarrollo de software es una forma de ciencia, y el software como producto es una forma de propiedad intelectual.*³ (Ostrom et al., 2007: 293)

Para que una computadora haga cualquier tarea, incluso 'iniciarse' y pedir nuestro usuario y clave de entrada, necesita de algún tipo *software*. Actualmente, ya no es necesario saber lenguaje de programación para poder iniciar cualquier aplicación, el ordenador recibe las instrucciones para funcionar sin que nosotros tengamos que hacer aparentemente nada. Sin darnos cuenta, nosotros mismos le decimos qué hacer al presionar los botones del ratón o las claves del teclado⁴. Lo que miramos en la pantalla es una interfaz que presenta de manera “amable” y comprensible lo que tenemos cargado en la máquina. Sin embargo, detrás de la interfaz las órdenes se generan en lenguaje lógico y son recibidas en lenguaje binario; es decir, ceros y unos. Explico esto:

³ Las citas cuyo idioma original es el inglés son traducción propia, excepto cuando así se indique.

⁴ En realidad, siguiendo a Latour (1992), es justamente el *software* el que le dice a la máquina lo que tiene que hacer. Nosotros estamos determinados por esta tecnología a realizar sólo ciertas acciones, y no otras, para indicarle al *software* qué orden enviar, y que así la máquina la realice. Por ejemplo, si queremos utilizar un procesador de texto, vamos al ícono de la aplicación y oprimimos una o dos veces el botón izquierdo del ratón para hacer que la aplicación se inicie. De nada sirve que oprimamos la tecla `ESC` para hacer que el procesador de textos nos abra una página en blanco. En este sentido es que Latour habla de los artefactos como actantes, pues inciden en el actuar de los actores humanos, determinan su acción.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Se necesitan dos procesos principalmente para crear y ejecutar un programa de cómputo. El primero consiste en escribir las instrucciones que ordenarán a la máquina las acciones que debe realizar. Estas instrucciones se escriben en lenguaje lógico, a base de palabras y frases, y corresponden a lo que se denomina “código fuente”. Este código es la base de la operación de una aplicación o programa; conforma los algoritmos que seguirá la máquina para trabajar. En la segunda parte del proceso, el archivo generado debe compilarse; es decir, traducirse a un lenguaje comprensible para la máquina. El lenguaje de la computadora se basa en ceros y unos. El fichero que lleva el código fuente, al compilarse, se convierte en un archivo binario.

En general, al hablar de *software* estamos hablando de una tecnología multifuncional que, junto al hardware, se emplea no sólo en actividades cotidianas de individuos, sino en grandes procesos industriales. Se utilizan programas, aplicaciones y sistemas integrados para manejar los flujos financieros de grandes empresas, mismas que funcionan ya de un modo global, donde el capital no se mantiene fijo ni comprometido con una localidad geográfica. El *software* se emplea en los grandes procesos industriales, como los de plantas automotrices, de alimentos, etcétera, que quedan automatizados al seguir líneas de código específicas para cada momento en el tiempo. Y así como estos programas regulan y organizan la producción masiva, pueden convertirse, junto con artefactos de hardware especializados, en una extensión de la mano del médico en operaciones quirúrgicas para darle mayor precisión y exactitud.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Estos programas, aplicaciones y sistemas⁵ pueden o no permitir que se revisen las líneas de código y se aprenda sobre sus formas de funcionamiento. Ya en muchos países (Malasia, Japón, Estados Unidos) existen empresas que hacen *software* a la medida, siguiendo indicaciones y necesidades específicas de empresas e industrias, sin necesariamente cerrarlo a la libertad de estudio, modificación y uso de usuarios individuales.

1.2.2 Diferencias entre *software* privativo, de fuente abierta y libre

Los programas y aplicaciones que no permiten el acceso a ninguno de los componentes del código fuente, privando de ésta y otras libertades a los usuarios, son conocidos como *software* privativo⁶. En general, se puede decir que la mayoría del *software* que se comercializa a gran escala es *software* privativo. Este tipo de programas utilizan licencias que siempre estipularán que el usuario no puede modificar el producto adquirido,

⁵ Cabe hacer la distinción entre programa, aplicaciones y sistemas. El primero consta de una serie de algoritmos para realizar una determinada función que, en general, es la base de otras instrucciones. Un programa realiza tareas complejas que no están a la vista del usuario común. Las aplicaciones, por su parte, constan de varios programas, y están pensadas para facilitar el trabajo de los usuarios 'legos', p.e. Un procesador de textos es una aplicación que, para funcionar, lleva en su código varios programas unidos. Finalmente, llamamos 'sistema' a todos los programas integrados que proporcionan la base para el funcionamiento de las aplicaciones, p.e. El sistema operativo.

⁶ Antes era llamado *software* "propietario", por la traducción del inglés de "proprietary *software*". Sin embargo, la traducción al español no era correcta, pues lo que se busca enfatizar es la privación en cuanto a la libertad de acceso, modificación y mejora del código fuente.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

tampoco distribuirlo haciendo copias del mismo, entre otras muchas restricciones avaladas por las leyes de derecho de autor, conocidas generalmente por su término en inglés, *copyright*.

Se tiene la idea de que el *software* privativo tiene ventajas por contar con servicio técnico y que, como pasa por controles de calidad para llegar al mercado, su manejo es más seguro y amable para los usuarios. Sin embargo, esto no siempre es verdad. A continuación presento algunos comentarios de usuarios y expertos en informática, comparando distintos sistemas operativos:

“Para echar a perder una caja con Linux, necesitas trabajar para ello; para echar a perder una caja con Windows sólo necesitas trabajar en ella” . Scott Granneman, columnista de Security Focus, en http://www.theregister.co.uk/2003/10/06/Linux_y_s_windows_viruses/

“Hay alrededor de 60,000 viruses conocidos para Windows, 40 más o menos para Macintosh, cerca de 5 para versiones comerciales de Unix, y tal vez 40 para Linux. La mayoría de los viruses de Windows no son importantes, pero muchos cientos de ellos han causado un daño extendido. Dos o tres de los viruses de Macintosh se han diseminado lo suficiente para ser de importancia. Ninguno de los viruses de Unix o Linux se ha diseminado –la mayoría quedaron confinados en el laboratorio.” Dr. Julian Satchell, en *Análisis del impacto del software de*

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

fuelle abierta. En <http://citeseerx.ist.psu.edu/>

“La gente tiene aversión al riesgo. No les gusta probar nuevas cosas. Es la inercia la que mantiene a Microsoft en su lugar.” Roger Kay, presidente de Endpoint Technologies Associates. En http://www.cio.com/article/41140/Windows_vs._Linux_vs._OS_X?page=2&taxonomyId=3081

“Gerentes que están restringidos con el presupuesto tienen que preguntarse a sí mismos si sus organizaciones podrían ahorrar algunos miles de dólares al año al usar un producto de fuente abierta que es casi como Microsoft Office... Esto es sobre asegurarse si la industria puede hacer lo que tiene que hacer mejor, más rápido, más barato y de modo más seguro.” John Halamka, Escuela de Medicina de Harvard. En http://www.cio.com/article/41140/Windows_vs._Linux_vs._OS_X?page=2&taxonomyId=3081)

Hay algunos programas que contienen elementos en su código que pueden verse y modificarse, pero también otros elementos a los que no se puede tener acceso (partes del código que están encriptadas). Estos programas serían semi-libres o semi-abiertos y, aunque también estarían protegidos por los derechos de autor, éstos indicarían restricciones de acceso para las partes privativas, mientras que sus partes abiertas o libres, estarían protegidas bajo diferentes argumentos en la propia

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

licencia. Las diferencias en las licencias sobre derechos de autor serán explicadas en el siguiente capítulo.

Se dice que un programa (*software*) es de fuente abierta (*OS, open source*) cuando permite al usuario observar el primer fichero; es decir, el código fuente. La apertura de códigos de programación es condición necesaria, aunque no suficiente, para poder calificar a una aplicación como *software libre*.

Si un programa muestra su código fuente, permite dos de las libertades fundamentales que son características del *software libre*: estudiar y modificar el código; pero para ser verdaderamente libre, además, debe permitir la distribución y el uso libres. Según palabras de Richard Stallman⁷, no necesariamente se trata de que las aplicaciones sean gratuitas, pero que sí otorguen la libertad al usuario de hacer uso del programa y de redistribuirlo virtualmente.

Dado que nos referimos a la libertad y no al precio, no existe contradicción alguna entre la venta de copias y el software libre. De hecho, la libertad para

⁷ Richard Stallman (RMS) es el fundador del movimiento de *software libre* y creador del concepto *copy-left*, que él denomina con humor como el “izquierdo de autor”, pues utiliza los aspectos legales del derecho de autor para proteger las 4 libertades del *software*. También es el presidente y fundador de la Fundación de *software Libre*, así como el desarrollador principal del proyecto GNU (que significa, GNU is not *Unix*), del que emanan diversas aplicaciones. Con el kernel o núcleo del sistema operativo, desarrollado por Linus Torvalds, Stallman y un equipo de hackers crearon GNU-*Linux* (que comunmente, aunque también incorrectamente, se conoce sólo como el sistema *Linux*).

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

vender copias es crucial: las colecciones de software libre a la venta en formato de CD-ROM son muy importantes para la comunidad y venderlas es una forma de recaudar fondos para el desarrollo de software libre. (Stallman, 2004: 28)

Para que un programa sea libre debe otorgar libertad de acceso al código fuente, libertad de modificación y mejora de dicho código, y libertad de distribución. Un programa puede ser de fuente abierta, pero no permitir la modificación del código, entonces, se dice que el programa es abierto, pero no libre. En general, el *software* de fuente abierta casi siempre también es libre, por lo que se han estudiado los dos tipos de manera conjunta, nombrándolos FLOSS, por sus siglas en inglés (*Free, Libre, Open Source software*), que es como también lo llamaremos en este trabajo.

Que un programa permita el acceso a su código implica que puede estudiarse y, según el tipo de licencia, modificarse, añadiendo, cambiando o borrando instrucciones para mejorarlo y/o personalizarlo a las necesidades de uno o muchísimos usuarios. Además, fomenta una forma distinta de interacción entre el usuario y el programa en la que es posible imaginar distintos modos de realizar acciones, aprender operaciones lógicas, desarrollar ideas y obtener productos personalizados.

Por último, el libre acceso al código fuente motiva a una interacción libre y solidaria entre usuarios, pues las modificaciones que un usuario hace al código fuente pueden compartirse, incluso se pide que así se haga, lo que incide en

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

procesos de innovación desde los usuarios que se convierten en aportaciones para el desarrollo de este tipo de tecnologías (Harhoff, Henkel, & Eric Von Hippel, 2003; E. Von Hippel, 2007; E. Von Hippel & Von Krogh, 2003; Eric Von Hippel, 1988, 2005; K. Lakhani & E. V. Hippel, 2000).

En la actualidad, una gran cantidad de empresas ven al *software* libre y abierto como una oportunidad de negocios. Respaldando con su nombre la calidad de las aplicaciones que producen, se insertan al mercado cobrando sólo por servicios especializados o de apoyo técnico, tal sería el caso de empresas como Red-hat, Canonical, Google, entre otras.

1.2.3 Estudios sobre *software* libre

El tema del *software* libre, en cuanto a las dinámicas de cooperación para su producción y las formas de organización de desarrolladores y usuarios, ha despertado el interés de algunas tradiciones teóricas en el estudio de la esfera social. Tal sería el caso de los abordajes desde los estudios sobre innovación, principalmente de Eric Von Hippel, Wolf, Van Krogh, R. Van Wendel de Joode, J. De Bruijn, y Karim Lakhani; el neo-institucionalismo y los bienes comunes de Lawrence Lessig, Charles M. Shweik, David Bollier, K. Milberry y S. Anderson; los estudios organizacionales de A. Bonaccorsi, A. Hemetsberger, C. Reinhardt y I. Qureshi. También son de mencionar los análisis de A. Karatzogianni y G. Michaelides, K. Crowston, H. Annabi y J. Howison, como el trabajo exhaustivo de Greg Madey, Yongoqin Gao y M. Van Antwerp de la Universidad de Notre Dame.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Desde la ingeniería y los estudios computacionales hay muchísimo trabajo técnico, pero destaco uno que ha sido retomado en el estudio de la colaboración: La Catedral y el Bazaar (Raymond, 1999). El trabajo no sólo describe diferentes dinámicas de organización en el desarrollo de *software* libre, sino que incluso influyó en la decisión de la empresa creadora de Netscape para abrir el código de su navegador, generando con ello el proyecto Mozilla.

Eric Raymond se convirtió en una figura importante dentro del movimiento de fuente abierta por sus aportaciones a la comunidad hacker⁸. La premisa básica del artículo, y que retomaremos en nuestro análisis, es que hay dos formas de organización en la producción de *software* libre. La primera sería la forma “catedral”, que es mas bien vertical, donde un desarrollador o un grupo de ellos dirigen un proyecto desde el inicio, asignando tareas específicas y delimitando la colaboración. La segunda manera de producir, sería el “bazar”, ejemplificado por el proceso que siguió Linus Torvalds con su kernel⁹ “*Linux*”, donde la organización es más horizontal. El

⁸ También es creador del Jargon file o The New Hacker's Dictionary y colaborador de varios proyectos libres como el administrador de correo Fetchmail, Emacs, la biblioteca de GNU, etcétera.

⁹ Kernel: Núcleo. Parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la computadora. Se encarga también del multiplexado, determinando qué programa accederá a un determinado hardware si dos o más quieren usarlo al mismo tiempo. El kernel también ofrece una serie de abstracciones del hardware para que los programadores no tengan que acceder directamente al hardware, proceso que puede ser complicado. Fuente: Diccionario de Informática Alegsa.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

proceso de desarrollo de *software* en la forma “bazar” se abre casi desde el inicio, permitiendo que la colaboración voluntaria se expanda sin limitaciones.

Yo creía que la mayoría del software importante (sistemas operativos y herramientas grandes como el editor de programación Emacs) necesitaba construirse como catedrales, cuidadosamente trabajado por magos individuales o pequeños grupos de magos laborando en perfecto aislamiento, sin versiones beta liberadas antes de tiempo. El estilo de desarrollo de Linus Torvalds –de liberación temprana y frecuente, delegando lo más posible, abierto hasta el punto de la promiscuidad– llegó sorpresivamente. (pp. 2-3)

Karl Fogel (2005), otro personaje conocido en las comunidades y redes de *software* libre y de fuente abierta, al referirse a proyectos de desarrollo de *software* exitosos, indica como importante la generación de una atmósfera que promueva la participación no planeada, a veces, incluso, si esto deriva en la imposibilidad de contar con una administración central.

Los estudios organizacionales analizan el FLOSS, a partir de las motivaciones de los actores para colaborar y el intercambio de conocimientos e información, como en los trabajos realizados por Andrea Hemetsberger, C. Reinhardt e I. Qureshi.

Disponible en <http://www.alegsa.com.ar/Dic/kernel.php>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Andrea Hemetsberger se enfoca en las motivaciones de los actores para contribuir y compartir conocimientos “en-línea”, tomando en cuenta el transfondo ideológico de sus acciones. Esta autora aborda la construcción de conocimiento y las dinámicas de aprendizaje, derivadas de la experiencia concreta y la reflexión colectiva (Hemetsberger & Reinhardt, 2009) y describe los procesos que han llevado a proyectos de FLOSS exitosos, a partir de sistemas que mezclan la acción individual y la colectiva (A. Hemetsberger, 2006).

Por el lado de la innovación, los análisis realizados por Eric Von Hippel con Karim Lakhani (2000), con Van Krogh (2003), Bonaccorsi y Rossi (2002), R. Van Wendel de Joode, De Brujin y Van Eccen (2003), y Lakhani y Wolf (2005), hablan de cómo se desarrolla la innovación, producción, distribución y consumo a través de redes horizontales de usuarios que colaboran entre sí. Estas redes emergen y se mantienen sin necesidad de una organización vertical; es decir, jerárquica, pues sus actores interactúan entre sí al mismo nivel, en cuanto a toma de decisiones e implementación de mejoras. Sus estudios describen el beneficio de aprovechar el conocimiento que está adherido a los usuarios, aportando soluciones abiertamente, poniendo al revés la pregunta sobre el actor racional: “[Si] la libertad de revelar la información lleva a garantizar el acceso a todos los agentes interesados sin imponer un costo. [...]¿Por qué alguna vez se pensó que revelar la información sobre innovaciones no sería común?” (E. Von Hippel, 2007: 13)

En cuanto al abordaje de FLOSS como bien común, de interés general resulta el trabajo de Lawrence Lessig (2004; 2001) quien, desde la perspectiva del Derecho, estudia y analiza

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

la forma de proteger los procesos y productos de estas comunidades. Lessig es un activista comprometido en la difusión y protección de FLOSS y de la cultura en tanto música, videos y textos, a través de las licencias libres, en especial, *Creative Commons*.

David Bollier (1999, 2002, 2006, 2008; Laisné, Aigrain, Bollier, & Tiemann, 2010; Ostrom et al., 2007), desde su habilidad periodística y su interés por los bienes comunes, ha difundido y apoyado el movimiento de *software* libre y de fuente abierta. Sus trabajos abordan las nuevas tecnologías, en especial, la Internet, como espacio de creatividad y colaboración constante. Bollier observa y describe cómo los gigantes empresariales del *software* y los medios de comunicación generan estrategias para la privatización y control del espacio virtual, y enfatiza la importancia de mantener las formas de colaboración solidarias y libres como la de la comunidad GNU-*Linux*.

Charles M. Schweik, desde el marco neoinstitucional y la perspectiva de los bienes comunes, ha analizado las comunidades de desarrolladores de FLOSS, describiendo sus instituciones y formas de organización y producción, así como sus diferencias con otros bienes comunes. (Ostrom et al., 2007; C. Schweik, 2007; C. M. Schweik et al., 2008; C.M. Schweik, English, Paienjtou, & Haire, 2010). Su trabajo será abordado más ampliamente para explicar la clasificación del *software* libre y abierto como bien común en el capítulo 3 de esta tesis.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Finalmente, Kate Milberry y Steve Anderson abordan el estudio del *software* libre y abierto en tanto una práctica liberadora que propone una forma distinta de organización social, más amplia y democrática. Su estudio se enmarca en el estudio de la configuración de las aplicaciones en Internet y de cómo éste espacio puede estar amenazado por la privatización y la restricción de las libertades de los usuarios.

1.3 La emergencia y desarrollo del *software*

El software, como un bien intangible asociado directamente al conocimiento, es un aspecto trascendente en la era de la información, la forma en cómo llegó a existir y cómo se desplegó en toda la economía representa la primera discontinuidad relevante dentro de las TIC (Chandler & Cortada, 2002: 363)

Una computadora, como tal, no puede pensarse sin *software*, por lo que los orígenes de éste deben rastrearse hasta la aparición de la primera computadora, incluso más atrás en el tiempo histórico. Lo que podemos llamar como antecedentes del *software* son productos o ideas, generadas y puestas en práctica, que hacían que una máquina realizara un determinado proceso; es decir, que siguiera una instrucción, un programa.

En este apartado describiré los acontecimientos que abrieron el camino para la emergencia del *software* libre y abierto como bien común. Abordaré la importancia de sus formas de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

producción, reproducción y distribución en la configuración de nuestras sociedades.

1.3.1 Antecedentes de la informática y la computación

El antecedente más claro y lejano del desarrollo de la informática y la computación es en 1642, cuando Blas Pascal (Francia) construyó la primera máquina sumadora. Ésta tenía un mecanismo que constaba de ruedas dentadas; cada diente representaba un número del 0 al 9. La forma en que las ruedas se conectaban entre sí permitía la suma de cifras mayores a dos dígitos. En 1670, la máquina de Pascal es perfeccionada por Gottfried Wilhelm Leibniz (Alemania), quien, además, inventa otra que no sólo sumaba, sino que también podía multiplicar. (Goldstine, 1972)

Estos inventos se dan en un contexto histórico marcado por el mercantilismo y la expansión colonial de los imperios europeos. Al mismo tiempo que los imperios, transformándose poco a poco en Estados-Nación, se expandían, las reformas al interior de los mismos, tanto políticas como económicas, implicaban nuevas configuraciones y cosmovisiones. Es en este tiempo que se da la reforma luterana, que implicó un fuerte cuestionamiento a la iglesia católica. También en el siglo XVII, la ciencia en general se cuestionó y transformó con los estudios previos de Galileo Galilei (Italia), René Descartes (Francia), Baruch Spinoza (Holanda) y, posterior y definitivamente con los de Isaac Newton (Inglaterra).

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

A fines del siglo XVII, medio siglo antes de la Revolución Industrial en Europa y algunos años después de la publicación de los Principia de Newton (1687), el interés por el conocimiento científico decreció, aunque los cambios técnicos seguían dándose (Goldstine, 1972). Si el avance parece haber sido lento, sólo es en comparación con el de la Revolución Industrial. Mientras que Inglaterra y Escocia abrían su camino con rapidez hacia el desarrollo de la Revolución Industrial, Francia no intentaba utilizar maquinaria en gran escala. Sin embargo, en esa época hay en este país un surgimiento brusco de la actividad científica, realizándose investigaciones industriales de importancia que, al no tener respuesta en Francia, fueron aprovechadas por los ingleses. El contexto de la Revolución Industrial en Europa es importante, pues fue la necesidad por automatizar procesos y manejar producción a gran escala lo que socialmente motivó el desarrollo de las máquinas, y con ello, de la programación de las mismas.

A inicios del siglo XIX, la historia de los lenguajes de programación y de la informática dió un salto evolutivo importante con el uso de las tarjetas perforadas. La innovación emergió en la industria textil cuando Joseph Marie Jacquard, inspirado por la forma de funcionar de las pianolas, utilizó moldes metálicos perforados en las máquinas tejedoras. Con estas plantillas, se programaban, a través de secuencias de pasos, las puntadas para controlar las tramas y dibujos del tejido.

Charles Babbage, en 1822, terminó el diseño de una máquina que podía realizar cualquier tipo de cálculo. La llamó “Artefacto analítico”. Las ideas concebidas para la construcción de esta

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

máquina se convirtieron en el antecedente más claro del tipo de procesos que una computadora hoy puede ejecutar. Aunque Babbage no pudo construir su artefacto, sus diseños influyeron en todas las máquinas posteriores del tipo. Las computadoras digitales de la actualidad tienen al artefacto analítico como base en su diseño. La máquina de Babbage sí se construyó, aunque fue hasta 1989. (Bernal, 1997)

En 1989, las tarjetas perforadas y el artefacto analítico trabajaron juntos, gracias a una sugerencia de Lady Ada Augusta Lovelace de 1843. Las tarjetas perforadas posibilitarían que la máquina de Babbage efectuara ciertas operaciones iterativamente. Por su trabajo, se considera a Charles Babbage el “padre de la informática” y a Ada Augusta, la primera programadora. (Copeland, 2008)

Sin embargo, como menciona Mattelart (2002), un ingrediente más, al menos, era necesario para que la informática, como disciplina pudiera desarrollarse. Este ingrediente era la formalización del concepto de 'algoritmo' y su manifestación en términos de lenguaje:

Para que el algoritmo, o secuencia ordenada de operaciones elementales extraídas de un repertorio finito de operaciones ejecutables en un tiempo dado, se convierta en concepto fundamental del procesamiento automático de la información habrá que esperar, sin embargo, a la mediación de la escritura algorítmica. Formulada en 1854 por el irlandés George Boole, permitirá, un siglo más tarde,

la construcción de la informática como disciplina autónoma. (p. 19)

1.3.2 El código fuente. Álgebra, lógica y lenguajes de programación

Un ingrediente fundamental para la programación es el álgebra de Boole, desarrollada por George Boole y publicada en 1854. El sistema booleano tiene como base el cálculo proposicional y la teoría de conjuntos¹⁰ y es la base del lenguaje de programación (Weeda, 2004). Fue William Stanley Jevons quien, en 1869, inventó la máquina que por primera vez utilizó el álgebra de Boole para resolver una serie de silogismos complicados. La máquina fue llamada “piano lógico”.

La utilización del álgebra booleana ha conformado la forma en la que los aparatos digitales trabajan y, con ello, la forma en que nosotros mismos nos relacionamos con este tipo de aparatos. Latour, sociólogo de la Ciencia y antropólogo francés del siglo XX, menciona al respecto de la acción de los objetos (y

¹⁰ De la Teoría de conjuntos es importante la definición de los mismos; a saber: un conjunto se define por extensión (se enumeran sus componentes) o por comprensión (definiendo una propiedad determinada de los elementos que lo conforman). Lo que interesa para la informática es cuando un elemento es parte o no de un conjunto, lo que unido al cálculo proposicional nos da valores de verdad: 1, si es cierto; 0 si es falso. Es decir, como base del lenguaje de programación se tendrán proposiciones lógicas y conectivas (y, o, si... entonces, no, etcétera) que, dependiendo de su valor de verdad (cierto o falso) echarán a andar o detendrán un proceso. El 0 y el 1 representan los valores del código binario que es el lenguaje que entiende la computadora. Para una explicación más precisa ver: (Borge Holthoefer, n.d.)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

lo que los hace funcionar) sobre la determinación del actuar de los sujetos¹¹ lo siguiente:

[...] basta que observemos el mecanismo que rige un semáforo o el funcionamiento de un sistema informático para darnos cuenta que el álgebra de Boole juega un papel nada despreciable no ya en el ámbito específico de la Lógica, sino en la civilización tal y como la conocemos. (Borge Holthofer, n.d.)

1.3.3 Máquinas, mecanismos y herramientas

La evolución de los artefactos y máquinas lógicas siguió su curso de manera acelerada. En 1884, apareció la primera calculadora que permitió a los usuarios presionar teclas, en lugar de hacer deslizar ruedas, para hacerla funcionar. Se llamó Computómetro, y fue creada por Dorr Felt. Otto Sieger, por su parte, hizo evolucionar la máquina de multiplicar de Leibniz, automatizándola (Borge Holthofer, n.d.).

Sin duda, otros inventos importantes en la cronología de la historia del *software* son el tubo de vacío y el uso y manipulación de los circuitos eléctricos. En cuanto al tubo de vacío, el primero fue llamado Audion e inventado por Lee De Forest en 1906. Sirvió para encontrar y ampliar señales de radio. Treinta años más tarde, sería parte del funcionamiento de una primera generación de computadoras. Por su parte, el flip-flop es un circuito multivibrador, desarrollado por Eccles y Jordan en 1919.

¹¹ Sobre las formas de determinación del sujeto por parte de los objetos, ver: (Latour, 1992)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Éste permitía que los circuitos electrónicos tuvieran dos estados estables para representar de modo alternativo el 0 y el 1. El flip-flop es parte de la base de almacenamiento y proceso de los bits en el código binario, el lenguaje que comprende el ordenador para funcionar (Copeland, 2008).

1936 fue un año clave para el desarrollo de la informática y las máquinas digitales modernas: se diseñó y fabricó la primera máquina programable que utilizaba cintas perforadas: la Z1 de Konrad Zuse; G.R. Stibitz desarrolló la primera calculadora que utilizó lenguaje binario; Alan Turing (Lee, 2011) formalizó el concepto de algoritmo con la descripción de su máquina y, S.B.Wright y E.R.Taylor desarrollaron una sumadora análoga que servía para controlar las instalaciones de radio trasatlánticas.

Entre 1939 y 1942, Atanasoff Berry construyó la primera computadora digital, la ABC, en la Universidad de Iowa. Mientras que Konrad Zuse perfeccionaba su máquina, construyendo la Z3, y Stibitz la Modelo II de los Laboratorios Bell, utilizando códigos de corrección de errores. Según Sampedro:

En la década de 1940, International Business Machines (IBM) introduce las primeras computadoras centrales (mainframes) fabricadas primero con bulbos y luego con transistores, y utilizadas en proyectos científicos y militares. (Barker-Plummer, 2011)

1.3.4 De las super-computadoras al ordenador personal

Los años que siguieron generaron nuevas computadoras: el *Colossus* en el 43, la *Mark I* de Harvard en el 44, la *ENIAC* (Computador e Integrador Numérico Electrónico) de la Universidad de Pensylvania en el 46 (Sampedro Hernández, 2011), la *Gypsy* o *GPAC* (Computadora Análoga de Propósito General) de E. Lakatos y la *EDSAC* (Calculadora Automática de Almacenaje Retardado Electrónico) de la Universidad de Cambridge en 1947. Este último año, el 47, también vio aparecer la Teoría de la Información de Claude Shannon¹² y los fundamentos para la construcción de una red de comunicación de Paul Baran¹³; además, la primera memoria, que fue desarrollada por Jay Forrester, desplazando a los tubos de vacío. Para 1948, la *SSEM* (Máquina Experimental de Pequeña Escala de Manchester), *Baby*, corrió su primer programa (Lee, 2011).

1950 se convirtió en el año que fue testigo del uso del transistor en los ordenadores, y con ello nuevas computadoras: la *EDVAC* (Computadora Electrónica Automática de Variable Discreta); en el 51, cuyo lenguaje ya era binario; Eckert y Mauchly dieron a conocer el *UNIVAC I* (Computadora

¹² Esta teoría determina la cantidad de información que puede ser enviada en cada unidad de tiempo, según un determinado sistema con una cantidad limitada de poder de transmisión.

¹³ Las ideas de Paul Baran, desarrolladas para el departamento de defensa estadounidense, son una base fundamental para entender el funcionamiento de Internet.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Automática Universal) que, según Forrester, fue la primera computadora comercial (Lee, 2011), y Grace Murray Hopper inventó el sistema A-0, un compilador para computadora. Ya para 1955, *IBM* había fabricado su primera computadora industrial, se crearon las memorias de núcleos, se usaron algoritmos más eficientes gracias a Edsger Dijkstra y su computadora *ARMAC* (Máquina de Computación Automática del Centro de Matemáticas, Dinamarca) (Copeland, 2008; Lee, 2011); se utilizaba ampliamente el lenguaje ensamblador y se desarrolló el lenguaje de programación *FORTTRAN* (Sistema de Traducción de Fórmulas) (Copeland, 2008; Lee, 2011). Además los laboratorios *Bell* comenzaron a probar el envío de datos a distancia a través de líneas telefónicas, y nació el concepto de inteligencia artificial.

La década de los 60 inicia ya con la segunda generación de computadoras, éstas ya no utilizaban los tubos de vacío. Una de las primeras computadoras de esta generación fue *ATLAS* de la Universidad de Manchester, la cual utilizaba tuberías, memoria entrelazada, virtual y paginada. Fue la máquina más poderosa en 1962. En cuanto a lenguajes de programación¹⁴, apareció *ALGOL*, *COBOL*, *APL*, *ALTRAN*, *BASIC*, *L6* (Wexelblat, 1981). Hart y Levin en el MIT crearon *Lisp*, un compilador

¹⁴ Los lenguajes de programación son conjuntos de símbolos y palabras clave, que tienen ciertas reglas gramaticales para generar instrucciones. Cada lenguaje sigue un paradigma de programación; es decir, una forma de razonar y resolver problemas, así como de estructurar los programas. La programación puede ser: imperativa, como en Pascal, Ada, Cobol, C, Modula-2, Fortran; funcional, como la de Lisp; lógica, como Prolog, u orientada a objetos, como en LOO, Smalltalk, C++, Java. (Ver Javier Rodríguez Sala, 2003)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

autocontenido. Steve Rusell escribió Spacewar, el primer juego de computadora. Ivan Sutherland desarrolló programas gráficos. Se definió el código *ASCII* (Código Estándar Americano para Intercambio de Información) como estándar para caracteres en computadoras.

Esta década vio las primeras computadoras que se comercializaron a gran escala: *DEC* (Sampedro Hernández, 2011) y *CDC6600*, y la emergencia del diskette o disco flexible. Los laboratorios *Bell*, el MIT (Instituto Tecnológico de Massachusetts) y *General Electric* desarrollaron el Servicio de Información y Computación Múltiplexed (*Multics*, por sus siglas en inglés) (Sampedro Hernández, 2011), que sería reemplazado por la aparición del primer sistema operativo en 1969: *Unics*, o como hoy se le conoce, *Unix* (Van Vleck, 2011)

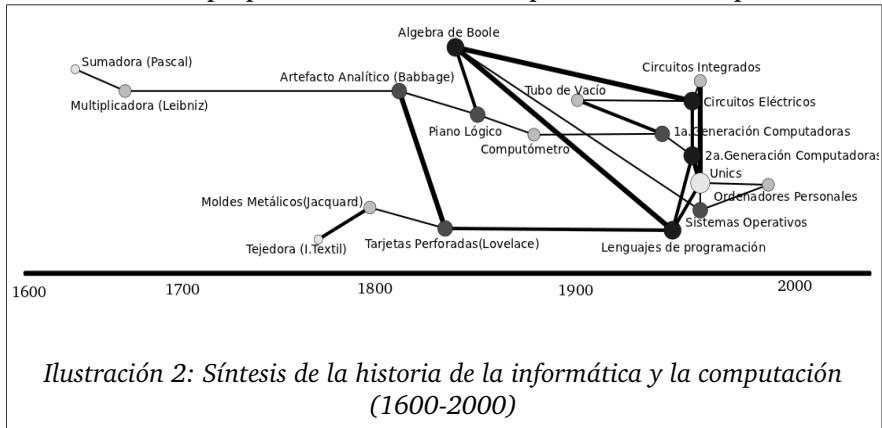
A inicios de 1970, y a lo largo de toda la década, las innovaciones en informática y computación empezaron a generar curiosidad en la población mundial. Ya no sólo se pensaba en producir ordenadores para laboratorios y centros de investigación universitarios, sino también para los hogares. Para lograrlo, era necesario disminuir el tamaño y el precio de las máquinas. La solución fueron los circuitos integrados o microchips¹⁵, que reemplazaron a los transistores y tubos de vacío. (Ritchie, 1996)

¹⁵ Los circuitos integrados, a grandes rasgos, son pequeñas cajas con material semiconductor, que contienen varios circuitos electrónicos en su interior.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

A lo largo de esa década se siguió evolucionando en videojuegos (en 1972 nació *Atari*), y, por supuesto, nuevos lenguajes de programación, como *Pascal*, *C*, *C++*. Para 1980, ya existía la empresa *Apple* de Steve Jobs y *Microsoft* de Bill Gates, así como el juego Pacman de *Namco*, y programas procesadores de texto y hojas de cálculo. Mientras tanto, el sistema *Unix* se licenciaba, preparando el terreno para su apropiación comercial, lo que marcó un cambio en la forma en que se desarrollaba y distribuía el *software* hasta ese momento.

Elaboración propia con base en la descripción histórica expuesta



El gráfico mostrado es una síntesis de lo que se ha expuesto; presentarlo visualmente, aunque de manera muy simplificada, permite notar la rapidez de los cambios tecnológicos en el ámbito informático desde mediados del siglo XX. Es de importancia destacar que el Artefacto Analítico de Babbage, concebido desde 1820, no fue construido sino hasta cien años después de su muerte. Las razones del gobierno para negarse a otorgar fondos a este científico son desconocidas y, por tanto, parecen responder a criterios diferentes al desarrollo técnico, tal

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

vez económicos o políticos. Fuera como fuera, la máquina vio la luz hasta 1989. Para ese entonces, como se ha comentado y puede distinguirse en el gráfico, ya estábamos viviendo con la 3ª. generación de computadoras e inmersos en el rápido desarrollo de sistemas operativos, lenguajes de programación, así como aplicaciones y programas diversos. A principios de 1990, el ordenador personal ya se comercializaba de manera cotidiana, insertándose en nuestras vidas de manera contundente. No puede saberse realmente lo que hubiera sucedido si, para finales del siglo XIX, el artefacto de Babbage hubiera visto la luz, utilizando, como lo dijo Lady Lovelace, las tarjetas perforadas como forma de programación para su funcionamiento.

También el gráfico muestra claramente cómo hasta mediados del siglo pasado la construcción de las máquinas integraba los mecanismos y procesos; es decir, se mantenía el hardware y el *software* como una sola entidad. La diferenciación entre ambas cosas puede explicarse de manera sociotécnica, como lo hace B.Jesiek (2006). A grosso modo, lo que Jesiek explica son las repercusiones de la especialización diferenciada entre ingenieros y programadores.

1.3.5 *Unix* y el movimiento de *software* libre y abierto

Como se describió en el apartado anterior, los productores de computadoras hasta finales de los 60, no veían una separación comercial entre hardware y *software*. Los ordenadores para funcionar necesitaban de códigos, algoritmos, y programas y, por tanto, se vendían integrados. El objetivo de las empresas

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

para acceder a un consumo masivo era hacer más eficientes, pequeñas y menos costosas las máquinas como un todo; es decir, el conjunto de hardware y *software*. Los programadores, según diversas fuentes, solían intercambiar el *software* desarrollado, siguiendo intereses científicos y técnicos; incluso, de colaboración para la innovación entre pares.

La compartición de software ha estado presente tanto como el software mismo. En los primeros días de las computadoras, los fabricantes sentían que las ventajas competitivas se encontraban principalmente en la innovación de hardware, por lo que no ponían mucha atención al software como un activo de negocio. Muchos de los clientes para estas máquinas fueron científicos o técnicos, quienes podían modificar y mejorar el software adquirido con la máquina. Los clientes algunas veces distribuían sus parches al propio fabricante y a otros propietarios de máquinas similares. Los fabricantes toleraban e, incluso, motivaban esto: a sus ojos, las mejoras al software, sin importar de dónde vinieran, sólo hacían la máquina más atractiva a otros clientes

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

potenciales.¹⁶ (Fogel, 2005)

Richard Stallman, creador del procesador GNU-Emacs y el sistema operativo GNU-Linux, relata en su libro *Free software, free society (2002)* cómo era la comunidad de programadores en los años 70, en tanto las formas de compartir y distribuir el *software* desarrollado.

No llamábamos «software libre» a nuestro software porque el término no existía todavía; pero era exactamente eso. Cuando alguien de otra universidad o de otra empresa quería instalar y utilizar un programa, se lo prestábamos de buen grado. Si descubrías a alguien utilizando un programa poco habitual e interesante, siempre podías preguntarle por el código fuente, leerlo, modificarlo o tomar partes de él para montar un programa nuevo. (Stallman, 2004: 19)

En general, los ordenadores que se utilizaban antes de la creación de *Unix* se encontraban fundamentalmente en las

¹⁶ *Software sharing has been around as long as software itself. In the early days of computers, manufacturers felt that competitive advantages were to be had mainly in hardware innovation, and therefore didn't pay much attention to software as a business asset. Many of the customers for these early machines were scientists or technicians, who were able to modify and extend the software shipped with the machine themselves. Customers sometimes distributed their patches back not only to the manufacturer, but to other owners of similar machines. The manufacturers often tolerated and even encouraged this: in their eyes, improvements to the software, from whatever source, just made the machine more attractive to other potential customers. La traducción es propia.*

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

universidades, grandes laboratorios y algunas empresas. No existían computadoras personales; las que había eran grandes máquinas que se compartían entre los investigadores, científicos o trabajadores. Aunque algunos mejoraron con el tiempo (por ejemplo, el caso de Multics), para finales de los 60, los sistemas en general eran muy lentos y las máquinas muy caras.

En 1969, los ingenieros que trabajaban en los laboratorios *Bell* crearon *Unics*¹⁷, un sistema operativo mucho más eficiente para cumplir con los requerimientos hacia el interior de la empresa. La innovación de este sistema operativo fue que era un diseño modular; es decir, sus componentes se dividían en diferentes partes que podían trabajar de manera relativamente independiente¹⁸, lo que lo hacía más ligero y veloz para cumplir con las tareas asignadas.

Sin entrar en detalles y resumiendo se puede decir que Unix responde de manera satisfactoria a lo establecido en los cuadernos de mando: está formado por un nodo (el kernel) muy reducido y sencillo, sobre el que se añaden distintos módulos que pueden activarse y detenerse conforme se desee, sin tocar el sistema general. En resumen, Unix representa, al contrario que en los sistemas anteriores que eran cerrados, un sistema compuesto de partes relativamente independientes entre sí, lo que le proporciona una estabilidad hasta entonces

¹⁷ Después cambió de nombre a *Unix*

¹⁸ Los sistemas *Unix* constan de tres niveles: el kernel o núcleo, que calendariza las tareas y administra el almacenamiento; la “shell”, que conecta e interpreta las órdenes del usuario, llama a los programas y los ejecuta; y finalmente, las herramientas y aplicaciones.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

desconocida y un fácil manejo. (Mounier, 2002)

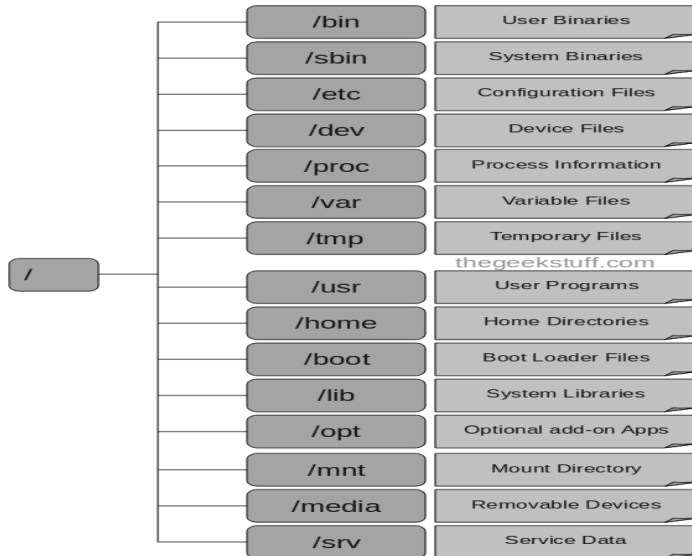


Ilustración 3: Estructura de Linux por Ramesh Natarajan
Fuente: <http://www.thegeekstuff.com/2010/09/linux-file-system-structure/>

Los laboratorios *Bell* trabajaban para la empresa telefónica estadounidense *AT&T*, la cual utilizó el sistema *Unix* para sus centrales telefónicas. En ese momento, *AT&T* no vio el sistema operativo como susceptible a ser comercializado por dos razones: 1) no era usual la venta de este tipo de productos, y 2) el decreto de 1956 restringía a la empresa a obtener beneficios sólo en el ramo de la telefonía; los sistemas operativos pertenecían a otro sector, el de la informática.

Según Manuel Castells, sociólogo español del siglo XX, en *Galaxia Internet* (2001), el sistema *Unix*, creado por Ken

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Thompson y Dennis Ritchie, fue distribuido a universidades, debido a que el gobierno norteamericano obligó a *AT&T* en 1974 a difundir los resultados de las investigaciones de los laboratorios *Bell*. Pierre Mounier, antropólogo francés del siglo XX, en cambio, en *Los dueños de la red (2002)* relata que fueron los propios ingenieros de los laboratorios *Bell* quienes decidieron compartir el sistema creado por ellos a las universidades que lo pedían. Lo que confirma la frase de Stallman sobre la cultura de compartir que se tenía entonces.

Haya sido por obligación o por solidaridad, el sistema *Unix* comenzó a distribuirse por una módica cantidad que cubría solamente los gastos de reproducción y envío. Los ingenieros de *Bell* no se comprometían a dar apoyo técnico, por lo que *Unix* se distribuyó con código abierto; es decir, permitía que los usuarios accedieran al código fuente y, además, lo modificaran y mejoraran para adaptarlo a sus necesidades.

*[la accesibilidad y falta de apoyo técnico de Unix]
son la causa del nacimiento de una comunidad de
usuarios que rápidamente desarrolla una cultura de
ayuda mutua por igual y la implanta en Internet.
(Mounier, 2002)*

Mounier comenta en su libro que el desarrollo de *Unix* fue apoyado por ARPA para unificar los lenguajes y procesos utilizados por las distintas culturas informáticas de entonces: *Unix*, con *C*; *PC* y microordenadores con *Basic* y *PDP-10* (que eran fundamentalmente las que utilizaba el proyecto de Arpanet) con *Lisp*. Así se crearon algunas variantes, todas

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

basadas en el sistema *Unix*, como *BSD*, la versión de Berkeley. Debido a este libre acceso al código, de *Unix* derivaron varios sistemas operativos.

La Universidad de Berkeley y *AT&T* se encontraron en un largo juicio legal por derechos de propiedad hasta 1994, ya que la empresa quedó desregulada en 1984, lo que permitió a los laboratorios *Bell* licenciar y empezar a comercializar el sistema *Unix*.

Con la desregulación y el deseo de comercialización de *AT&T* y *Bell*, el sistema *Unix* perdió las libertades de las que antes gozaba: acceso al código, posibilidad de mejorarlo y distribuirlo libremente. Al mismo tiempo, el desarrollo de microordenadores y la conexión a Internet se aceleró, abriendo la posibilidad al público de tener computadoras personales y conectarse virtualmente. Era necesario un nuevo sistema operativo libre que funcionara de manera eficaz para permitir las conexiones a Internet y, en el mundo hacker, fortalecer y continuar la cooperación para el desarrollo de programas. Este sistema fue *GNU-Linux*, creado por Richard Stallman con base en el núcleo de Linus Torvalds.

Dado que cada uno de los componentes del sistema GNU se implantó en un sistema Unix, todos ellos podían ejecutarse en sistemas Unix mucho antes de que existiera el sistema GNU. Algunos de estos programas se hicieron muy populares y los usuarios empezaron a ampliarlos y a transportarlos a las diversas versiones incompatibles de Unix, y también

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

a otros sistemas. El proceso dotó de mayor potencia a estos programas, y atrajo tanto fondos, como colaboradores al proyecto GNU [...] En 1991, Linus Torvalds desarrolló un kernel compatible con Unix y lo llamó Linux. En el año 1992, la combinación de Linux con el incompleto sistema GNU dió como resultado un sistema operativo libre. (R.M. Stallman, 2004: 35 y 37)

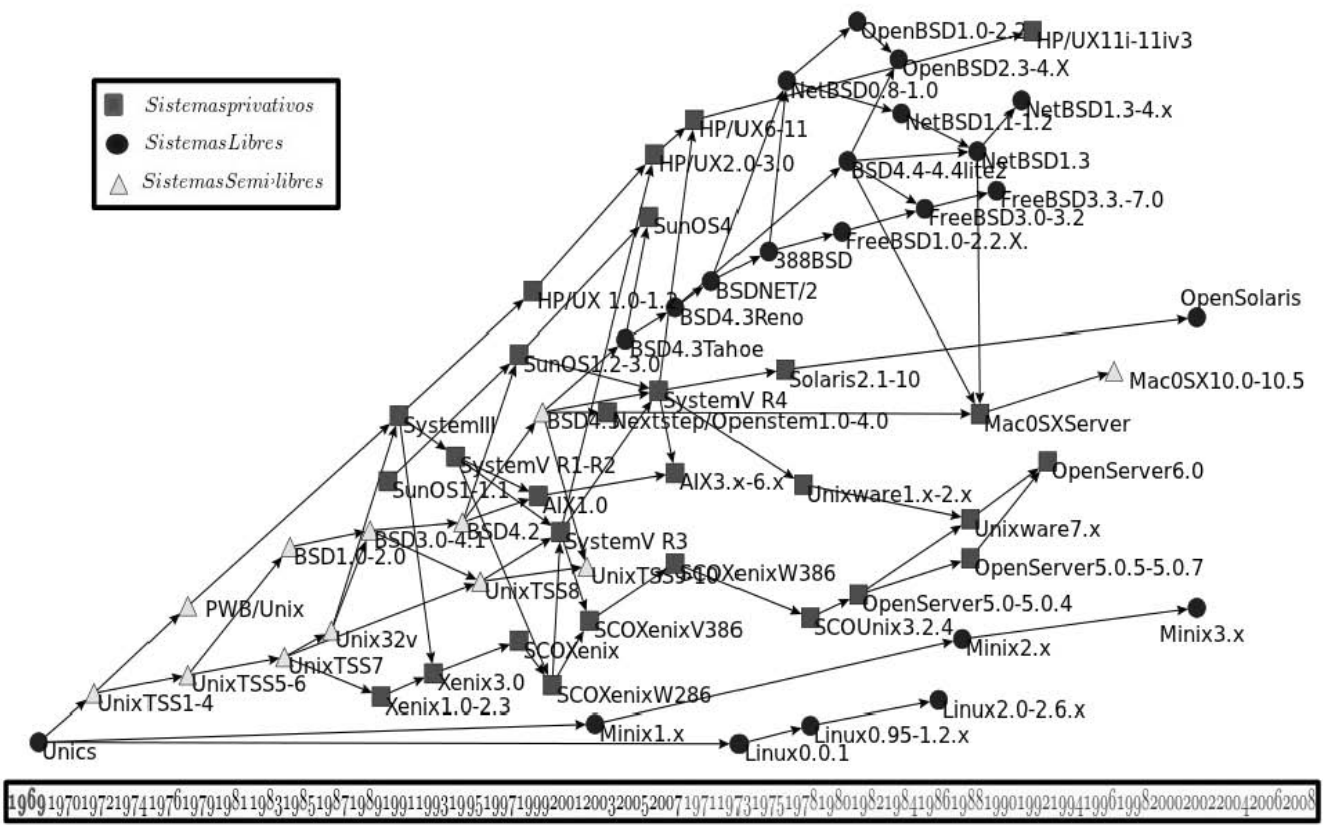


Ilustración 4: Red histórica del desarrollo de sistemas operativos, a partir de Unics 1969-2008. Elaboración propia

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

La red presentada en la ilustración 4 muestra, como se indica, el desarrollo de distintos sistemas operativos a partir de *Unics*, el creado por los laboratorios *Bell*. Los círculos representan sistemas operativos libres; los cuadrados, privativos, y los triángulos semilibres; es decir, aquellos que tienen componentes privativos y libres en su código.

El diagrama en red demuestra que en un inicio, en 1969, el producto del desarrollo de sistemas operativos fue libre en *Unics*, y semilibre a partir de él. Es hasta los 80 con *Xenix*, *Sun* y *System*, que las empresas cierran sus códigos, o bien, desarrollan de manera cerrada estos sistemas, lo que es lógico pues, como se mencionó, la regulación que afectaba a *AT&T* para la comercialización de *Unix* quedó derogada en 1984. Antes de esta fecha, el sistema operativo *Unix* se podía utilizar libremente, de menos, aquellas partes del código que ya habían sido expuestas desde 1969.

Finalmente, la ilustración indica también que el producto del desarrollo de sistemas libres se dio hasta finales de los 80 con la emergencia de *Minix* y, posteriormente, *BSD Tahoe*. Berkeley, como se mencionó, estuvo en un largo proceso legal con los laboratorios *Bell*, lo que explica porqué sus desarrollos (desde *BSD1.0* hasta *BSD4.3*) tuvieron que mantenerse como sistemas semi-libres (teniendo que mantener el código de *Unix* cerrado, aunque los componentes desarrollados por la propia universidad no).

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Merece mención la aparición, como muestra la red de la ilustración 4, de *Open Solaris* en 2005, cuyos antecedentes fueron sistemas privativos de uso comercial. No tan radical, pero se puede decir lo mismo de la presentación en 2002 de *MacOSX10-10.5* como sistema semilibre. Lo que puede explicar la decisión de esta empresa de abrir uno de sus productos y presentarlo como sistema libre es, por un lado, que existe un incremento en el uso de sistemas y *software* libre y, más importante desde el punto de vista del mercado, que es posible generar *software* libre y aún así obtener importantes ganancias. Esto es una muestra del cambio de perspectiva que deviene de la evolución hacia la economía de servicios.

Los desarrollos semilibres desaparecen prácticamente de la escena desde mediados de los 80 con la aparición de los desarrollos libres, *BSD-Tahoe* y *Minix*. Cabe mencionarse que es en 1985 cuando Richard Stallman decide defender la libertad de usuarios y programadores y encabeza el movimiento de *software* libre. Su primer desarrollo fue el programa *E-macs*, pero su objetivo desde entonces fue lograr un sistema operativo completamente libre, lo que logró con la emergencia del kernel *Linux* en 1991. Richard Stallman, junto con cientos de programadores, desarrollaron el sistema *GNU-Linux*, base de la mayoría de las distribuciones libres que se conocen en la actualidad.

Richard Stallman siguió y sigue motivando y apoyando del desarrollo del *software* libre a través de la Fundación de *software*

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Libre¹⁹. En un principio, las aplicaciones generadas no eran fáciles de utilizar para el usuario común (sin conocimientos en programación) lo que repercutió en la adopción y expansión de la filosofía de la libertad y de las aplicaciones libres. Mientras tanto, con AT&T bloqueando el desarrollo de *software* por “defender” sus derechos de propiedad sobre *Unix*, *Microsoft* se convirtió en un monopolio comercial que, hasta la fecha, mantiene su dominio en las formas de relacionarnos con nuestra información en el ordenador, con las formas en que accedemos a ella y la manipulamos, con los procesos dictados por el código que, con los sistemas privativos, no conocemos. En palabras de Lawrence Lessig:

El «código» es la tecnología que hace que los ordenadores funcionen. Está inscrito en el software o grabado en el hardware, es el conjunto de instrucciones, primero escritas como palabras, que dirigen la funcionalidad de las máquinas. Estas máquinas (ordenadores) definen y controlan cada vez más nuestras vidas. Determinan cómo se conectan los teléfonos y qué aparece en el televisor. Deciden si el video puede enviarse por banda ancha hasta un ordenador. Controlan la información que

¹⁹ En 1985, Richard Stallman crea la Fundación de *software* Libre. El objetivo principal de la misma es la protección y difusión de los programas libres, a través del apoyo al desarrollo, uso y distribución de los mismos, y también apoyando a desarrolladores en el uso de licencias GNU-GPL para proteger su trabajo de la privatización por parte de empresas. La Fundación, además promueve diversas campañas de apoyo a la difusión de la libertad de los usuarios y contra el uso y abuso de los programas y aplicaciones restrictivos

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

un ordenador remite al fabricante. Estas máquinas nos dirigen. El código dirige estas máquinas. (R.M. Stallman, 2004: 11)

El desarrollo de *software* libre no puede entenderse sin el espacio en el que la colaboración se expresa. El desarrollo de *software* libre, como se ha visto, existe desde antes de Internet, pero la expansión de la colaboración, el fortalecimiento de las comunidades de desarrollo, el mejoramiento de las aplicaciones por la retroalimentación entre pares y la construcción y mantenimiento de una enorme red de hackers y usuarios se debe a la existencia de la red de redes, que aún permite el libre acceso sin ningún tipo de distinción.

1.3.6 Arpanet, Usenet, Internet. El espacio de la colaboración

Es en la década de los 70 en el siglo XX que el desarrollo de la red de redes sentó las bases más firmes, pues se desarrollaron las formas de conexión, así como los sistemas *LAN* y *Ethernet*, y los protocolos de transmisión de datos: *FTP* y, en los 80's, *TCP/IP*, *http* y *www*. Además se colaboró en el desarrollo de programas de correo electrónico.

En 1980, la red DARPA (antes ARPA) se encontraba en funcionamiento, aunque sólo entre sus miembros, por lo que se extendió la red alternativa Usenet, creada por Tom Truscott y Jim Ellis. El sistema *Unix* tenía una herramienta llamada *UUCP* (por *Unix to Unix copy program*) que permitía hacer copias de programa. Truscott y Ellis aprovecharon esta herramienta para

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

intercambiar mensajes y con ello emergió la red Usenet. Es con la red Usenet y el *UUCP* de *Unix* que inicia lo que hoy conocemos como *P2P* o el intercambio entre pares (*peer to peer*). Motivados, entre otras cosas, por el gusto por la ciencia ficción, buscaron que su red pudiera conectarse a la red Arpa (Mounier, 2002: 52).

Desde mediados de los años 60, se habían desarrollado entre los arpanautas grupos de intercambio de información a través del correo electrónico, lo que actualmente se llaman listas de debate. De entre varias, dos destacaban por su popularidad: Human-net y SF-lovers. [...] La posibilidad de conectarse a Arpanet fue el factor que hizo que la red [Usenet] despegase realmente.

SF-Lovers se convirtió en un foro de debate en el que todos querían participar. La posibilidad de hacerlo era conectándose a la red Arpa. Usenet lo lograba con muchas dificultades, pero con el apoyo de la Universidad de Berkeley, que participaba en Arpanet, se logró que se generaran interfaces entre las dos redes.

Usenet empezó a crecer en número de usuarios, generando una identidad común entre ellos que sentó las bases de una ideología de igualdad y libre acceso. Al mismo tiempo, el movimiento de *software* libre, liderado por Richard Stallman, comienza a surgir como respuesta al cierre comercial del sistema *Unix*. Tomando en cuenta que los usuarios de Usenet también

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

eran usuarios de *Unix*, ellos también se vieron afectados por la decisión de los laboratorios *Bell* sobre el sistema operativo.

Sin duda, estas condiciones particulares, sumadas a que los primeros participantes de la red Usenet pertenecían exclusivamente al mundo académico, determinan el ambiente predominante contrario a la comercialización que todavía domina en los foros. (Mounier, 2002)

Otras redes también emergieron en diferentes universidades estadounidenses, como Fidonet de Tom Jennings, que utilizaba las líneas telefónicas para conectar ordenadores, y BitNet para usuarios de *IBM* de Ira Fuchs y Greydon Freeman (Mounier, 2002).

La forma en la que se organizó horizontal y sin jerarquías la red Usenet, así como Fidonet, en especial, en los foros de debate, fundó la forma de intercambiar conocimientos y habilidades en la producción, uso y distribución de FLOSS. Las necesidades técnicas para generar nuevos sistemas operativos, aplicaciones nuevas entre usuarios y mejorar las interfaces entre las redes generaron un sistema colaborativo, basado en que “el enriquecimiento colectivo se consigue gracias al intercambio libre y gratuito.” (ibid: 56).

El impacto de las redes autónomas fue también decisivo en la expansión global de las redes informáticas [...] Las redes basadas en el UUCP se globalizaron mucho antes que Internet, sentando así

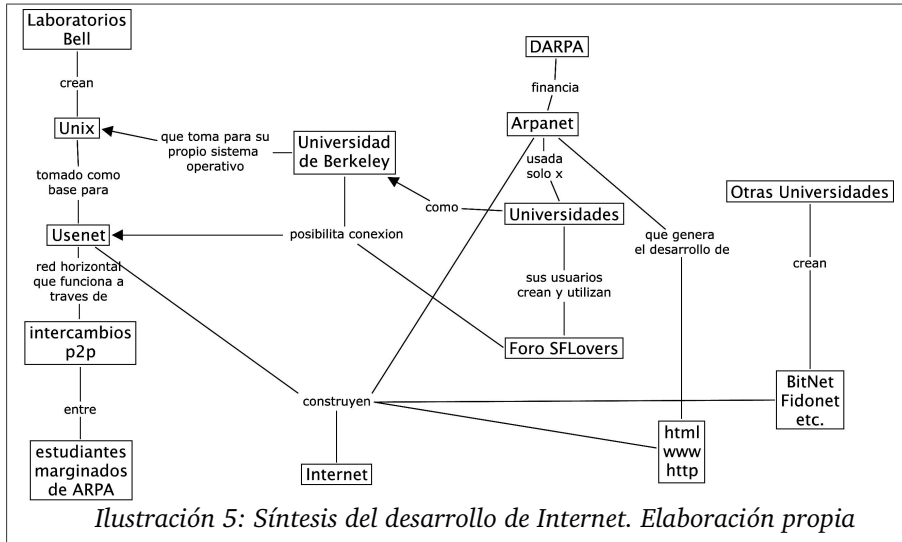
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

las bases para su universalización, en cuanto las redes pudieron conectarse unas con otras. (Castells, 2001)

El Departamento de Defensa de Estados Unidos liberó la red ARPA de su control a inicios de los años 90, optando por la comercialización de la tecnología de Internet, con lo que muchas empresas entraron al mercado de la provisión de servicios de Internet (ISPs, por sus siglas en inglés), a través de *gateways*. Fue Tim Berners-Lee quien, con el desarrollo de lo que hoy conocemos como URLs, hizo que Internet se convirtiera en una red global. Junto con Robert Cailliau desarrollaron el sistema que unía al navegador con el editor, el sistema de hipertexto *www* (siglas que corresponden a *world wide web*) (ibid: 28-29).

La amplia difusión de Internet a principios de la década de 1990 aceleró el movimiento de SWFA [software de fuente abierta]. En este periodo nacen nuevos proyectos de fuente abierta como Linux, cuya difusión se acelera a través de las redes creadas en Internet. (Sampedro Hernández, 2011)

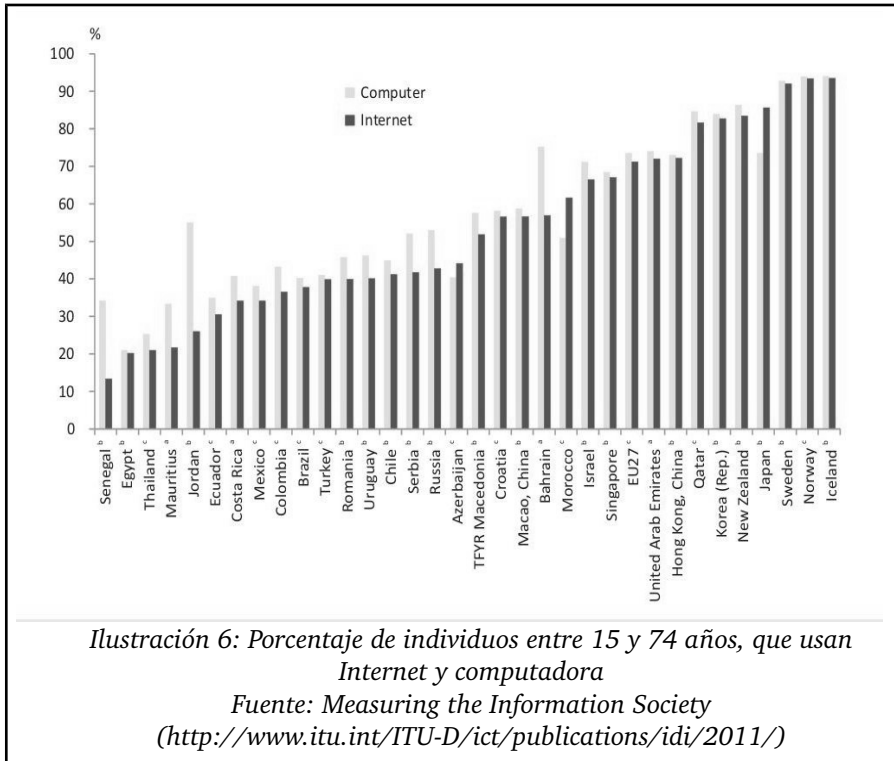
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



1.4 Software libre y abierto en la actualidad (ejemplos y circulación en diferentes países)

El *software* funciona dentro de un ordenador, por lo que es importante determinar la utilización de computadoras para poder hablar del uso de aplicaciones, programas y sistemas operativos. En este sentido, el acceso a Internet y el uso de computadoras, que es de interés para este trabajo, el reporte de ITU (Sampedro Hernández, 2011), mencionado al inicio de este capítulo, muestra los porcentajes de uso y acceso de un grupo seleccionado de países, entre ellos México.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



Como se puede ver en el gráfico de la ilustración no. 6, prácticamente la mitad de los países seleccionados tiene porcentajes de uso de computadoras y acceso a Internet por arriba del 50% de su población. México se encuentra entre aquellos cuyos porcentajes se encuentran en ambos rubros entre el 30 y el 40% de cobertura.

Para utilizar computadoras y entrar a Internet, es necesario utilizar *software*. Del uso que le damos a los ordenadores, depende el tipo de programas y aplicaciones que utilizamos; sin

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

embargo, también nuestro conocimiento sobre la variedad de *software* que existe determina lo que usamos y, con ello, la forma en que nos relacionamos con la máquina.

En 2007 la OECD estimó a nivel mundial el gasto en TIC en 3,433 billones de dólares [...] el 56% se concentra en comunicaciones, 21% en servicios de cómputo, 14% en hardware y 9% en software. Durante el periodo 2003-2007, el crecimiento promedio anual del gasto en software fue de 10%. (Sampedro Hernández, 2011: 38)

En la actualidad, la mayoría de los consumidores de tecnología digital, como el *software*, conocen y utilizan aplicaciones y programas privativos que se comercializan a gran escala. Aunque programas y aplicaciones libres y abiertas existen en igual o mayor cantidad que los privativos, son menos conocidos por la mayoría de los usuarios. Sólo por poner un ejemplo, el sitio Sourceforge.net, que se dedica a apoyar proyectos de fuente abierta (la mayoría, libres y/o abiertos)²⁰ indicaba en su página de Internet que para finales de agosto de 2011 habían ayudado a generar más de 260mil proyectos exitosos a 2.7 millones de desarrolladores.

²⁰ No todo el *software* de fuente abierta; es decir, que muestra su código, es libre, pues puede mostrar el código, pero no permitir la modificación del mismo o bien su libre distribución.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

El directorio de *software* libre de la Fundación de *Software Libre*²¹, en octubre de 2011 registraba 232 aplicaciones de audio, 28 de biología, 369 para negocios, 109 de charla virtual, 2010 de bases de datos, 136 herramientas de documentación, 112 editores, 81 programas educativos, 157 de correo electrónico, 204 juegos, 36 de geografía, 158 de gráficos, 6510 interfaces, 479 aplicaciones de Internet, 56 de localización, 332 de referencias, 654 de lenguajes de programación, 241 de ciencia, 290 de seguridad, 681 de sistemas de administración, entre muchos más. En total, su directorio enlista a 20,312 aplicaciones libres, no sólo de código abierto, sino libres en tanto que permiten mirar el código, modificarlo y mejorarlo, y distribuir el programa. Para ejemplificar, a continuación presento una tabla que muestra algunos de los programas y aplicaciones privativas más utilizadas y algunas de sus versiones libres y abiertas:

Versiones Privativas	Versiones Libres/Abiertas
Distribuciones: Windows Vista, Windows XP, MacOS X	Ubuntu, Trisquel, Blag, Dragora, gNewSense, dynebolic, Musix, Parabola, tuto, Venenux, Fedora, Arch, Puppy <i>Linux</i> , <i>Linux</i> Mint, Debian
Navegadores: Internet Explorer, Safari	Firefox, Opera, Chrome
Suite Office (word, excel, power-point, access)	Open-Office, Libre-Office (mismas aplicaciones)
Estadísticos y modelos: SPSS, Matlab	R, PSPP, Net-Logo, Payek, Scilab

²¹ Ver http://directory.fsf.org/wiki/Main_Page

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Educativos: Blackboard	Moodle
Gráficos: Adobe Illustrator, Adobe Photoshop, AutoCad	Inkscape, Gimp, Archimedes
Video: Flash player	Open Laszlo, Totem
Audio: I-tunes, Windows Media Player	Audacious, Rythmbox, Songbird
Editores: Adobe Acrobat, Microsoft Publisher	PDFCreator, Xpdf, PDFeditor, PDFSam, Scribus
Administradores de Referencias: End-note	Pybliographic, JabRef, Bibus, Mendeley, Zotero
Comunicación: MSN, Skype	Pidgin, Empathy, aMSN, Ekiga, Wengophone
Correo: Microsoft Outlook	Evolution, Thunderbird
Antivirus: Norton, McAfee, Kaspersky	ClamWin, ClamAv, Winpooch

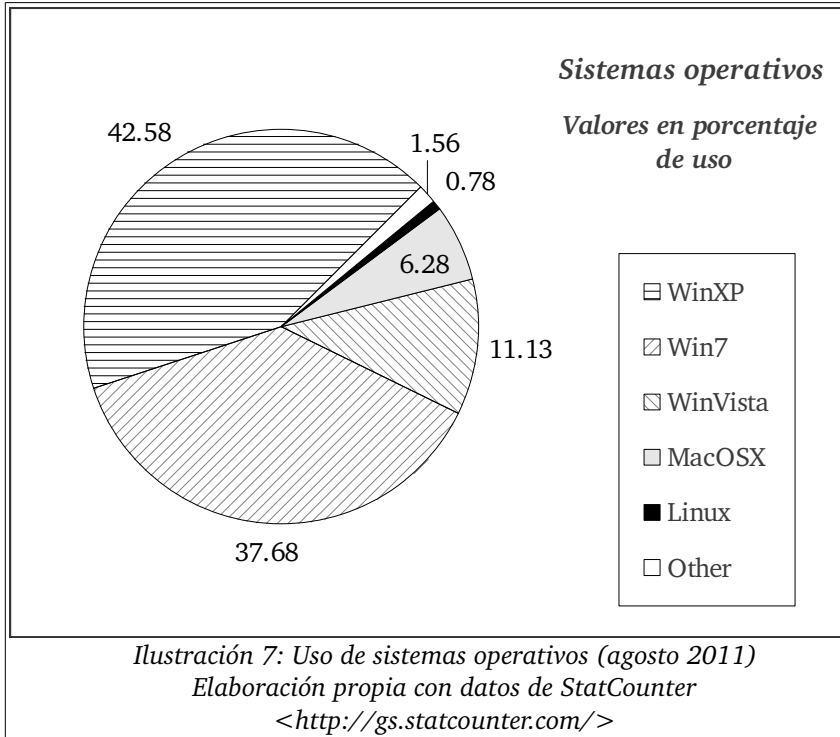
*Cuadro 2: Comparativo entre aplicaciones libres/abiertas y privativas.
Elaboración propia a partir de investigación comparativa*

Datos sobre el uso de *software* privativo en comparación con el libre y abierto los encontramos en el sitio StatCounter²². En el siguiente gráfico, se muestran los porcentajes de uso de sistemas operativos. Se puede ver que los sistemas privativos (*Windows*,

²²<http://gs.statcounter.com> Consultado el 6 de septiembre de 2011. Las estadísticas de este sitio, según los administradores del mismo, “se basan en datos agregados, recolectados por StatCounter de una muestra que excede los 15 billones de visitas por mes a mas de 3 millones de sitios de la red de StatCounter. Sus estadísticas están disponibles cada 4 horas, y se prueban y revisan a los 7 días de su publicación.”

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

en sus distintas versiones, y Mac) cubren el 97.67% de las computadoras registradas en los sitios analizados. *GNU-Linux*, el sistema operativo libre más conocido, sólo se encontró en .78% de las máquinas registradas.



Cabe mencionarse que, según noticias de distintos sitios de Internet, el sistema operativo *GNU-Linux*, por su seguridad y operatividad, es usado por laboratorios, universidades, gobiernos, organismos y organizaciones, como el parlamento

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

francés²³, el parlamento italiano²⁴, el gobierno de Japón²⁵, la bolsa de Nueva York²⁶, el Departamento de Defensa de E.U.A.²⁷, incluso el Colisionador de hadrones²⁸.

En los últimos años ha aumentado el número de empresas que utilizan sistemas operativos libres como Linux, principalmente en servidores, y muchas otras han empezado a utilizarlo en sistemas incorporados. (Sampedro Hernández, 2011: 35)

Este tipo de datos muestra que la razón de que se utilicen más los sistemas operativos privativos que los libres responde a un desconocimiento o conocimiento erróneo generalizado de los últimos. Ahora que si se analiza el uso de navegadores, la perspectiva da un giro:

²³ <http://www.genbeta.com/actualidad/Linux-sigue-expandiendose-el-parlamento-frances-migra-a-ubuntu>

²⁴ <http://www.mastermagazine.info/articulo/12015.php>

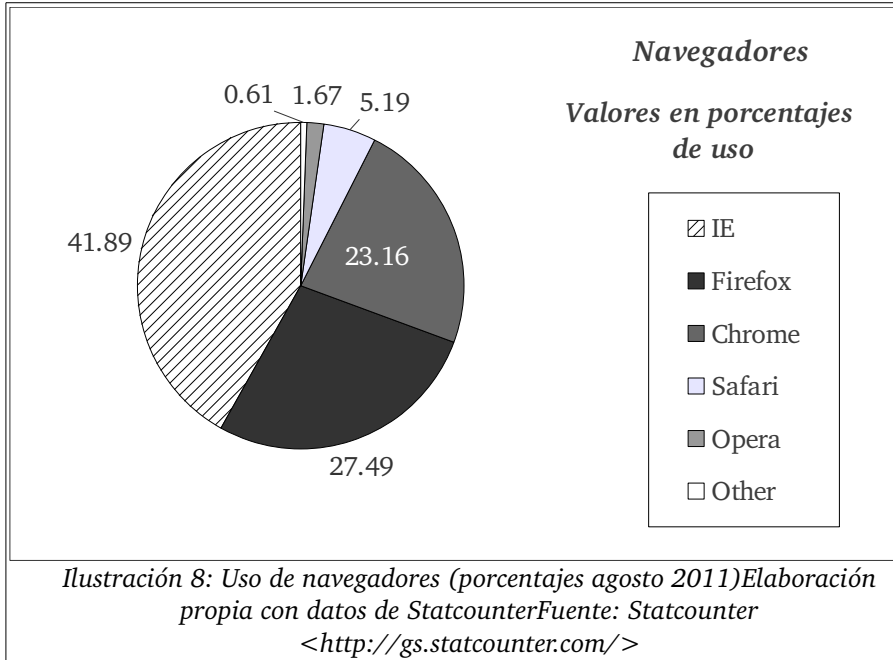
²⁵ <http://www.enriquedans.com/2007/05/Linux-en-japon.html>

²⁶ http://www.theinquirer.es/2007/05/17/la_bolsa_de_nueva_york_migra_a.html

²⁷ http://www.uls.edu.sv/index.php?option=com_content&view=article&id=349:hasta-el-departamenteo-de-defesnad-de-los-estados-unidos-utiliza-Linux&catid=27:new-to-joomla&Itemid=38

²⁸ <http://www.Internetnews.com/skerner/2008/09/large-hadron-collider---powere.html>

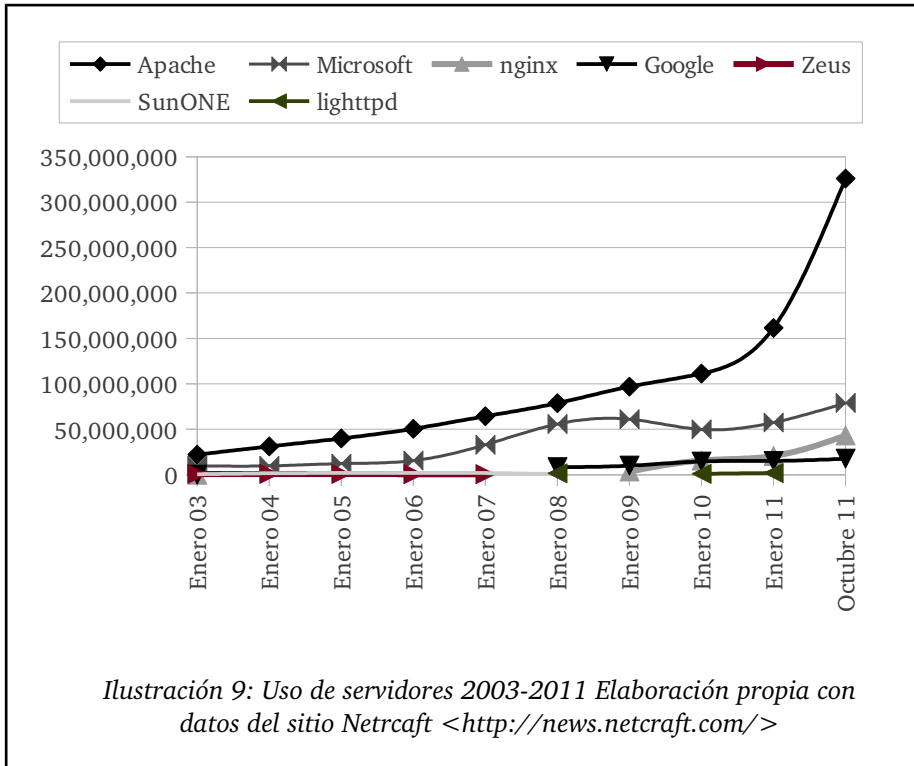
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



Lo que puede verse en el gráfico de la ilustración no.8 es que los navegadores libres y abiertos, como *Firefox* y *Chrome*, son utilizados por el 50.65% de los usuarios, mientras que *Internet Explorer* y *Safari*, que son privativos, cubren sólo al 47.08% *Opera* es un navegador semi-abierto, pues tiene componentes encriptados y otros de libre acceso en su código.

En cuanto al uso de servidores en sitios de Internet activos, registrados por la encuesta permanente del sitio *Netcraft*, el servidor libre *Apache* es utilizado por arriba del de *Microsoft*:

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



La razón del desconocimiento sobre los programas libres, su uso y calidad, tiene que ver con la historia del desarrollo de los ordenadores y el *software*, misma que se ha abordado en el apartado 1.3 de esta tesis. Sin embargo, también existen otros factores que determinan la preferencia de los usuarios por el *software* privativo contra el libre y abierto, como la piratería y una percepción errónea sobre la calidad y seguridad de las aplicaciones libres y abiertas, así como su facilidad de uso y la falta de soporte técnico. En ese sentido, y como se mencionó en el apartado 1.2, esto no se apega a la realidad. Las ideas

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

anteriormente mencionadas se han convertido en parte del imaginario social sobre las computadoras y el uso de las mismas.

En las sociedades occidentales, cuando hablamos de Alta Tecnología tenemos representaciones muy claras y casi podríamos decir homogéneas: *Microsoft* y *Bill Gates*, *Apple* y *Mac*, *Silicon Valley*, *Nintendo* y su *X-Box*, *Sony* y *Play Station*, *Nokia*; hardware y software; *I-phones*, *I-pods*, *Mp3*, *Blu-Ray*, 3a. dimensión. Todo lo anterior, aunado a la imagen de un pequeño grupo selecto de trabajadores super-especializados que ganan muchísimo dinero, trabajando un número reducido de horas al día. Estos trabajadores, además, cuentan gratuitamente con todos los artefactos mencionados para poder hacer 'bien' su trabajo y, en general, se visten como Neo en *The Matrix*.

Otra parte de las imágenes que nos vienen a la mente cuando hablamos de tecnología, específicamente de *software* –sobre todo en México-- se deriva de ideas sobre ilegalidad: piratería, virus informáticos, *hackers*. Cabe mencionarse que *hacker* es un término que en realidad, se refiere a los programadores expertos, aquellos que han sido reconocidos por la propia comunidad cibernauta como los desarrolladores de aplicaciones. El término ha sido mal utilizado y difundido de manera negativa, dando lugar a confusiones y prejuicios.

Los hackers no son lo que los medios de comunicación dicen que son: no son una banda de informáticos locos sin escrúpulos que se dedican a vulnerar los códigos, a penetrar ilegalmente en los sistemas o a desbaratar el tráfico informático. Los

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

que actúan de ese modo reciben el nombre de crackers, y la cultura hacker los rechaza [...]
(Castells, 2001: 56)

Todas las representaciones expuestas implican un marco ideológico que refuerza la idea de competencia individualista, de maximización de ganancias, de propiedad privada y acumulación. Nuestro imaginario está constituido por imágenes del mundo que representan dinero, mercado, capital.

Si bien es cierto que, en general, las aplicaciones libres y abiertas no comprometen recursos económicos a la calidad de las mismas, casi todas las aplicaciones y programas libres y abiertos cuentan con canales de charla interactiva y foros de consulta y retroalimentación. Estos espacios son alimentados y soportados por una gran comunidad de desarrolladores y usuarios expertos que voluntariamente aportan su tiempo y su conocimiento para resolver problemas específicos de otros usuarios. Las comunidades y redes de apoyo se conforman por usuarios de diferentes partes del mundo, por lo que, en general, el soporte que otorgan se encuentra disponible las 24 horas del día, los 365 días del año.

La historia social sobre la emergencia y permanencia del *software* libre y abierto queda incompleta sin una descripción del contexto económico en el que actualmente se está desarrollando esta tecnología. El mercado en el sistema-mundo parece dictar las pautas sobre la propiedad y las formas de producción. El *software* libre y abierto y las comunidades y redes que lo producen, por su forma de organización y protección frente a la

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

privatización o cercamiento han adoptado una vieja forma de mirar los tipos de bienes, clasificando al *software* libre y abierto como un bien común. En el siguiente capítulo, se describirán los argumentos que clasifican a este fenómeno y sus productos como tal, haciendo un recorrido por los tipos de bienes y sus formas de protección, así como la discusión sobre la emergencia de los nuevos bienes comunes de la información y el conocimiento.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Capítulo 2

Tradicionales y nuevos bienes comunes

El conocimiento de la historia sobre la emergencia del *software*, particularmente del *software* libre y abierto, permite comprender la situación actual en la que se encuentra el desarrollo y dinámicas de producción de este tipo de aplicaciones y programas. Las transformaciones del contexto histórico, económico y social configuran las formas de intercambio y producción de bienes y servicios, así como la clasificación de los mismos. En este capítulo describiré el contexto económico en el que se desarrolla la producción y distribución de *software* libre y abierto y la forma en la que éste se inserta en el mercado, a partir de su clasificación como bien común. Brevemente, explicaré la diferencia entre los tipos de bienes para fundamentar que el *software* libre y abierto es un bien común por las prácticas que se desarrollan para su producción y uso. Estas prácticas no son nuevas, ni exclusivas del desarrollo de *software*; tampoco quiero decir en esta tesis que las dinámicas de producción de FLOSS sean contrarias a la organización del trabajo en el desarrollo de *software* libre, incluso pueden ser complementarias. Sin embargo, el acceso al conocimiento de los procesos de desarrollo y las dinámicas de intercambio del *software* libre y abierto sí lo convierten en una forma alternativa de producción al *software* privativo, una más

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

incluyente, que permite el aprendizaje y posibilita la creatividad individual y de grupo.

2.1 Globalización y nuevas tecnologías

En el marco de la perspectiva de los sistemas mundiales (Amin, 1997, 2000; Wallerstein, 2000), la globalización es un fenómeno que se ha desarrollado en el tiempo, a través de varios siglos. Desde los inicios de la humanidad, las sociedades se han entrelazado de distintas maneras para el intercambio de recursos, información, generación de alianzas y supervivencia.

La idea del sistema total significa que todas las redes de interacción humana, pequeñas o grandes, desde los hogares hasta el comercio global, constituyen el sistema-mundo. (Chase-Dunn, 2008)

El sistema mundial está constituido por las instituciones económicas, políticas y culturales, así como sus interacciones y las conexiones que establecen entre sí a lo largo del tiempo. El moderno sistema mundial se enmarca en los procesos de acumulación capitalista.

La historia del cambio social no puede ser entendida sin considerar las estrategias y tecnologías de los ganadores y las estrategias y formas de lucha de aquellos que han resistido a la dominación y explotación. (Chase-Dunn, 2008)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Lo que hoy llamamos “globalización” es una parte de un largo proceso histórico de la humanidad. Se refiere a un entrelazamiento mundial a través del comercio, los flujos financieros y las inversiones de capital desde y hacia diferentes países que, en la actualidad, se ha intensificado y expandido. Es una integración económica fundamentalmente, cuya base principal es el liberalismo del mercado o, como se le nombra en la actualidad, el neoliberalismo.

La expansión de la integración económica global se ha posibilitado de manera más eficiente por las nuevas tecnologías, específicamente las tecnologías de comunicación e información que, hoy por hoy, mantienen los flujos del mercado en movimiento constante. Como dice Castells (2005):

La revolución de la tecnología de la información ha sido útil para llevar a cabo un proceso fundamental de reestructuración del sistema capitalista a partir de la década de los ochenta. (p.9)

El nuevo paradigma tecnoeconómico que se basa en la microelectrónica y se caracteriza por el uso intensivo de la información posibilita el desarrollo de procesos productivos, supuestamente, más limpios que producen un menor impacto ambiental.

Ya que en la actualidad la preocupación por el ambiente y el desarrollo sustentable se ha hecho un imperativo, de manera global se piensa en cambiar hacia modelos productivos que se basen más en el uso intensivo del conocimiento que en el de materias primas y energía. Los países desarrollados marcan una

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

tendencia a aumentar la complejidad de la actividad tecnoproductiva y a fortalecer los espacios de política científica y tecnológica, impulsando una creciente y costosa actividad de investigación y desarrollo.

La base estructural del auge del proyecto globalizador es el nuevo nivel de integración alcanzado por la clase capitalista global y la crisis de la sobre-acumulación que emergió en los 70, cuando Alemania y Japón alcanzaron a Estados Unidos en la producción de artículos de alta tecnología. (Chase-Dunn, 2008)

Sin embargo, no hay que dejar de tomar en cuenta que no todas las sociedades cuentan con el desarrollo tecnológico y el conocimiento y habilidades técnicas para insertarse en el mercado global de manera satisfactoria.

Como la globalización, la nueva tecnología también produce desigualdad: no todos pueden costearse los nuevos dispositivos tecnológicos. En muchos países de África y Asia, por ejemplo, los bajos niveles de alfabetización y los casi inexistentes vínculos entre la gente ordinaria hacen burla de las visiones más utópicas sobre el poder global del ciberespacio (Paasi, 2002).

Castells (2005) explica históricamente las desigualdades en la adopción de las tecnologías informacionales y sus consecuencias, a partir del análisis socio-político del desarrollo y la

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

transformación capitalista. Como dice en “La Era de la Información” (2005):

La revolución de la tecnología de la información ha sido útil para llevar a cabo un proceso fundamental de reestructuración del sistema capitalista a partir de la década de los ochenta (p.9) Aunque la reestructuración del capitalismo y la difusión del informacionalismo fueron procesos inseparables, a escala global, las sociedades actuaron/reaccionaron de forma diferente ante ellos, según la especificidad de su historia, cultura e instituciones.(p.14) En este contexto, la globalización no supone el impacto nivelador de los procesos universales sino, por el contrario, la síntesis calculada de la diversidad cultural en la forma de lógicas y capacidades de innovación regionales diferenciada. (p. 30)

En este sentido, es importante lograr el mayor acceso a las tecnologías de información y comunicación, así como al conocimiento básico para su utilización. Internet, hasta ahora, es un espacio común de prácticamente libre acceso; los contenidos a los que se puede acceder, como los que pueden integrarse a la red, son casi ilimitados. El espacio virtual se ha convertido en un espacio donde la comunidad global puede intercambiar opiniones, información, música, videos y programas computacionales. Todo lo cual aporta al conocimiento y las habilidades de los usuarios que libremente pueden elegir la forma en la que se relacionan, tanto en ese ambiente como, con la plataforma en sí. Sin embargo, esto podría dejar de ser así.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Fuerzas poderosas están usando rápidamente la ley y la tecnología para “domesticar” Internet. La innovación, otra vez, será dirigida desde arriba hacia abajo, aumentando el control por los dueños de las redes, los poseedores de los más grandes portafolios de patentes y, más injustamente, acaparadores de derechos de autor. (Lawrence Lessig, 2001)

Como política pública, para disminuir la brecha digital, se hace necesario fomentar el conocimiento tecnológico y las habilidades técnicas de la población en general, específicamente, de jóvenes. Asimismo, fomentar la innovación tecnológica en cada país para así disminuir la brecha digital que el desarrollo vertiginoso de la tecnología, particularmente las de información y comunicación, está generando. Tal y como lo decía en 2003 el, entonces, senador Javier Corral:

Estamos en el proceso más importante de convergencias tecnológicas: microelectrónica, telecomunicaciones, radiodifusión, multimedios y tecnologías de la información y la comunicación. Este es un proceso que genera nuevos productos y servicios, así como nuevas formas de gestionar la economía, hacer negocios, nuevas relaciones comerciales, etcétera. Pero los efectos no son sólo de naturaleza económica y no pueden estar enmarcados en los intereses del mercado; más allá de las relaciones comerciales, el impacto social es indiscutible y la política debe atenderlo en forma

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

inmediata. (Solís Leree, 2003: 11)

Estas palabras fueron pronunciadas en la sexta edición de la “Conferencia de Mayo” en el Congreso de la Unión, espacios de discusión encaminados a preparar la participación de México en la Cumbre Mundial de la Sociedad de la Información en diciembre de 2003. En estas discusiones, senadores y académicos abordaron temas sobre inclusión digital, educación con apoyo de las nuevas tecnologías, derecho al acceso a la información y la comunicación, apoyo a la innovación nacional; las preocupaciones giraban en torno a lograr disminuir la brecha digital, en principio, dentro del país. Fundamentalmente, era necesario, en palabras del Dr. Trejo Delarbre:

1. Ampliar la cobertura, pero también la calidad de las conexiones a la internet; 2. Desarrollar y extender la educación para el uso de la Red y del conjunto de nuevas tecnologías de la información y la comunicación; 3. Impulsar contenidos nacionales para la Red; 4. Defender la libertad de expresión y la privacidad en internet. (Solís Leree, 2003: 101-102)

Entre las propuestas, que incluyeron la ampliación de la cobertura, la revisión y re-estructuración del programa e-México, la inclusión de contenidos de interés nacional en la red, entre otros, destaco:

El profesor Robinson cuestionó a los programas gubernamentales que en vez de promover la innovación tecnológica, como la que constituyen los

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

programas de código abierto, protegen a los mercados y proveedores ya existentes. La periodista Burch propuso alentar el desarrollo de software nacional y promoción del software libre. El maestro Jorge Lizama, del Posgrado en Ciencias Sociales de la UNAM, habló de la pertinencia de crear soportes de información, o software, que sean libres, regidos por formatos estándar y abiertos. Enzo Molino también habló de las posibilidades del software libre. (Trejo Delarbre, 2003)

La cita indica la importancia de buscar acabar con la dependencia tecnológica, tanto para fomentar la innovación en el país, como para alentar una relación distinta con las máquinas al usar programas y aplicaciones de código abierto. De la autonomía en la producción, uso y distribución de *software* puede depender el futuro del acceso al conocimiento y a la información, los cuales son bienes comunes que deben permanecer abiertos para la sociedad en su conjunto.

A lo largo de tres días de debate y exposición de perspectivas, los participantes de esta conferencia concluyeron con definiciones en común, identificación de problemas y retos a vencer. Incluso, como se describe en la relatoría:

El secretario de Relaciones Exteriores, Luis Ernesto Derbez, había subrayado en la inauguración. “Nuestro país estima que los gobiernos deben actuar en esta materia como agentes promotores del cambio, instrumentando políticas públicas que

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

posibiliten la interacción social, nuevas prácticas económicas y comerciales, así como para extender con equidad, servicios de vanguardia a la población, con transparencia y [sin] discriminación.”(Solís Leree, 2003: 273)

A ocho años de pronunciadas esas palabras, no contamos en México con una política pública que garantice el acceso de la población general a internet, mucho menos que los contenidos en la red sean de interés nacional. Tampoco contamos con apoyos consistentes a la investigación, desarrollo e innovación tecnológica.

En el caso de la industria de *software* en México, existe el Programa para el Desarrollo de la Industria de *software* (Prosoft) que busca fomentar el desarrollo y competitividad de empresas e industrias de *software* a nivel internacional. La tarea del programa es otorgar financiamientos, junto con Estados, empresas y academia. Sin embargo, el programa no permite mediciones de impacto, por lo que no puede aún medirse la efectividad del mismo.

[... la inversión total] se concentró en [2006] en proyectos productivos (30%), innovación y desarrollo tecnológico (25%), fortalecimiento de la capacidad regional y empresarial (12%), capital humano (10%), y otros (22%). (Sampedro Hernández, 2011: 51)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

La cita anterior sería esperanzadora si no se tuviera el dato de que en el sector de desarrollo de *software*, el gasto, respecto al PIB, *representa apenas el 0.1% (SE, 2005), esta participación es 6 veces inferior al promedio mundial y 9 veces menor que el de Estados Unidos (SE, 2006). (Sampedro Hernández, 2011: 41)*

Sin investigación local, no puede haber desarrollo interno real, lo que implica aún mayor dependencia al exterior en cuanto a tecnología desarrollada para los diferentes sectores de la economía de los países.

2.2 Diferencia entre bienes públicos, privados y comunes

No existe sólo una definición de bien común. David Bollier (2002), escritor y activista norteamericano, lo define como una “infraestructura social de instituciones culturales, reglas, tradiciones, y cuyos recursos están restringidos al uso personal (no mercantil) de los miembros de la comunidad.” Charlotte Hess, economista norteamericana fundadora de la Biblioteca Digital sobre los Bienes Comunes, describe a éstos como “un recurso compartido por un grupo, donde el recurso es vulnerable a la apropiación/privatización, sobre-uso y conflicto social. A diferencia de un bien público, requiere administración y protección para sostenerlo.”(Hess, 2008: 38)

De una manera simplificada, se puede decir que un bien común es un recurso que se comparte, pero también es el proceso que lo produce y la dimensión social que tiene, en tanto su función en, y para, la sociedad. “Lo que brinda el elemento social elemental de los bienes comunes”(Walljasper, 2010).

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

A diferencia del mercado, que se apoya en los precios como la única dimensión del valor, un bien común se organiza alrededor de una mezcla más rica de necesidades humanas –para la identidad, comunidad, fama y honor-- que son indivisibles e inalienables, así como de recompensas mas 'tangibles'. (Clippinger & Bollier, 2005)

Desde tiempos de la antigua Roma, existe la división entre bienes privados, públicos y comunes. El derecho romano dividía la propiedad de las cosas en *res privatae*, que designaba las cosas que los individuos podían poseer; *res publicae*, que definía todo aquello que era administrado por las organizaciones del Estado, y *res communes*, como aquellas cosas que pertenecían a todos. (Berry, 2005; Rose, 2003)

Sin embargo, es necesario entender de qué se habla cuando se define el tipo de derechos de propiedad, pues éstos varían según el tipo de bien. Los derechos sobre los bienes se pueden ver según su calidad o su carácter. La calidad define la constitución básica de los bienes (materiales o intelectuales); el carácter es definido por lo que la sociedad misma define como valor atribuible al bien y que, en su contexto, lo constituye como distinto de otro bien (creadores, titulares de los derechos, acceso, alcance, disponibilidad, etcétera).

En cuanto a su calidad, esto es si son bienes materiales o intelectuales. Como bienes materiales, su carácter puede ser: privados, cuando tienen un dueño; públicos, cuando el titular de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

los derechos de propiedad es el Estado; o comunes, cuando, como indican Vercelli y Thomas (2008)

[...] pertenecen y responden al interés de todos y cada uno de los integrantes de una comunidad. Son bienes que redundan en beneficio o perjuicio de todos y cada uno de estos miembros o ciudadanos por su condición de tal. [...] El carácter común puede surgir 'por defecto' de la legislación o 'por contrato' a través de la autonomía de la voluntad. (pp. 2 y 10)

En su calidad de bienes intelectuales, la distinción entre privados, públicos o comunes resulta inadecuada, pues el producto de un proceso creativo o el pensamiento mismo no es susceptible de ser apropiado. Al mismo tiempo, el uso no agota este tipo de bienes, mas bien, al contrario: una idea se complejiza y complementa, mientras más gente puede pensarla.

El carácter típico de los derechos sobre bienes intelectuales es la capacidad de inclusión y la búsqueda de un balance / equilibrio entre los creadores de obras intelectuales y los intereses de las comunidades. Bajo determinadas condiciones, a diferencia de la propiedad, el sistema de los derechos intelectuales tiende a la distribución de los bienes intelectuales. (Vercelli & Thomas, 2008: 8)

En este sentido, el carácter de este tipo de bienes atiende a derechos intelectuales, no de propiedad, y éstos pueden ser:

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

privativo, cuando su autor elige una legislación que priva del derecho de uso y reproducción de la obra a un sector de la población; público, cuando se produce por un Estado y su forma de utilización como privativa o común quedará determinada por la legislación aplicada; o común, cuando es una comunidad la que detenta los derechos sobre los bienes, y ésta define que cualquier integrante de la comunidad puede acceder a los mismos.²⁹

[...] una obra intelectual es común cuando ofrece a los miembros de una comunidad de forma directa, inmediata y sin necesidad de solicitar permiso, la posibilidad de acceder, usar, reproducir, ejecutar, distribuir, estudiar y transportar la obra hacia diferentes soportes de acuerdo con las formas y condiciones establecidas. Las obras que respetan estas condiciones también se las describe como de carácter abierto. Cuando, además, se permite usar la obra con cualquier finalidad, adaptarla y derivarla y, sobre todo, cuando la comunidad indica que sus obras derivadas deben regularse bajo las mismas condiciones, se considera que las obras además de comunes tienen un carácter libre. (Stallman, en Vercelli & Thomas, 2008: 11)

²⁹Ver cuadro 1: “Clasificación de los Bienes”

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Bienes	Calidad	Legislación aplicable	Carácter
	Material es	Derechos Materiales: Propiedad/Dominio	Privado
			Público
			Común (Apropiable-No Apropiable)
Intelectuales (y obras)	Derechos Intelectuales: Derecho de Autor; Patentes; Marcas; Diseños Industriales; Conocimientos tradicionales; etc.	Privativo ³⁰	
		Público	
		Común (Abierto-Libre)	

*Cuadro 3 : "Clasificación de los bienes"
Fuente: Vercelli & Thomas, 2008: 12*

Es importante hacer la distinción de los bienes comunes con los regímenes de libre acceso. Los primeros son propiedad de un grupo o comunidad. Cuentan con reglas, están montados sobre instituciones culturales y su uso se restringe a los miembros del grupo o comunidad que los producen y/o mantienen, o a lo que decidan los mismos—como es el caso del *software* libre. Los regímenes de libre acceso, en cambio, son aquellos en los que “nadie tiene el derecho legal de excluir a otro del uso de un recurso.”(Ostrom, 2000: 335)

³⁰ El término “privativo” ha sido retomado por la comunidad de *software* libre y abierto con referencia a todas aquellas aplicaciones que no otorgan la libertad de acceso al código fuente, principalmente. “Te privan de la libertad de acceso, uso y mejora”. Antes se utilizaba el término “propietario”, que no era claro para describir esta privación de la libertad.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

La noción de 'bienes comunes' es una construcción / definición técnica que permite, a su vez, incluir, englobar y explicar otros conceptos: comunes, riqueza común, activos comunes, recursos comunes, propiedad común, bienes comunitarios, propiedad comunitaria, patrimonio común. (Vercelli & Thomas, 2008: 2)

Si una comunidad o grupo genera y utiliza sus propios recursos, los miembros no tienen necesidad de adquirirlos afuera. Los productos que genera el grupo o la comunidad, a partir del trabajo colectivo, son propiedad de todos los miembros de la misma. Por tanto, cada uno puede hacer uso de estos recursos.

... la producción 'basada en bienes comunes' es cuando nadie tiene derechos exclusivos para organizar el esfuerzo o apropiarse de su valor, y cuando la cooperación se logra a través de mecanismos sociales más que en señales de precios o instrucciones administrativas. Instancias de larga escala de este tipo de cooperación es la producción entre pares. (Benkler, 2004, citado en Hess, 2008: 36)

2.3 Breve historia de los bienes comunes

La historia de los bienes comunes se remonta a los tiempos anteriores a la Revolución Industrial. El antecedente inmediato de este periodo es la época de la manufactura, cuyo sistema de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

organización intermedio entre el taller artesanal de la Edad Media y la fábrica moderna, se basó en la división del trabajo a través de los gremios no mayores de 20 personas y una jerarquía entre oficiales y aprendices. El uso de las máquinas que se prefiguró en la manufactura dependió de la consolidación de ésta como sistema fabril dominante.

Según Hobsbawm (2004), el tránsito del capitalismo mercantil al industrial fue un proceso lleno de contradicciones, en el que las soluciones coyunturales se convertían muchas veces en factores que retrasaban el desarrollo posterior de las relaciones capitalistas. El primer límite estaba dado por la estructura feudal de la sociedad, que entre otras cosas fijaba la fuerza de trabajo a la tierra. En segundo lugar, aun cuando las innovaciones técnicas estuvieran disponibles era necesaria una expansión del mercado para la utilización del nuevo potencial de la producción industrial.

El proceso de urbanización que construyó el contexto general para la manufactura dependió, dice Maurice Dobb(1996), de los cercamientos impuestos por los propietarios de tierras que produjeron el desplazamiento de la población del campo a las ciudades para obtener trabajo en las fábricas como mano de obra. En 1801 se decreta la Ley General de Cercamientos en Inglaterra que pretendía incrementar la productividad de las tierras convirtiéndolas en tierras de pastoreo.

Los cercados han sido llamados apropiadamente una revolución de los ricos contra los pobres. [...] Literalmente estaban robando al pobre, de su parte

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

en el bien común, derribando las casas que, por la hasta entonces inquebrantable fuerza de la costumbre, los pobres habían, durante largo tiempo, considerado como suyas y de sus herederos (Polanyi, 2004).

Según Ashton(1965), muchos labradores ya habían vendido sus tierras antes de que se decretara la ley, y que muchos otros aprovecharon sus ahorros obtenidos en tiempos de guerra para comprarlas. Sin embargo, se dieron muchos favoritismos y privilegios que resultaron en injusticias que desposeían a los antiguos agricultores para otorgar sus tierras a nuevos burgueses.

[...] por lo general ignoraron las solicitudes fundadas tan sólo en la equidad, de aquellos cuyo único título era la posesión proveniente tal vez de sus abuelos, y que habían labrado la tierra hasta entonces sin disputas.(Ashton, 1965: 32)

La creencia en la propiedad privada, por encima de la propiedad comunal, para ordenar la sociedad tiene sus raíces aún más atrás en el pensamiento de John Locke, quien establece el derecho a la propiedad privada como inherente del hombre.

...cada hombre tiene la propiedad de su propia persona...el esfuerzo de su cuerpo y la obra de sus manos son también auténticamente suyos... siempre que alguien saca alguna cosa del estado en que la Naturaleza lo produjo y la dejó, ha puesto en esa

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

cosa algo de su esfuerzo... la ha convertido en propiedad suya. (Locke, 2005)

Siguiendo a Fernández Santillán(1996), es, de hecho, a partir de este concepto de propiedad que su propuesta avanza hacia la constitución de la sociedad civil. Ésta se inaugura por consenso para salvaguardar, primordialmente, las libertades y propiedades del hombre.

Un sistema de creencias político y legal que vio el origen de la propiedad misma en los esfuerzos de propietarios individuales de ocupar la tierra dio a propietarios de tierras un papel especial en la sociedad. Estas creencias ayudaron a justificar el paso de una legislación para eliminar los derechos colectivos de propiedad de la tierra y autorizar los cercamientos y la apropiación de la propiedad comunal por propietarios individuales. (Ostrom, 2000: 2)

Un famoso artículo³¹ de Hardin, titulado “La Tragedia de los Bienes Comunes”³² llevó a la racionalidad económica a utilizar

³¹ El supuesto de la “Tragedia de los bienes comunes” es que los recursos sólo pueden estar regulados por el Estado o privatizados, pues de no ser así, la explotación de agentes que buscan su beneficio individual, lleva al agotamiento de los recursos. Esta premisa correspondía a los supuestos teóricos de la economía neoclásica, cuya influencia, aún hoy, es hegemónica en todo el mundo.

³² “A partir de un ejemplo de pastores usando un prado en común, Hardin describe cómo un recurso escaso abierto a cualquiera es mermado y dejado

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

la metáfora para justificar la privatización de los bienes para, supuestamente, asegurar la sustentabilidad y supervivencia de los mismos. Sin embargo, la sobre-explotación y destrucción de recursos descrita en el artículo refiere más a los regímenes de libre acceso que a los de bienes comunes. (Bollier, 2002)

En un artículo clásico, Ciriacy-Wantrup y Bishop (1975) claramente indican la diferencia entre regímenes de propiedad que son de libre acceso, en los que nadie tiene el derecho legal de excluir a cualquiera de usar un recurso, de la propiedad común, donde los miembros de un grupo claramente demarcado tienen el derecho legal de excluir a no-miembros del grupo del uso de un recurso. (Ostrom, 2000: 335)

2.4 Nuevos bienes comunes

El término bienes comunes está volviendo a utilizarse desde la década de las últimas décadas del siglo XX, especialmente a partir de la emergencia de los movimientos ecologistas en los 90, que buscan preservar los recursos naturales. Sin embargo, existe una nueva clasificación de bienes comunes, susceptibles también a la privatización, que se están estudiando desde distintas perspectivas.

en ruinas. Los 'bienes comunes' decaen porque cada pastor obtiene beneficios directos de sobre-explotar el bien, mientras sufre sólo costos indirectos. Eventualmente, la sobre-explotación destruye el recurso.“ (Bollier, 2002) (La traducción es propia)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

La comunidad jurídica ha invocado de manera creciente los “bienes comunes” como argumento en contra de la expansión de los derechos de propiedad intelectual y el incremento de las ambigüedades legales en el advenimiento del ambiente digital en-línea. (Hess, 2008: 2)

La clasificación de estos nuevos bienes comunes es difícil, puesto que en la literatura sobre los mismos no se comparte una definición de bien común. Éstos se describen como *recursos, movimientos o fenómenos, aunque tienen en común el sentido de “compartir” y de propiedad conjunta. (Hess, 2008: 6)*

Hay muchas maneras diferentes en que los nuevos bienes comunes se desarrollan o nacen. Algunos evolucionan de las nuevas tecnologías, que han permitido la captura de lo que previamente eran bienes públicos incapturables, como Internet, información genética, el espacio exterior, los mares profundos, y el espectro electromagnético. (Hess, 2008: 4)

Los nuevos bienes comunes engloban bienes culturales, médicos y de la salud, de vecindad, de infraestructura, globales, mercados y del conocimiento. Todos estos son parte de infraestructuras culturales, políticas, incluso económicas, construidas socialmente en beneficio de las propias comunidades que las generaron. Estos bienes son susceptibles de 'cercamiento', siguiendo una lógica de comercialización,

privatización o 'protección' para evitar, en su caso, su sobre-explotación.

2.4.1 Nuevos bienes comunes de la información y el conocimiento

El conocimiento, como se ha expuesto, es un bien común, pero tradicionalmente no se le clasificaba como tal. La categorización de los bienes, en general, se hacía sobre aquellos bienes tangibles, como la tierra o el agua. Actualmente, el conocimiento, en tanto que los productos de éste pueden ser privatizados o enajenados, se retoma en la discusión como un bien que debe ser protegido.

[...] es simplemente un redescubrimiento de los fundamentos sociales que siempre han soportado a la ciencia, la investigación académica y la creatividad. La comunidad científica ha honrado siempre el intercambio de conocimiento y recursos, el diálogo abierto y las sanciones en contra de la investigación fraudulenta. Por años, la academia ha florecido con la misma ética de intercambio y apertura entre miembros de una comunidad auto-gobernada. (Bollier en Ostrom, et al., 2007: 36)

Se considera el conocimiento como un bien común pues, siguiendo a Polanyi *adquirir y descubrir el conocimiento es un proceso, tanto social como, profundamente personal.* (Polanyi en Elinor Ostrom et al., 2007), pero, además, porque aquello que puede conocerse, no puede privatizarse; es decir, apropiarse,

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

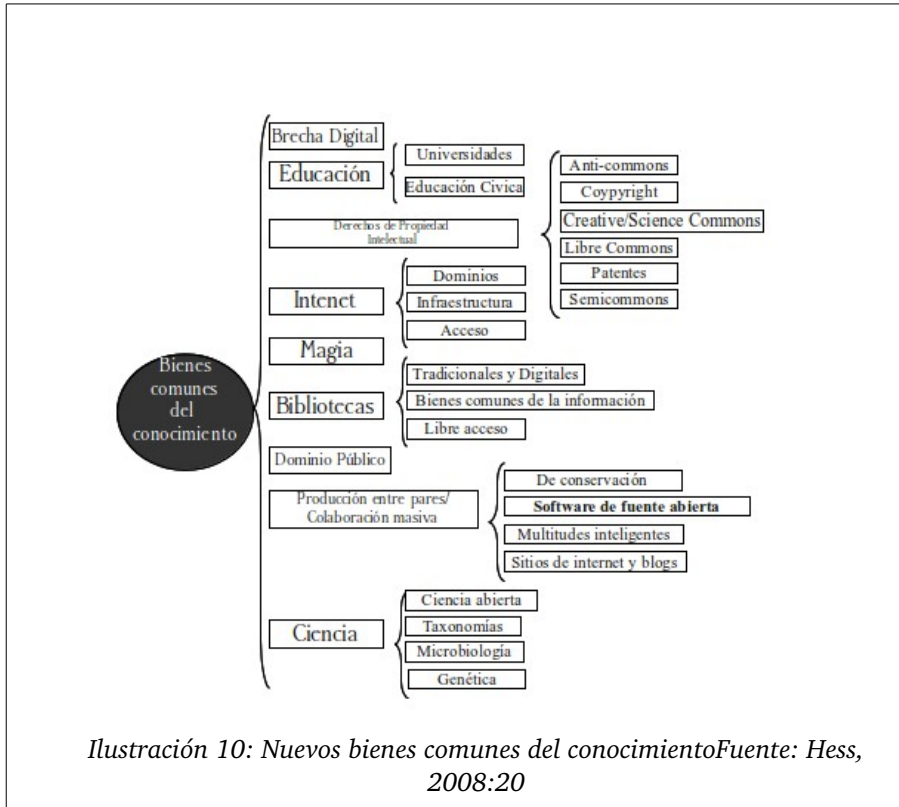
excluyendo a otros de conocerlo. El conocimiento no es un recurso agotable por la difusión y el uso.

De hecho, mientras más gente comparta conocimiento útil, mejor es el bien común. Considerar el conocimiento como bien común, entonces, sugiere que el lazo que vincula es que se usan y administran de manera colectiva por grupos de distintos tamaños e intereses. (Ostrom et al., 2007: 20)

En tanto seres humanos, todos somos capaces de pensar y conocer (en este sentido, se habla aquí de conocimiento), y, como sociedad, históricamente hemos construido juntos el conocimiento. *Sin sociedad, la cognición no es posible.* (Obregón, 2002: 48)

El dominio público es fundamental para la evolución cultural y artística de la comunidad. Nadie puede crear en un «vacío»: toda creación se sustenta de creaciones anteriores. (Xalabander Plantada, 2006)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



El *software*, así como las bibliotecas y los avances científicos, están amenazados no por su sobreuso, sino por el subuso, debido a las restricciones otorgadas por las leyes de copyright y patentes. Como indica Elinor Ostrom, premio Nobel de Economía (2007):

Actualmente, existe una enorme serie de amenazas

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

de privatización de la información y el conocimiento –incluyendo código computacional en leyes (Lessig 1999) y nuevas legislaciones de propiedad intelectual ([...])-- que mina el libre acceso a información pública, científica y del gobierno. (p.13)

Este tipo de bienes no se degradan por el uso excesivo, pero pueden perder su utilidad si no llegan a la sociedad. La tragedia, en el caso de los bienes comunes de la información y el conocimiento, sería que por leyes que protegen más al mercado y a las ganancias empresariales, se sustraigan importantes beneficios para la población del mundo en general.

2.4.2 Aspectos económicos, políticos y culturales de los nuevos bienes comunes del conocimiento y la información

La tecnología es social; es decir, el desarrollo técnico responde a necesidades sociales. Son los hombres y mujeres de la sociedad los que crean y desarrollan tecnología, y ésta emerge del modo en que lo hace porque hombres y mujeres que la producen están insertos en un contexto social determinado que les permite percibir, construir y pensar de determinadas formas.

En este sentido, el funcionamiento o no-funcionamiento de un artefacto, la plenitud de un derecho o la efectividad de un bien común no responden a una propiedad intrínseca. Por el contrario, son el resultado de un complejo proceso de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

construcción socio-técnica en el que se evidencian las negociaciones e imposiciones de significados atribuidos por los diferentes grupos sociales. (Vercelli & Thomas, 2008: 4)

El que un producto tecnológico se desarrolle de manera colectiva y busque compartirse del mismo modo; o bien, si ésta emerge en un laboratorio cerrado con el fin de la comercialización y la ganancia individual, responde a contextos sociales específicos. Por eso, Vercelli & Thomas (2008) indican que:

[...] la acción distribuida de los diferentes grupos sociales diseña, produce, hace funcionar las tecnologías y, en el mismo acto / momento, también regula los espacios y las conductas del entorno socio-técnico común. (Vercelli & Thomas, 2008: 4)

La tecnología y la sociedad se mueven en una relación dialéctica. Hay estructuras que constriñen y/o posibilitan la creación de nuevos artefactos tecnológicos. Son los individuos que, en su interacción con las estructuras, las transforman, abriendo el camino a nuevas posibilidades de creación y producción técnica.

Los productos tecnológicos, una vez generados y reproducidos socialmente, interactúan también con las estructuras y los individuos, generando nuevas necesidades y haciendo posible, también, nuevas formas de entender el mundo. Esta idea, como se explicó ya, es muy cercana a la idea de paradigma de Kuhn.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En la esfera del desarrollo tecnológico, la innovación y el desarrollo de determinados productos, así como el uso y regulación de éstos, responde a luchas entre los mismos grupos que producen y usan las tecnologías. Esto puede verse, tanto en el uso de ciertos productos tecnológicos sobre otros, como en los conflictos emanados de la sobre-especialización técnica (p.e. la separación de programadores e ingenieros en el desarrollo de hardware y *software*). (Ver Jesiek, 2006)

[...] el carácter común de algunos bienes es parte de duras luchas y negociaciones. Son los grupos de actores, las comunidades, los ciudadanos los que van construyéndolos a lo largo del tiempo. (Vercelli & Thomas, 2008: 14)

En el caso de los nuevos bienes comunes del conocimiento y la información, la lucha se da entre las comunidades productoras que creen que estos bienes no deben ser enajenados contra los intereses del mercado. Existe la idea de que los derechos de propiedad sobre bienes intelectuales motivan a los creadores y científicos, como se ha dicho, a crear y compartir sus descubrimientos o productos, pues la protección de su obra, según argumentan, les permite obtener un beneficio económico al ser replicada o utilizada por otros. Sin embargo, este tipo de argumentos, según otras posturas, son resultado de una forma de organización social que busca premiar el individualismo, más que el bienestar social. Como dice Stallman (2004):

Si hay algo que merezca una recompensa es la contribución social. La creatividad puede

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

considerarse una contribución social, pero sólo si la sociedad es libre de aprovechar sus resultados. Si los programadores merecen ser recompensados por la creación de programas innovadores, bajo esta misma lógica deben ser castigados si restringen el uso de estos programas. (p. 53)

2.4.2.1 Industrias culturales e industrias creativas

Actualmente, se están utilizando nuevos términos para describir al tipo de industrias que producen en base a la creatividad y la cultura de individuos y comunidades. Son términos distintos, aunque aún no bien delimitados el uno del otro.

[...] una industria cultural es cualquier entidad con la misión de crear, producir y distribuir eventos, bienes, productos y servicios culturales. (Ciuro, 2009)

Cuando se habla de industria cultural, en general, se toman en cuenta actividades dentro del campo editorial, producción multimedia, audiovisual, fonográfica, cinematográfica, artesanal y de diseño; quedando fuera otro tipo de actividades que suponen procesos creativos como la producción de algunos bienes o servicios. Estas actividades de industrias creativas pueden dividirse en: antigüedades; arquitectura; artesanía; cine, video y fotografía; diseño de modas; diseño gráfico y de interiores; edición; música, artes visuales y escénicas;

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

publicidad; *software*, videojuegos y edición electrónica; televisión y radio.

Con las tecnologías de información y comunicación, muchos productos de industrias creativas y culturales, como el *software*, pueden fácilmente difundirse a través de internet. Asimismo, artesanías y otros productos de la cultura se comercializan ya de manera global. La clasificación en industrias culturales busca proteger estas expresiones, generando a su vez beneficios para las comunidades productoras.

Los productos emanados de este tipo de actividades se protegen por las leyes de derechos de autor, buscando la remuneración económica de los creadores y, con ello, el crecimiento económico del país generador. Sin embargo, las mismas comunidades creadoras, en algunos casos, están utilizando métodos de protección alternativos, como licencias *creative commons*³³ o libres.

[...] un modelo de negocio basado en licencias Creative Commons no viola la propiedad intelectual de ningún artista, sino que ofrece una alternativa para aquellos autores que quieran licenciar sus contenidos de forma más libre, permitiendo la reutilización de sus creaciones en obras ajenas. (Ciuro, 2009)

³³ Las licencias Creative Commons manejan distintos niveles de protección de los derechos de autor. Más adelante se explican las diferencias.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Algunas publicaciones científicas³⁴ y la producción de *software* libre y abierto son parte de estas industrias creativas que también son bienes comunes. Las comunidades y redes que producen y protegen el conocimiento científico y FLOSS son muy parecidas, en tanto que siguen una ética de intercambio y revisión entre pares; de hecho, están concientes que sólo de esa manera pueden desarrollar su propio trabajo, llevando a la ciencia y a la tecnología hacia los descubrimientos y la innovación. Los productos de la ciencia y la ingeniería necesitan de difusión para llegar a su aplicación social; muchas veces, la comercialización apoya esta difusión, pero otras veces lo obstaculiza. En este sentido, el interés económico en cada uno de estos campos debería quedar superado por el interés científico y social.

2.4.2.2 Protección legal: patentes y licencias

En cuanto a protección legal de obras y productos tecnológicos, hay que diferenciar entre licencias y patentes. Las primeras protegen obras o productos concretos; las segundas, en diferentes grados, buscan proteger ideas o inventos.

El copyright regula las condiciones de expresión de una obra, no protege ninguna idea. Las patentes sólo protegen las ideas y el uso de las ideas [...] La patente es un monopolio absoluto sobre el uso de una idea. Incluso si pudieras probar que la idea es

³⁴ El movimiento, denominado Open Access, relativamente nuevo en la Academia, busca la apertura de las publicaciones científicas. Más información en <http://www.doaj.org/>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

tuya, sería completamente irrelevante si la idea ha sido patentada por otro. (Stallman, 2004: 134 y 135)

Las patentes protegen inventos que caben dentro de lo que se denomina propiedad industrial. En su origen, su función era fomentar la invención y la difusión de la misma, otorgando, por parte del Estado, el derecho exclusivo de uso y explotación de un invento por un tiempo determinado (20 años) al inventor de la misma. Éste podía explotar su idea, otorgando licencias de uso a cambio de un beneficio económico. Con esto se aseguraba que el invento se daría a conocer y no se guardaría como secreto industrial, pudiéndose hacer uso libre de él al término de la patente.

Los sistemas de patentes se instauraron para motivar a los inventores a compartir sus descubrimientos, además de para protegerlos de las grandes empresas. Se buscaba, al igual que con los derechos de propiedad intelectual, que artistas y científicos aportaran los productos de su creatividad, y con esto se beneficiaran ellos mismos y a la sociedad en su conjunto.

Todos los derechos de propiedad intelectual son únicamente licencias que otorga la sociedad porque se pensaba, correcta o equivocadamente, que la sociedad en conjunto se beneficiaría al ser otorgados. Pero en cualquier situación particular, necesitamos preguntarnos: ¿nos beneficia haber otorgado tal licencia? ¿Qué tipo de acto estamos permitiendo que haga una persona? (Stallman, 2004: 55)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En el caso del *software*, las patentes se convierten en una pesadilla pues, como se ha dicho, hablar de *software* es hablar de algoritmos, de una secuencia de órdenes, de una serie de ideas organizadas para que la máquina haga un proceso determinado. Cada programa se compone de miles de líneas de código; es decir, miles de algoritmos, de ideas organizadas que pueden ya estar patentadas. Como dice Stallman:

[...] cuando escribes un programa, estás usando una enorme cantidad de ideas diferentes, cada una de las cuales puede estar patentada por alguien. Un par de ellas podría estar patentada por alguien en una combinación. Podría haber varias maneras distintas de describir una idea, que podrían estar patentadas por gente distinta. Así que posiblemente hay miles de cosas, miles de puntos vulnerables en tu programa, que podrían estar ya patentadas por cualquier otro. (Stallman, 2004: 150)

Varios estudios confirman que la innovación tecnológica y el crecimiento económico no necesariamente se ven beneficiados con el uso de patentes y derechos de propiedad (Ver Chang, 2010; Von Hippel, 1988). Es posible que el uso de este tipo de legislaciones motiven la inversión, pero su uso extensivo puede no sólo dificultar, sino reducir, la adopción y el crecimiento.

El debate [sobre derechos de propiedad intelectual] ha revelado que, mientras algo de protección puede ser necesario para motivar a las firmas a invertir en la generación de conocimiento, al menos en ciertas

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

industrias (p.e. Química, farmacéutica, software), demasiada protección intelectual puede ser mala para la sociedad (Chang, 2001; Stiglitz, 2007: ch.4, en (Chang, 2010).

Lo que explica Chang en su artículo es que el exceso de protección incrementa los costos, lo que puede opacar los beneficios de la innovación misma. Por un lado porque la innovación es un proceso incierto, pero, además, los altos costos pueden hacer que la difusión tecnológica se dificulte, al no permitir el intercambio y revisión entre pares. Al no existir innovación, ni difusión, los beneficios para la sociedad de la creatividad protegida, entonces, quedan en entredicho. Cuando el mercado impone su interés sobre el interés de la sociedad, los resultados se deshumanizan y pierden el sentido. Como dice Stallman:

El paradigma de la competencia es una carrera: al premiar al ganador, estamos alentando a todos a correr más rápido. Cuando el capitalismo realmente trabaja de esta manera, hace un buen trabajo; pero sus partidarios están equivocados al asumir que siempre funciona así. Si los corredores olvidan por qué se otorga el premio y se centran en ganar sin importar cómo, pueden encontrar otras estrategias —como atacar a los otros corredores. Si los corredores se enredan en una pelea a puñetazos, todos llegarán tarde a la meta. (Stallman, 2004: 55)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Los nuevos bienes comunes del conocimiento y la información necesitan de protección, pero no de una protección que enajene este tipo de recursos, sino que, al contrario, logre su difusión y utilización social a gran escala.

Cuando hablamos del entorno de la información, del espacio cultural y simbólico que ocupamos como individuos y ciudadanos, la diversificación de las restricciones bajo las que operamos, incluyendo la creación de espacios relativamente libres de las leyes que estructuran el mercado, va al mismísimo corazón de la libertad y la democracia. (Benkler en Helfrich, 2008: 132)

Las comunidades y redes productoras de conocimientos que creen en que éstos son bienes comunes que deben ser compartidos, cada vez más optan por el tipo de protección legal que otorgan las licencias semi-libres y libres, como *Creative Commons*.

Desde su aparición a finales de 2002, más de un millón de obras se han hecho públicas en Internet a través de alguna licencia Creative Commons (CC), convirtiendo este proyecto en un fenómeno social en Internet. Su objetivo es poner al alcance de los autores un modelo de licencia estandarizada que, en lugar de prohibir el uso (la idea del «todos los derechos reservados»), lo autorice bajo algunas condiciones (es decir, «algunos derechos reservados»). (Xalabander Plantada, 2006)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En sí, las licencias *Creative Commons* (CC) motivan y protegen la difusión de las obras en internet, facilitando a autores y usuarios la selección e identificación de los derechos y usos posibles de la obra así protegida. Todas estas licencias se basan en el derecho de autor, conocido como copyright, para así tener un reconocimiento legal; sin embargo, le dan la vuelta a esta legislación, indicando, a través de cláusulas explícitas en el texto de la licencia, las libertades de uso y modificación en cada caso.

Creative Commons, así como las licencias semi-libres y libres utilizadas actualmente tienen como base la idea de *Copyleft*, desarrollada por Richard Stallman. El término copyleft denota el movimiento de protección legal, utilizando la legislación de derechos de autor, sobre la libertad de uso, modificación y distribución de obras y productos.

Por defecto, *Creative Commons*, permite reproducir, distribuir, transformar y difundir las obras con cualquier fin y de manera gratuita. Los autores que eligen utilizar *Creative Commons* pueden excluir o limitar estas libertades, resultando en diferentes tipos posibles de protección. Partiendo de que, sin excepción, una constante es el reconocimiento del autor de la obra licenciada, se puede excluir la posibilidad de modificación para crear una obra derivada, el uso comercial, y/o imponer el copyleft; es decir, que toda obra derivada lleve la misma licencia que el original. Los autores o investigadores pueden utilizar CC libre, con alguna de estas restricciones o combinaciones de las mismas, estipulando esto en el texto que se adjunta a la obra publicada en internet. Existen otras cláusulas posibles, según las legislaciones nacionales.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Además de *Creative Commons*, existen licencias semi-libres y libres, entre ellas, y como base de las mismas, la Licencia Pública General GNU (*GNU-GPL*, por sus siglas en inglés), creada por Richard Stallman para proteger el *software* libre, bajo la idea de *Copyleft*.

La GNU GPL original fue toda una revolución, ya que por primera vez se había diseñado una licencia de derechos de autor con el fin de defender la libertad de cooperación de todos. (Richard Stallman en Helfrich, 2008: 289)

Ésta licencia, en principio, permite el uso, modificación, acceso y distribución en sentido amplio, y de ella se derivan o son compatibles, entre otras: *LGPL* (licencia *GNU-GPL* con restricciones), *AGPL* (Afferro; licencia para programas que se corren en una red), Apache Licence (con algunos requerimientos), Artistic Licence, Berkeley Database Licence, Boost *software* Licence (compatible, pero no es *copyleft*), *BSD*-modificada (compatible y con *copy-left*), CC0 (Creative Commons de dominio público), etcétera.³⁵

2.4.2.3 Apoyo y financiamiento

Los trabajos mencionados de Elinor Ostrom a lo largo del presente capítulo muestran cómo las comunidades de bienes comunes administran los recursos de manera organizada, siguiendo reglas formales o informales en mayor o menor grado;

³⁵ Para más información sobre licencias libres: www.fsf.org

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

es decir, estas comunidades se autogobiernan y han demostrado su sustentabilidad a largo plazo (Ostrom, 1990). La preservación de los bienes comunes, en general, se apoya en el trabajo y esfuerzo de las comunidades protectoras, pero también en financiamiento directo de actores dentro o fuera de la comunidad, a través de donativos de individuos o empresas, así como inversión estatal.

[...] los recursos comunes y la propiedad sobre dichos recursos existen en un espectro de ordenaciones institucionales. Que en este espectro un sistema de gestión de recursos sea más o menos sostenible y eficiente depende de las características tecnológicas del recurso y de los patrones de su uso en cualquier momento histórico dado (Benkler en Helfrich, 2008)

Los bienes comunes presentan una nueva manera de entender el significado y origen de la riqueza. En esta perspectiva, la preservación y la amplia distribución es más importante que la ganancia monetaria y la acumulación. Los recursos pueden venderse, pero a bajo costo y sólo si las ventas no provocan daños a las comunidades y a los mismos bienes. Más allá de valores de cambio, los bienes comunes ponen el énfasis en los valores de uso y en las formas en las que estos valores se reproducen y preservan.

Para mucha gente, es usual ver estas dos esferas de poder [gobiernos y empresas] como los únicos regímenes efectivos para la administración de los

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

recursos. Sin embargo, ha quedado claro (en años recientes) que hay un tercer ámbito de soluciones en gran medida soslayado: los bienes comunes. (Bollier en Helfrich, 2008: 30)

El Estado puede apoyar, a través de programas, la gestión de los bienes comunes, pero no necesariamente es necesario para el sostenimiento de estos recursos, como bien lo explica Elinor Ostrom (2005) y David Bollier en la siguiente cita:

[...] es importante no mezclar programas gubernamentales con la gestión de los bienes comunes. Pueden estar imbricados, pero no son lo mismo.... puede ofrecer un apoyo para los bienes comunes facilitando el establecimiento de nuevas instituciones que puedan ser manejadas por sus propios comuneros [...] Como ejemplos tenemos cooperativas, organizaciones locales en favor de la conservación de la tierra (land trusts), radio y televisión comunitarias, y mercados comunitarios. (Bollier en Helfrich, 2008: 38)

En el caso de los nuevos bienes comunes de la información y el conocimiento, p.e., el conocimiento científico, la mayoría de los países destina parte de su producto interno bruto a investigación y desarrollo. Muchos científicos reciben financiamiento del Estado para el desarrollo de sus investigaciones. El producto de éstas, plasmado en artículos científicos, no tiene como finalidad la ganancia económica, sino

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

el desarrollo de la sociedad que ha generado estos nuevos conocimientos.

Otra forma de financiamiento es aquella que se obtiene a partir de donativos de miembros y simpatizantes de las comunidades de bienes comunes, como sucede en la producción de *software* libre. Los donativos para las comunidades de bienes comunes son parte de arreglos institucionales para la sustentabilidad de los mismos. En general, son los miembros, más o menos cercanos, de las propias comunidades los que aportan voluntariamente para este fin. Los donadores aportan cuotas para el mantenimiento de un programa con cuyas ideas están de acuerdo, no sólo como un acto de beneficencia o filantropía, sino al contrario, pues su interés está basado en ideales éticos y convicciones ideológicas, que los identifican con el grupo apoyado.

Por un largo período, he sido partidario de las ideas de la FSF y de todo el movimiento de software libre. Hoy, he querido hacer una contribución tangible a la FSF, así como abiertamente declarar mi apoyo para la defensa clave de las libertades del software. (Cathal McGinley, miembro de FSF #5886)

El compromiso de miembros y simpatizantes, tanto corporativos como individuales, de este tipo de comunidades hace posible que éstas sigan funcionando. Este compromiso se ve respaldado por el trabajo de mantenimiento y difusión de estos nuevos bienes comunes del conocimiento y la información. Las comunidades en este tipo de bienes desarrollan sus recursos

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

y los organizan hacia adentro, pero parte importante de su subsistencia tiene que ver con la difusión de su trabajo hacia poblaciones más amplias, lo que consiguen a través de redes de intercambio.

2.5 Software libre como bien común

Varios autores, como se mencionó en el capítulo primero, han abordado el tema del *software* libre; sin embargo, su estudio como bien común lo ha hecho principalmente Charles M. Schweik, desde la perspectiva teórica del neo-institucionalismo propuesta por Elinor Ostrom (Ostrom et al., 2007; C. Schweik, 2007; C. M. Schweik et al., 2008; C.M. Schweik et al., 2010; Charles M. Schweik & Kitsing, 2010).

Las instituciones son las reglas y normas sociales introyectadas o incorporadas en los individuos. Siguiendo a Norbert Elías, el individuo se entiende como la parte de una estructura de individuos interdependientes en una sociedad. El individuo tiene un esquema fijo de comportamiento, devenido de la sociedad, en tanto estructura de reglas y normas que se incorpora dentro del individuo a través de la red de interdependencias.

El orden imperceptible directamente a los sentidos, el orden invisible de esta convivencia, ofrece a la persona individual únicamente un abanico más o menos limitado de posibles modos de comportamiento y funciones. (Elías, 1990: 29)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

El neoinstitucionalismo es la teoría que explica cómo emerge y cambia la red de interdependencias, a través del cambio en las instituciones formales e informales de la sociedad. *Son las limitaciones ideadas por el hombre que dan forma a la interacción humana, estructuran incentivos en el intercambio humano.* (North, 1993: 13). Las instituciones se presentan como marcos que organizan las acciones de intercambio para impedir o limitar los posibles conflictos, pero al mismo tiempo que limitan, pueden abrir posibilidades y oportunidades de acción; además, reducen la incertidumbre.

Definimos instituciones como las reglas formales e informales, que son entendidas y usadas por una comunidad [...] Son las reglas que establecen lo que se permite y lo que no para los individuos en una situación que se puede analizar y explicar. (Ostrom et al., 2007: 42)

Desde esta perspectiva teórica, Elinor Ostrom (1990; 2007) desarrolló el marco de desarrollo institucional (IAD, por sus siglas en inglés), que sirve para describir y entender la organización y evolución de comunidades de producción de bienes comunes. Con él facilita el análisis institucional comparativo para responder *una de las preguntas sociales y políticas fundamentales: ¿Cómo humanos falibles se reúnen, crean comunidades y organizaciones, toman decisiones y crean reglas para sostener un recurso o lograr un resultado deseado?* (Ostrom et al., 2007: 42)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Charles M. Schweik, aplicando el marco de Ostrom, ha estudiado FLOSS como bien común, presentando sus resultados en tres artículos fundamentalmente³⁶. El primero de ellos (C. Schweik, 2007) describe el diseño institucional de varios proyectos FLOSS para determinar su capacidad para evitar el abandono. Sus conclusiones indican que la participación en proyectos no es sólo voluntaria, sino que también existen empresas, organizaciones y agencias del gobierno que financian a algunos desarrolladores. En sus observaciones indica que la colaboración en proyectos es de grupos, en general, pequeños.

Cabe mencionarse, aunque esto se desarrollará en el capítulo 4 de esta tesis, que se observó que la categorización hecha en los repositorios (*FLOSSmole*) hace una división muy desagregada de la participación en el desarrollo de un programa. De hecho, clasifica como 'proyecto' a componentes de una aplicación (p.e. un 'proyecto' puede ser el desarrollo de la interfaz gráfica, que es independiente en un momento del proceso de construcción de un programa, del desarrollo del código de sonido, o de la traducción, etcétera). Esto implica que un 'proyecto', así definido, puede tener en un momento del tiempo 1 ó 2 desarrolladores, y otro componente sólo 1 ó 10 programadores trabajando juntos; en otro momento del tiempo, esto puede cambiar por las revisiones entre pares y la integración de todos

³⁶ Existe un libro por publicarse con los resultados de un largo estudio de 5 años, financiado por la Fundación Nacional Estadounidense para la Ciencia, que responde a los factores que llevan al éxito o abandono de proyectos libres y de fuente abierta. En este análisis, Schweik estudio los datos obtenidos de Agosto a Octubre de 2006 sobre 107,747 proyectos FLOSS hospedados en el repositorio de internet Sourceforge.net

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

los componentes para obtener la aplicación final completa. Para poder realmente ver la colaboración de manera integrada se tiene que hacer un corte más amplio en el tiempo y definir cuáles proyectos corresponden a cuáles aplicaciones, sólo entonces puede verse qué proyectos se interrelacionan y, por lo tanto, qué desarrolladores colaboran juntos en el desarrollo de un sistema, programa o aplicación.

Retomando el trabajo de Schweik en cuanto al diseño institucional, encontró que los proyectos *FLOSS* son bienes comunes basados en internet, lo que es particularmente importante porque es el espacio donde se hace posible la colaboración entre desarrolladores desde distintos puntos espacio-temporales, y porque esto motiva a que se siga un régimen de propiedad común para desarrollar un bien que sirva a todos los participantes. Sin embargo, cada programa tiene uno o varios propietarios, definidos en las licencias seleccionadas para protegerlo, otorgándoles a éstos una especie de control sobre los participantes en, y beneficiarios de, el proceso de construcción del programa. En este artículo, Schweik encontró en los nueve proyectos estudiados, que existían pocas reglas formales dentro de los equipos de trabajo, y que la forma de organización era más bien horizontal.

Al aplicar el marco de desarrollo institucional de Ostrom (Charles M. Schweik & Kitsing, 2010) en 7 programas geográficos *FLOSS* encontró que en la mayoría de los proyectos existían líderes de los mismos, y en todos un Comité de Vigilancia; que tenían reglas formales sobre la participación y las tareas designadas, así como arreglos de acción colectiva en los que se definía quién participaba y cómo en los distintos

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

momentos y procesos de decisión, aunque las decisiones clave, en general, se manejan bajo normas sociales, más que formales.

De acuerdo a un líder de proyecto, un gran número de reglas formales puede desmotivar a los programadores para tomar parte en un proyecto. (p. 22)

La información es manejada por correo electrónico y por canales de IRC (internet relay chat). No existen reglas definidas sobre los pagos a programadores, ni sobre los incentivos y/o castigos por incumplimiento; estos siguen reglas informales, entendidas en la comunidad de programadores.

Se indicó muchas veces en nuestras entrevistas que las recompensas monetarias no son el único motivador y que la compensación no está directamente vinculada con el desempeño [...] El papel de la compensación monetaria se entiende más por el factor de que muchos de estos casos empezaron como proyectos voluntarios y que el éxito llevó a la oportunidad de obtener un pago por el trabajo. (p. 23)

Finalmente, en el artículo que entrega avances sobre el libro sobre el éxito o abandono de proyectos FLOSS (C.M. Schweik et al., 2010), Schweik destaca que el éxito de los proyectos está asociado con comunidades mayores de usuarios del programa, así como por equipos de trabajo más grandes; asimismo, que los factores de abandono o éxito difieren según las etapas del

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

proceso de desarrollo del programa. Un resultado importante indica que hay evidencia de que la idea de fuente abierta está creciendo en escala y alcance; más allá de motivaciones ideológicas, la colaboración se está fomentando no sólo desde los voluntarios, sino desde empresas, agencias del gobierno y organizaciones no lucrativas.

[...] mientras que el software libre es ya altamente visible, es, de hecho, sólo un ejemplo de un fenómeno social-económico mucho más amplio. Sugiero que lo que estamos viendo es la emergencia amplia y profunda de un nuevo modelo de producción en el ambiente de la red digital. Le llamo modo basado en la producción entre pares para distinguirlo de los modelos basados en la propiedad y el contrato de las firmas y mercados. Su característica central es que grupos de individuos colaboran exitosamente en proyectos de larga escala, siguiendo un grupo diverso de motivaciones y señales sociales, más que las órdenes gerenciales y los precios del mercado. (Benkler 2002 en Hess, 2008: 26)

Como puede verse, el estudio del *software* libre y abierto como bien común desde el neoinstitucionalismo está arrojando resultados.

Sin embargo, aunque éstos son importantes para entender las dinámicas de las comunidades de desarrollo de *software* libre y abierto, estos resultados se presentan a un nivel micro; es decir, poniendo énfasis en los procesos internos al desarrollo de los

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

programas y en las acciones y motivaciones individuales para colaborar o dejar de hacerlo. En este trabajo, se busca ampliar la perspectiva hasta el nivel que permita observar la interrelación de los programas y los programadores, si no en toda la red, sí en el intercambio general de varios repositorios. Este abordaje se realizará a través del análisis de redes sociales (ARS), a la luz del interaccionismo estructural, que, como parte del marco teórico se explicará en el siguiente capítulo.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Capítulo 3 Dinámicas de interacción en redes y comunidades de producción de FLOSS

El *software* libre y abierto es producto de un proceso de producción colectiva que sigue reglas y se organiza en comunidades y redes. Su organización en comunidades emerge, a pesar de que los participantes no se conozcan físicamente, pues su interacción virtual se basa en intereses y objetivos comunes que logran una identidad de grupo. Citando a Rheingold (2002):

Las comunidades virtuales son agregados que emergen de la Red cuando suficiente gente se involucra en discusiones públicas el tiempo suficiente para generar sentimientos humanos suficientes que formen redes de relaciones personales en el ciberespacio (Introducción, p. 6)

La apertura del código de programación permite el aprendizaje activo de los usuarios, a través de la exploración y comprensión de las acciones que ejecutan las aplicaciones dentro del ordenador. Como se explicó en el capítulo precedente, se considera un bien común, ya que 1) es producido por grupos o comunidades definidos; 2) sus formas de producción y distribución tienen reglas formales e informales; 3)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

se sustenta sobre una cultura definida –la cultura *hacker--*, que preexiste a, y sobrevive en el contexto virtual; además, es la propia comunidad que lo genera la que así lo define en la regulación o tipo de licencia que escoge para protegerlo como bien común.

[Sus] dos focos primarios son la producción entre pares y las iniciativas de acción colectivas [...] Este tipo de acción colectiva en masa ilustra cooperación y escala sin precedentes en un bien común. En muchos bienes comunes de acción colectiva podemos detectar un tipo común de pensamiento afín a estos recursos: la creencia en el bien común y el trabajo para crear productos compartidos basados en participación voluntaria y reciprocidad. (Hess, 2008: 8)

La participación en las comunidades de desarrollo de *software* libre y abierto implica que se cuenta con conocimientos de programación y habilidades técnicas, además del propio conocimiento y seguimiento de la cultura. Sin embargo, existe la posibilidad de aprender las bases técnicas y adoptar la cultura, a partir de las reglas tácitas y explícitas de los miembros. Una vez dentro de una comunidad, la inserción en las dinámicas colectivas (p.e. soporte a través de chat o correo electrónico, participación en wikis y foros, retroalimentación más o menos especializada, etcétera) se da de manera casi espontánea y generalizada. Esto lo pude corroborar en la observación participativa, tanto en los salones de chat de resolución de problemas, como en la suscripción a la lista de correo

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

electrónico de la aplicación *Libre-Office* para el mismo fin. En este sentido, la acción colectiva de los miembros de una comunidad determinada de *software* libre y/o abierto no resulta extraña y, según lo observado, no necesita de incentivos económicos, a veces ni siquiera de reconocimiento de pares, para desarrollarse.

La Teoría de la Acción Racional³⁷ ha planteado que la acción colectiva en la sociedad es problemática, puesto que no puede explicarla. Esto es así porque el planteamiento de sus supuestos (actor maximizador, óptimo social, equilibrio del sistema) no permite ver dinámicas relacionales que no respondan a intereses individuales egoístas. El problema de la acción colectiva³⁸, desde la Teoría de la Acción Racional, se plantea que, siendo que los

³⁷ Esta teoría deriva de la teoría de la elección racional de la Economía neoclásica. El concepto de acción racional hace referencia a una acción teleológica (orientada por los resultados), producto de una evaluación 'racional' sobre la utilidad (beneficios y costos) que se puede obtener según el curso de acción elegido para llegar a un objetivo determinado. Esta acción es diferente según los objetivos que se quieren alcanzar y los valores que tienen los actores en cada contexto (los valores están determinados socialmente). Cabe mencionar que esta teoría en Sociología ha adaptado y replanteado los conceptos básicos, tomando en cuenta a las instituciones, que no estaban presentes en la perspectiva de la Economía neoclásica. Entre los principales autores que desarrollaron esta teoría están James Coleman, John Goldthorpe y Mancur Olson.

³⁸ Este problema fue planteado por Mancur Olson (1992) en "La lógica de la acción colectiva". Según él, la acción colectiva planteaba tres conflictos esenciales: el free-rider o gorrón, el beneficio de los bienes públicos sin tener que correr con los costos de participación, así como los altos costos de participación para aquellos que contribuyen, contra los que no lo hacen, hacen que los primeros dejen de contribuir.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

individuos son egoístas y buscan su máximo beneficio, no tendrían razones para cooperar en la sociedad, pues los intereses de cada actor son divergentes y no siempre tienen intereses comunes. Mancur Olson (1992), autor de este planteamiento, argumenta que la colaboración sólo puede existir en grupos pequeños donde existan mecanismos de coacción e incentivos especiales para los cooperadores.

[...] mientras más grande sea el grupo, más lejos estará de proporcionar una cantidad óptima de un bien colectivo” [...] la norma para determinar si un grupo tendrá capacidad para actuar, sin coacción ni estímulos externos, en favor del interés del grupo es (como debe serlo) la misma para los grupos del mercado y para los ajenos a él: depende de si las acciones individuales de uno o más miembros cualesquiera de un grupo son perceptibles para cualesquiera otros. Esto está, muy obvia, pero no exclusivamente, en funcion del número de miembros del grupo.” (pp. 45 y 55).

Sin embargo, siguiendo el debate de la Sociología Económica, los actores sí actúan racionalmente, pero esta racionalidad tiene un contexto, en el que a veces se toman decisiones para maximizar beneficios, pero otras veces, no. Esta perspectiva sociológica reconoce y analiza las distintas lógicas a las que pueden responder los actores en un momento dado.

[...] en muchas situaciones, los agentes optan por opciones sistemáticamente diferentes de las que cabe

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

prever a partir del modelo económico [...] el cálculo estrictamente utilitario no puede dar cuenta por completo de las prácticas que permanecen inmersas en lo no económico. (Bourdieu, 2000: 21 y 23)

Si bien es cierto que los planteamientos de Olson sobre la necesidad de incentivos para motivar la colaboración aparecen en la realidad de estas comunidades de producción, la cooperación en las redes de FLOSS excede la cantidad de incentivos, y no se limita a su existencia. Los procesos de desarrollo de FLOSS demuestran, no sólo que la acción colectiva es posible y puede darse de manera aparentemente espontánea, sino además, que es posible en grandes grupos de gente que sólo se conoce virtualmente. Este tipo de cooperación hace que, como se explicó en el capítulo precedente, el *software* libre y abierto sea un bien común. Siguiendo los argumentos de Rheingold (2002), se puede decir que la acción colectiva en este tipo de grupos no sólo es posible, sino necesaria para sus miembros:

Todo grupo cooperativo de gente existe encarando un mundo competitivo, pues ese grupo de gente reconoce que hay algo valioso que gana sólomente estando unidos. Buscar los bienes colectivos de un grupo es un modo de buscar los elementos que mantienen unidos en comunidad a individuos aislados. (p.14)

En este capítulo, describiré las formas en que se organiza la retroalimentación para la producción de *software* libre dentro

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

de Internet, así como los fundamentos que motivan la colaboración y solidaridad entre desarrolladores y usuarios de este tipo de sistemas, aplicaciones y programas. El análisis de redes sociales (ARS), que se ocupa en este trabajo y se explicará a continuación, permite la visualización de este tipo de acción colectiva en línea a través de repositorios.

3.1 Teoría del Esquema y el Conexionismo: motivaciones

La literatura que busca explicar la colaboración y la acción colectiva es vasta. Los estudios, en general, abordan los análisis intentando encontrar la racionalidad de actores que cooperan aunque otros no lo hagan (Boyd & Richerson, 2009; Brown & Vincent, 2008; Buchan et al., 2009; Dasgupta, 2009; Gilbert, 2006; Goldthorpe, 1998; Mathew & Boyd, 2009; Postmes & Brunsting, 2002; Puurtinen & Mappes, 2009; Tropp, 2004). En el caso del desarrollo de *software* libre, la literatura sobre las motivaciones de los actores desde la perspectiva de la acción racional (Bonaccorsi & Rossi, 2002; a. Hemetsberger & Reinhardt, 2009; K. R. Lakhani & Wolf, 2005; Stewart, 2005), se une a la de los estudios sobre innovación (E. Von Hippel, 2007) y los neo-institucionales (Charles M. Schweik & Kitsing, 2010).

En esta tesis, se abordará el estudio de las motivaciones de los actores para la colaboración en el desarrollo de *software* libre y abierto desde la perspectiva de la Teoría del Esquema y el Conexionismo (Strauss & Quinn, 1997). En esta perspectiva teórica de la Antropología Cognitiva se establece que los individuos procesamos la información a través de estructuras

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

mentales que nos ayudan a organizar y vincular elementos cognitivos. Los esquemas son pautas de comportamiento que, aprendidos en un momento, permiten dar sentido a cualquier situación.

Según la teoría del conexionismo, los esquemas son redes de elementos cognitivos interconectados, como las neuronas en nuestros cerebros, pero éstos representan conceptos genéricos aprendidos y almacenados en la memoria. La conexión de los elementos cognitivos, dentro de las estructuras individuales, genera un significado que se refuerza o desdibuja, a partir del ensayo y el error en la experiencia objetiva de los individuos. Así, los elementos y el modo en que se conectan entre sí en cada situación, inclinan a los individuos a actuar y pensar de determinada manera.

Un significado es resultado de la interpretación que hacen los individuos de los estímulos del exterior. Son necesarios para dar sentido a las situaciones, por lo que implican que los actores puedan identificar el estímulo (objetos, eventos o comportamientos) y tengan expectativas sobre el mismo. Estos significados se basan en la relación de estructuras mentales intrapersonales y extrapersonales para funcionar. Cuando refieren a interpretaciones recurrentes y compartidas por un amplio número de personas, se conocen como significados culturales.

Los significados culturales se producen, siguiendo la teoría del esquema y el conexionismo, a partir de una doble estructura que consta de elementos interiorizados en forma de esquemas,

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

“cultura en las personas”, y elementos objetivados, “cultura pública”.

Los esquemas, que se aprenden a través de la cultura, son estructuras de largo plazo introyectadas en los individuos y que interactúan dialécticamente con las estructuras externas. Estas estructuras mentales organizan los conocimientos y los relacionan, lo que permite a los individuos dar sentido a sus experiencias y actuar en consecuencia de los estímulos recibidos del exterior. Lo intrapersonal y lo extrapersonal no pueden disociarse, pues están en constante intercambio.

La forma en la que emergen los esquemas es a través del discurso de los individuos. Pueden identificarse a partir de las metáforas utilizadas por los actores para ejemplificar situaciones, mismas que son utilizadas como ilustración y no como analogías. Éstas, a su vez, son alimentadas por la experiencia cotidiana y los objetos y eventos del mundo.

En el caso del desarrollo de *software*, el esquema de la cooperación contiene varios elementos interconectados que tienen que ver con la creatividad, la libertad, etcétera. Los conceptos utilizados por muchos desarrolladores tienen que ver con la organización horizontal o vertical del trabajo, utilizando metáforas como “la forma catedral” o “la forma bazaar”, devenidas de un famoso libro de Eric Raymond. Otro ejemplo, aún más generalizado es la utilización del término “libertad” en términos de acción creativa, es decir, posibilidad de modificar, usar, mejorar y distribuir.

3.2 Interaccionismo estructural y Análisis de Redes Sociales (ARS)

La Sociología en general, y particularmente la corriente estructuralista desde Marx, reconoce que hay estructuras que constriñen y/o posibilitan la acción de los individuos, al mismo tiempo que observa cómo los individuos y los contextos interactúan con estas estructuras y mutuamente se transforman. Además, observa la importancia de diferentes ámbitos que influyen en las motivaciones y toma de decisiones de los actores. Sin embargo, los individuos no están aislados, por lo que las razones y motivos de su actuar se analizan en términos relacionales.

Al observar las relaciones entre grupos, emergen las diferencias individuales y grupales según los contextos espacio-temporales. Esto sucede porque la relación que se establece entre cada parte con el todo es específica en tiempo y lugar, lo que genera formas particulares de sociedades e individuos, estructuras específicas en los grupos humanos.

Cada uno de los seres humanos que caminan por las calles aparentemente ajenos e independientes de los demás está ligado a otras personas por un cúmulo de cadenas invisibles [... vive] dentro de una red de interdependencias que él no puede modificar ni romper a voluntad sino en tanto lo permite la propia estructura de esa red ... Este contexto funcional posee una estructura muy específica en cada grupo humano. (Elías, 1990: 29)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Siguiendo los planteamientos de Norbert Elías, la sociedad es el orden ofrecido al individuo, en tanto las posibilidades limitadas en que éste puede comportarse y funcionar de acuerdo a la posición que ocupa en el todo social. El orden social es así el tejido de una red de relaciones entre individuos, un sistema en tensión flexible, donde la relación entre los individuos y la sociedad tiene una estructura y una regularidad propia. La integración social debe buscarse, entonces, en esa red de interdependencias. Es de la estructura del todo social que debe partirse para comprender la forma de las partes, pensándolas dentro de un contexto relacional y funcional con respecto al todo.

[...] en el momento en que tuvimos la idea de que las personas están conectadas por inmensas redes sociales, nos dimos cuenta de que nuestra influencia no termina en las personas que conocemos. Si es cierto que tenemos alguna influencia en nuestros amigos y si es también cierto que éstos tienen influencia en sus amigos, entonces en teoría nuestras acciones pueden alcanzar a personas a las que ni siquiera conocemos. (Christakis & Fowler, 2010: 12)

El análisis de redes sociales (ARS), como tal, aparece desde 1920 en Harvard, pero toma fuerza hasta finales de los 40 con los trabajos de Freeman, Flament, Merton, Moreno, etcétera. Derivado del estructuralismo, destaca y profundiza la observación de las relaciones o interacción entre las personas.

La teoría de redes se basa en la suposición de que las

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

relaciones sociales son la clave para explicar, tanto la acción individual, como los resultados colectivos [...] se refiere al estudio de las formas estructurales –o patrones—de los lazos que vinculan a los nodos (Schmidt, 2006).

Este tipo de análisis cruza disciplinas y perspectivas teóricas, usándose en Sociología, Antropología, Psicología, Matemáticas, Economía, Ciencia Política, Comunicación, Estadística, Medicina, Epidemiología, Física, etcétera.

[...] creemos que nuestras conexiones con otras personas son lo que más importa y que, al vincular el estudio de los individuos con el estudio de los grupos, la ciencia de las redes sociales tiene mucho que decir sobre la experiencia humana. (Christakis & Fowler, 2010: 15)

Aunque la observación de estructuras o patrones existe desde tiempo atrás, el ARS representa un nuevo paradigma por la forma en que aborda y sistematiza la realidad, convirtiéndose en un campo de estudio definido. Las redes permiten ver la interacción social y el producto de la misma a través de organizaciones formales e informales. Por ello, Schmidt (2006) indica que:

Las características de las redes se usan para describir de modo más preciso grupos o configuraciones existentes de unidades específicas. (Schmidt, 2006)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Una red, desde el análisis de redes sociales (ARS), se visualiza como un grafo³⁹, es decir, un conjunto de nodos y líneas uniéndolos. Para el análisis social, la red se entiende como actores y vínculos entre los mismos. Así, entre los conceptos básicos que sirven para observar y describir redes en este tipo de análisis se encuentran:

- *Orden*, que representa el número de elementos; es decir, el número de nodos,
- *Grado*, que representa el número de líneas; es decir, vínculos entre actores o nodos,
- *Distancia*, que es la longitud de la trayectoria más corta entre dos nodos, siendo una trayectoria un camino secuencial entre nodos y vínculos uniéndolos donde no se repiten nodos. Cabe mencionarse que un *paseo*, es aquel camino que no repite vínculos,
- *Geodésico* es el camino más corto entre dos nodos o actores,
- *Diámetro*, representa en una red conectada, la longitud del geodésico más grande,
- *Conexidad* es el concepto que se utiliza cuando cada par de nodos de una red tiene al menos un vínculo,

³⁹ Los grafos son una forma de representar redes, por lo que mucha de la terminología usada para referirse a las propiedades estructurales de una red proviene de la terminología que se usa en la teoría de grafos.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En general, el análisis de redes sociales responde a cinco fundamentos analíticos, a saber: 1) el poder explicativo de las estructuras relacionales es mayor al de la descripción de los atributos de los miembros de un sistema, 2) la emergencia de las normas responde de la propia estructura relacional en el momento de la interacción, 3) el funcionamiento de las relaciones entre dos actores o nodos está determinado por la propia estructura social, 4) son las redes, y no los grupos, los que conforman el mundo, y 5) los métodos del estructuralismo explican de manera más amplia los fenómenos, que los métodos individualistas. (Ver De la Rúa, 2009: 262-263)

En el Análisis de redes sociales, como dice De la Rúa (idem: 268):

Se presta una especial atención a la posición de los actores y la posición de las relaciones. Los actores interactúan, negocian, se ajustan, se ponen de acuerdo, se hacen presión... eso tiene un efecto en las normas emergentes, sobre las acciones de los miembros del sistema y, a término, afecta la transformación de la estructura.

El Análisis de Redes Sociales: a) es motivado por una intuición estructural, b) se basa en la sistematización de datos empíricos, c) utiliza imágenes gráficas, y d) se apoya en modelos matemáticos y/o computacionales.

Los métodos de análisis tratan directamente la naturaleza relacional de la estructura social, lo que

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

exige la creación de modelos de análisis específicos a partir de la teoría de grafos y del cálculo matricial (De la Rúa, 2009: 267)

La utilización de imágenes gráficas, generadas por computadora, es una estrategia de análisis y presentación de los datos innovadora y funcional, ya que permite la visualización de las relaciones entre los actores, y con ello, la identificación de las formas estructurales emergentes.

La presentación visual de las dinámicas de las redes permite incorporar una extensión amplia de datos con lo que se facilita el planteamiento de hipótesis que hacen avanzar el conocimiento científico, así como perspectivas distintas sobre problemas ya planteados. (Molina, Ruiz, & Teves, 2005: 3)

La rapidez y efectividad que proporcionan las computadoras permiten detectar en las imágenes si existen inconsistencias en los datos, o vínculos que llevan a nuevas observaciones. Así, se da un proceso de ida y vuelta entre la realidad y la abstracción, permitiendo profundizar en los análisis sin perder de vista la realidad estudiada.

[...] entendemos que incorporar de forma intensiva la visualización tanto en la obtención de datos reticulares como en su análisis representa una oportunidad para desarrollar todo el potencial de la aproximación. (Molina, Ruiz, & Teves, 2005: 3)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

El desarrollo de ARS ha dependido, y motivado, el desarrollo de distintos programas computacionales (Freeman, 2004), como *Socpac*, *Blocker*, *Sock*, *Complt*, *Concor*, *Structure*, *Negopy*, *Sonet*, *Center*, *Cobloc*, *Sonis*, *Ucinet*⁴⁰, etcétera. Para este trabajo, por su carácter abierto, se utiliza *Pajek*, desarrollado por Vladimir Batagelj and Andrej Mrvar⁴¹.

3.3 Organización en la producción de *software* libre y abierto

Como se comentó en el primer capítulo de esta tesis, la construcción y apertura de Internet fue fundamental para permitir la colaboración masiva entre desarrolladores en la producción de aplicaciones libres y abiertas. La práctica colaborativa existía desde antes de la aparición de la red de redes, pues, como menciona el Dr. Trejo es uno de los rasgos de la Sociedad de la Información:

[La] colaboración es atributo que se traduce lo mismo en acciones solidarias que en proyectos intelectuales que no serían posibles sin el soporte que proporciona la Red. (Trejo-Delarbre, 2006: 4)

Como se describió, la forma de construcción y operación de Internet fue producto de una construcción social, ya que heredó

⁴⁰ Existe una lista completa de aplicaciones y sus descripciones en <http://www.insna.org/software/index.html>

⁴¹ Información y adquisición del programa en forma gratuita para uso no comercial: <http://pajek.imfm.si/>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

las prácticas de sus creadores y primeros usuarios, caracterizadas por la experimentación e investigación en grupos, así como la revisión entre pares del trabajo realizado.

Aún hoy, los primeros internautas (Castells, 2001) navegan en el ciberespacio haciendo lo que saben hacer: programas. Richard Stallman, Karl Fogel, Linus Torvalds, Vint Cerf, entre otros siguen desarrollando códigos para nuevos programas o mejorar los ya existentes, a través de pequeñas comunidades y/o grandes redes de producción.

Podemos definir a una comunidad en red como un grupo de personas que están mucho más conectadas entre sí de lo que lo están con otros grupos de personas conectadas en otras partes de la red (Christakis & Fowler, 2010).

Estas redes de intercambio de códigos y conocimientos prácticos están presentes en todo Internet, incrementando su número de usuarios y, con ello, mejorando el tipo y cantidad de programas libres y abiertos disponibles como alternativa al *software* privativo.

Las comunidades de desarrollo de software libre son un tipo particular de comunidad en línea, orientadas a la finalización de un producto (un programa de ordenador) y con herramientas bastante estandarizadas, tanto de comunicación como para el desarrollo del producto. Una característica clara que las define es la aportación

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

voluntaria de trabajo por parte de sus miembros, para posteriormente liberarlo para terceros bajo un esquema de licenciamiento de los considerados libres o de código abierto. (Fernández M., 2011: 7)

La organización en la producción de *software* libre y abierto está dada a través de la infraestructura virtual. Índices y repositorios o forjas son los sitios de Internet donde, tanto desarrolladores como usuarios, se encuentran intercambiando conocimiento, trabajo y retroalimentación. Fundamentalmente, en Internet, se pueden encontrar sitios que enlistan programas para descargar y la información sobre los mismos; algunos de estos sitios, incluso prestan su plataforma para desarrollar algunas aplicaciones, otorgando algunos servicios al equipo de programadores para este fin. Estos sitios son llamados repositorios –cuando apoyan a los desarrolladores con infraestructura virtual y servicios para cargar, controlar versiones, modificar, repartir tareas, etcétera-- e índices, que llevan a los interesados hacia los sitios y plataformas de los programas y equipos de programadores que buscan.

Weinberg observó que en los sitios donde los desarrolladores no eran territoriales sobre su código y motivaban a otras personas a encontrar 'bugs' (fallas) y mejoras potenciales en él, la mejoría sucedía dramáticamente más rápido que en ningún otro lugar. (E. Raymond, 1999: 20)

Si bien es cierto que gran parte del proceso de producción de aplicaciones requiere de conocimiento especializado, existen

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

muchos usuarios de FLOSS que, a través de la interacción con los programas y los desarrolladores de éstos, aún sin conocimientos específicos sobre programación, colaboran de manera importante en el desarrollo de componentes esenciales de aplicaciones.

Los usuarios son algo maravilloso para tener, y no sólo porque demuestran que estás atendiendo una necesidad, que estás haciendo algo bien. Cultivados propiamente ellos se pueden convertir en co-desarrolladores. (E. Raymond, 1999: 6)

La cooperación entre desarrolladores, entre éstos y usuarios, y entre usuarios entre sí, en la producción de FLOSS se da en varios niveles y en diferentes momentos. La producción de este tipo de programas y aplicaciones, como podrá verse, está basada en la comunidad de desarrolladores, pero también en la interacción de éstos con los usuarios y, en la infraestructura básica que proporcionan las forjas y repositorios para mantener eficazmente tanto a los productos, como a los flujos de comunicación en los procesos de desarrollo y mantenimiento.

3.3.1 La comunidad hacker

Como se mencionó en el capítulo 1, al desarrollar la historia del desarrollo de *software*, los *hackers* no son delincuentes de Internet que buscan robar información o romper candados de seguridad de empresas y gobiernos. Los que así lo hacen son rechazados por la cultura '*hacker*' y reciben el nombre de *crackers*. El término ha sido confundido y mal empleado en medios de comunicación y algunas publicaciones, lo que ha

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

derivado en una mala comprensión y aceptación de las personas que, por su conocimiento en programación, son catalogadas de este modo. Al respecto, Lizama Mendoza (2002) indica que:

La mayor parte de estos discursos en contra de los hackers provienen y se alimentan básicamente de dos fuentes: a) del sector empresarial (America On Line-Time Warner, Microsoft, Amazon, etc.) que tiene apostado una gran parte de su capital en las NTIC y que, para obtener rentabilidad, requieren convertir a la Internet en un mercado global de servicios y a los usuarios en consumidores aislados de información privatizada; y b) de una investigación académica que al no estar lo suficientemente ilustrada en la dimensión tecnológica del fenómeno, termina por realizar, sin quererlo, lecturas parciales sobre éstos. Ambos discursos son reproducidos por los medios masivos de comunicación, obteniendo un gran impacto en la opinión pública. (Lizama Mendoza, 2002: 93)

Hacker, para programadores y usuarios concientes de la historia de la computación e Internet, se define correctamente como “la persona que disfruta explorando los detalles de sistemas programables y en saber cómo expandir sus propias capacidades, en oposición a la mayoría de los usuarios que prefieren sólo aprender lo mínimo necesario.”(E. S. Raymond, 1992). Otras definiciones hacen hincapié en la capacidad para, y el gusto por, programar; la sabiduría y facilidad para hacerlo de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

un modo rápido y frecuente. En general, se enfatiza el placer intelectual de vencer en forma creativa las limitaciones y expandir el conocimiento y habilidades personales.

La clasificación como *hacker* implica la adhesión de la persona a la comunidad y alguna versión ética de esta cultura, lo que implica fundamentalmente no sólo el gusto por el desarrollo de código, sino también la necesidad de compartirlo. Ser parte de una comunidad *hacker* implica la aceptación de reglas tácitas y explícitas, como lenguaje utilizado para la interacción verbal y el intercambio de código, formas y lugares de participación según el grado de conocimiento en programación, etcétera, todo lo cual otorga un sentido de pertenencia.

Las comunidades *hackers* siguen en mayor o menor medida los ideales de libertad para compartir información y conocimiento. Las variaciones éticas entre estas comunidades se definirían por la radicalidad de sus posiciones con respecto a esta libertad y en contra del *software* privativo.

Casi todos los hackers desean activamente compartir trucos técnicos, software y (cuando es posible) recursos de cómputo con otros hackers. Redes de gran cooperación como Usenet, FidoNet e Internet pueden funcionar sin un control central por este trato; se apoyan y refuerzan un sentido de comunidad. (E. S. Raymond, 1992)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

El concepto de 'comunidad' como dice González Casanova, “plantea la cohesión de lo Uno y la interacción recíproca que se realiza sobre la base de simpatías y diferencias, de empatías y oposiciones [...] También corresponde a la unidad, al interés común que mueve a todos sus miembros, por encima de sus diferencias.” (González Casanova, 2004). En este sentido, la comunidad *hacker*, aunque con diferencias ideológicas al interior, emerge cohesiva de la interacción entre sus miembros con base al interés común sobre el desarrollo e intercambio de códigos.

Existe una comunidad, una cultura compartida, de programadores expertos y brujos de redes, cuya historia se puede rastrear décadas atrás, hasta las primeras minicomputadoras de tiempo compartido y los primigenios experimentos de ARPAnet. Los miembros de esta cultura acuñaron el término hacker. Los hackers construyeron la Internet. Los hackers hicieron del sistema operativo UNIX lo que es en la actualidad. Los hackers hacen andar Usenet. Los hackers hacen que funcione la www. Si usted es parte de esta cultura, si usted ha contribuido a ella y otra gente lo llama a usted hacker, entonces usted es un hacker. (Raymond en Lizama Mendoza, 2002: 95)

Sin duda, la historia del desarrollo de Internet es fundamental para entender la emergencia y construcción de las comunidades *hackers*, puesto que el espacio virtual posibilitó el intercambio

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

de conocimientos informáticos a una escala gigantesca. Como menciona Castells (2001: 64):

Internet es la base organizativa de esta cultura. En general, la comunidad hacker es global y virtual. Aunque se producen algunos encuentros presenciales, tales como fiestas, congresos y ferias, la interacción suele tener lugar electrónicamente.

Se puede hablar de una gran red de intercambio y desarrollo de *software* libre, misma que se divide en distintas comunidades, cuyas identidades o cohesión a las mismas está dada por el interés en programas o sistemas particulares. En este sentido, se pueden mencionar comunidades en varios niveles. El primero, más amplio, incluiría a las comunidades generadas a partir de *Linux*, como núcleo básico de distintas distribuciones –casi todos los usuarios y desarrolladores de sistemas que son llamados libres, están incluidos, puesto que *Linux* es el *kernel*⁴² de la mayoría de los sistemas operativos libres y abierto-. Un segundo nivel estaría caracterizado por la interacción generada entre los usuarios y programadores de distribuciones como *Ubuntu* o *Fedora*, o bien desarrolladores que prefieren trabajar en un repositorio/forja⁴³ específico, como *Sourceforge*, *GoogleCode*,

⁴² Kernel es el nombre que se le da al núcleo de un sistema operativo, es la parte que cumple las funciones más básicas e importantes en el ordenador, como la entrada al hardware.

⁴³ Un repositorio o forja es un sitio de Internet que proporciona diversas herramientas a los equipos de desarrolladores para realizar su trabajo y poder comunicarse entre sí. El término, así como las herramientas y formas de operación se describirán y explicarán en el subtema 3.2.3 de este capítulo.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

ObjectWeb, entre otros: en este nivel, las comunidades pueden dividirse por la preferencia en el uso de una forja o distribución específica, sea por su facilidad de manejo, su rapidez, sus foros y retroalimentación al usuario, etcétera. El tercer nivel de creación o emergencia de una comunidad, que puede estar empalmado con la preferencia en el uso de una forja, se generaría por el intercambio entre los desarrolladores de paquetes como *Debian*, o sistemas como *GNU*⁴⁴.

Las comunidades de software libre son extremadamente heterogéneas en su composición, existiendo comunidades locales que desarrollan adaptaciones, traducciones y localizaciones de algún otro producto (como Joomla hispano), comunidades que desarrollan un producto sectorial pero útil para multitud de países, como la comunidad Moodle24, o proyectos de tamaño extremadamente considerable y que abarcan no solo el desarrollo del software, sino el marketing, diseño gráfico e incluso protección legal, como GNOME o la propia Free Software Foundation. (Fernández M., 2011: 8)

Sea cual sea el nivel de agregación o desagregación, el estudio del fenómeno de producción de *software* libre muestra que los desarrolladores y usuarios del mismo conforman una red muy amplia en la que en cualquier momento puede generarse algún tipo de colaboración entre usuarios y desarrolladores. Esta

⁴⁴ Término por el que se le conoce al complemento de Linux, el sistema Gnu, cuyas siglas significan Gnu is not Unix, en referencia al rechazo de Richard Stallman por el cierre de Unix por AT&T en los 80.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

cooperación puede darse tanto en la producción de código, como en la retroalimentación, o bien, resolución de problemas de uso. La colaboración, así como puede darse de manera espontánea, igualmente se difumina, dando espacio a otro tipo de interacciones; es decir, cuando se habla de cooperación, no se quiere decir que ésta se da de manera continua a través de todos los actores involucrados; sin embargo, ocurre aleatoriamente de manera constante.

“Las comunidades” del siglo XXI, como las que forman el movimiento del software Libre, pueden organizarse desde larga distancia, pueden construirse en redes virtuales pueden, en sus nuevas dimensiones, transformar en reflejo real y multicolor el concepto de la comunidad humana.(Bollier en Helfrich, 2008)

La diferencia más clara hacia el interior de esta gran red es ideológica y difiere por la posición frente a la libertad y el uso de sistemas y programas privativos. Por el lado de la defensa radical de la libertad de uso, modificación, distribución y mejora está la comunidad reunida bajo la figura de Richard Stallman y la Fundación de *software* Libre, creada desde 1984. Las diferencias de Stallman para con el movimiento de código abierto se muestran en la siguiente cita:

[...] la retórica del «código abierto» se concentra en la posibilidad de crear un software de alta calidad y capacidad, pero rehuye las nociones de libertad, comunidad y principios [...] Los términos «software

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

libre» y «código abierto» describen más o menos la misma categoría de software, pero implican cosas muy distintas acerca del software y sus valores. El Proyecto GNU sigue empleando el término «software libre» para expresar la idea de que la libertad, y no sólo la tecnología, es importante. (Stallman, 2004: 43)

Bajo iniciativa de Bruce Perens y Eric S. Raymond en 1998, quienes buscaban una posición más neutral que la de Stallman para lograr alianzas con empresas, se creó el movimiento de código abierto, como escisión del de *software* libre. La iniciativa de código abierto, aunque defiende también la libertad, no censura el uso de componentes privativos en sistemas y programas y restringe la libertad de modificación y mejora del código fuente y, a veces, de distribución del mismo. Lo que buscan es ampliar la base de usuarios a una escala que permita mayor sustentabilidad. Como describen Lakhani y Von Hippel (2000):

El movimiento de fuente abierta se inició en 1998 por algunos prominentes hackers de computadoras, como Bruce Perens y Eric Raymond. Este grupo tenía algunas diferencias políticas con el movimiento de software libre, pero estaba de acuerdo en general con las prácticas de licencia que inició, y también tenía algunas nuevas ideas sobre cómo difundir estas prácticas de manera más amplia. (Lakhani & Hippel, 2000: 2)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

La división de las dos perspectivas, *software* libre y de fuente abierta, devino principalmente por el término “*free*” del movimiento de *software* libre. En inglés, “*free*” se refiere a libre, pero también a gratis, lo que no explica ni define correctamente la idea de libertad. Richard Stallman sorteó este problema bajo el lema: *Free as in freedom, not as in free beer*⁴⁵. Sin embargo, parte importante de la comunidad de usuarios y desarrolladores de *software* libre y abierto no quedó satisfecha. Además de la cuestión del término, muchos *hackers* buscaron adoptar una posición más neutral que la de Stallman para con las empresas informáticas.

La diferencia técnica entre las aplicaciones libres y abiertas, como se mencionó algunos párrafos arriba, tiene que ver con la restricción de algunas de las libertades que proclama el *software* libre. El *software* de código abierto permite el estudio del mismo, pero no la modificación y mejora del mismo o, en su caso, deja estipulado en sus licencias hasta qué punto el código fuente puede ser modificado y qué reglas deben seguirse para hacerlo. En el caso del *software* libre, las licencias estipulan que no hay condicionante alguna para trabajar con el código.

A pesar de estas diferencias, tanto el movimiento de *software* libre como el de fuente abierta tienen una idea en común: la libertad de acceso al código de programación. Este libre acceso implica la libertad de conocer y aprender del trabajo de otros, fomenta la interacción entre desarrolladores y la relación de éstos con los usuarios. A fin de cuentas, tanto el *software* libre

⁴⁵ Libre como en libertad no como en cerveza gratis (la traducción es propia)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

como el abierto construyen comunidades solidarias y defienden ideas afines. Esta ideología y su práctica son la base que sostiene al *software* libre y abierto como bien común.

3.3.2 Interacción entre desarrolladores y usuarios. La colaboración en línea

Las comunidades de producción de *software* libre están compuestas por desarrolladores y usuarios de aplicaciones. Todos los desarrolladores son usuarios, pero no todos los usuarios saben programar y desarrollar código. Puede ser que en el pasado la falta de conocimientos sobre programación fuera un obstáculo para que usuarios 'legos' o no experimentados adoptaran el *software* libre y abierto, pero actualmente, las aplicaciones, sobre todo las más usuales, son tan amigables como las privativas de uso comercial.

La comunidad *hacker* se dio cuenta de la necesidad de expandirse para sobrevivir, abriéndose a la participación de todo tipo de usuarios. Para lograr esta apertura fue necesario trabajar más en las interfaces y abrir los foros de discusión y retroalimentación a voluntarios que, con pocos o nulos conocimientos sobre programación, aportaran sus conocimientos en rubros como la traducción y redacción de manuales. Los usuarios de FLOSS no necesariamente aprenden a programar; sin embargo, como se ha dicho a lo largo de este trabajo, pueden hacerlo al tener acceso al código fuente y a los innumerables canales de charla (IRC) donde se reúnen programadores, simpatizantes y usuarios de las distintas comunidades de la red de producción e intercambio de *software* libre.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Estos canales, al estar abiertos a todos los usuarios y estar disponibles en cualquier distribución de cualquier parte del mundo, ofrecen un espacio de servicio y apoyo las 24 horas del día, los 365 días del año. En esos espacios, así como por el correo electrónico o en los foros de discusión, cualquier persona puede pedir ayuda para la resolución de un conflicto en su sistema. La ayuda puede o no ser inmediata, pero conociendo cómo y dónde preguntar, cualquier problema puede ser resuelto. No es necesario que, como en el *software* privativo, los conflictos lleguen a una 'masa crítica'; es decir, se repitan determinadas veces para ser tomados en cuenta. La ayuda puede venir de los desarrolladores principales de un programa, o bien, de otro usuario. Estas dinámicas de apoyo son parte del mantenimiento horizontal de la red de intercambio y producción de FLOSS.

Una red de intercambio es un caso especial de red social en la que los nodos de la red son actores sociales (personas o grupos) y las líneas que conectan a los nodos son posibles interacciones entre actores particulares, que pueden ocurrir durante el proceso de intercambio social. (Friedkin, 2004)

El intercambio de conocimientos entre desarrolladores y usuarios, y usuarios entre sí, no sólo ayuda a la resolución de conflictos individuales, sino que se convierte en el motor de un gran proceso de innovación tecnológica constante.

Estas redes horizontales de innovación tienen una gran ventaja sobre los sistemas de desarrollo de innovaciones de fabricación centralizada, que han

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

sido el mayor sostén del comercio por cientos de años: posibilitan a cada entidad para desarrollar exactamente lo que quiere, en lugar de estar restringidos a las opciones disponibles del mercado [...] Más aún, los usuarios individuales no tienen que desarrollar todo lo que necesitan por sí mismos: se pueden beneficiar de las innovaciones desarrolladas por otros, que son libremente compartidas dentro y más allá de la red de usuarios. (E. Von Hippel, 2007: 2)

La retroalimentación de usuarios 'comunes' y usuarios-programadores a los desarrolladores de las aplicaciones, y la posibilidad de ingresar al código para depurar errores, ha hecho posible una forma de producción horizontal que ha demostrado ser eficiente y efectiva. Esta producción al estilo bazaar (E. Raymond, 1999), inducida por la publicación temprana del código de Linus Torvalds a principios de los 90, se ha replicado en la mayoría de los procesos de producción de las aplicaciones y programas libres.

Eric Raymond en “La Catedral y el Bazaar” (1999) expone algunas reglas para la producción de FLOSS que ilustran las dinámicas de interacción y la ideología que subyace a estos procesos. El cuadro del lado derecho, muestra una selección de estas reglas:

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Lo que se ejemplifica en estas reglas es la importancia que se da al desarrollo conjunto y a la participación de los usuarios como pares para la retroalimentación y avance en la evolución y mejoramiento del código. Los desarrolladores escriben el código de una aplicación para adaptarla, complementarla y poder trabajar de forma integrada. Se puede decir que un programa de FLOSS es un “texto colectivo”, en el que los párrafos son las líneas de código que, articuladas de modo coherente, no sólo cuentan una historia, sino que construyen una herramienta.

Hay una idea fundamental sobre el intercambio solidario y la administración horizontal del proceso productivo. El tipo de dinámicas se

- 1) SI PIERDES INTERÉS EN UN PROGRAMA, HERÉDALO A UN SUCESOR COMPETENTE.
- 2) TRATAR A TUS USUARIOS COMO CO-DESARROLLADORES ES LA MEJOR RUTA PARA MEJORAR RÁPIDAMENTE EL CÓDIGO Y DEPURAR DE MODO EFECTIVO.
- 3) LIBERA TEMPRANO Y SEGUIDO, Y ESCUCHA A TUS USUARIOS.
- 4) SI LA BASE DE CO-DESARROLLO Y DE PROBADORES-BETA ES AMPLIA, CUALQUIER PROBLEMA SE CLASIFICARÁ RÁPIDAMENTE Y SU ARREGLO SERÁ OBVIO PARA ALGUIEN.
- 5) SI TRATAS A TUS PROBADORES-BETA COMO TU RECURSO MÁS VALIOSO, ELLOS RESPONDERÁN CONVIRTIÉNDOSE JUSTO EN ESO.
- 6) LA SIGUIENTE MEJOR COSA DE TENER BUENAS IDEAS ES RECONOCERLAS EN TUS USUARIOS.
- 7) SI EL COORDINADOR DE DESARROLLO TIENE UN MEDIO, DE MENOS, TAN BUENO COMO INTERNET, Y SABE CÓMO GUIAR SIN COERCIÓN, MUCHAS CABEZAS SON INEVITABLEMENTE MEJOR QUE UNA SÓLA.

*Texto 1: Reglas de producción de FLOSS
(Fuente: Eric S. Raymond: 1999)*

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

establecen por los flujos de información entre desarrolladores y usuarios. Esta forma de trabajo ejemplifica lo que Harris llama redes colaborativas:

Las redes colaborativas, algunas veces llamadas alianzas estratégicas, reemplazan la tradicional relación contractual entre participantes por un compromiso mutuo para aplicar los recursos de uno en favor de lograr una meta común. La característica esencial es que los arreglos se llevan a cabo para ventaja mutua, pero no involucran propiedad directa por una parte o las otras, y se extienden más allá del simple contrato de compra-venta o acuerdo de diseño-construcción. (Harris, 2007)

Estas comunidades de producción en redes de usuarios y desarrolladores también involucran empresas y fundaciones que apoyan con infraestructura y recursos la producción de FLOSS. Empresas como Red-Hat o Cygnus apoyan el *software* libre y abierto con recursos económicos para desarrollo de nuevas aplicaciones y programas. Su ganancia está en el ofrecimiento de servicios de mantenimiento y soporte a otras empresas. Otros tipos de actores o empresas son los repositorios o forjas –que se describirán en el siguiente subtema-, que dotan de la infraestructura virtual a los programadores para el desarrollo de programas. La emergencia de las redes de desarrollo apoya los fundamentos del interaccionismo estructural sobre las relaciones entre actores y cómo éstas devienen en estructuras dinámicas. El análisis de redes sociales, a través de sus herramientas, permitirá

observar estas interacciones dentro de la infraestructura virtual de los repositorios.

3.3.3 Proyectos, Programas y Repositorios/Forjas

La participación de programadores en diferentes proyectos se hace posible por la existencia de sitios específicos de desarrollo de *software*. Estos son los repositorios que se han estado mencionando en este estudio. Estos sitios son plataformas virtuales de organizaciones, fundaciones o empresas que, para desarrollar una aplicación o programa, dan espacio en servidor y proporcionan distintos servicios.

Un repositorio (forja) provee varias herramientas y facilidades para distribuir los equipos de desarrollo, como sistemas de control de código, listas de correo y foros de comunicación, sistemas de búsqueda de errores, espacio virtual, etcétera [...Además] – escritura de código y documentación, distribución de nuevas versiones—de un modo que cubre las necesidades de grupos de desarrolladores dispersos (en tiempo y espacio). (Squire & Williams, 2012)

En un principio, estos sitios se generaron por las propias comunidades de desarrollo de *software* libre y, por tanto, sólo alojaban este tipo de aplicaciones y programas. Sin embargo, las herramientas que proporcionan estos espacios han demostrado funcionar de manera efectiva en tareas de coordinación, por lo que ahora muchos repositorios son privativos o, los que antes

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

eran sólo libres/abiertos, ahora aceptan programas de código cerrado.

Para el desarrollo de un programa es necesario tener comunicación a varios niveles entre los involucrados en esta tarea, así como espacio para subir y descargar avances en el trabajo. Al mismo tiempo, se necesita que exista un control sobre la división del trabajo para que no se dupliquen tareas y se trabaje siempre sobre las partes del código más actualizadas. Todo esto requiere de espacio –igual que cuando se trabaja en una computadora se necesita espacio para guardar archivos de texto, música y video--, pero este espacio debe estar abierto a que varios actores accedan desde distintos lugares a su contenido, y que esto no sature el sitio; es decir, se necesita de un espacio virtual lo suficientemente grande para hacer todas estas tareas. En algunos casos, los programas tienen su propio sitio virtual que ofrece la infraestructura necesaria para soportar la cooperación en línea.

[...] proyectos grandes (GNOME, KDE, Apache, Eclipse, OpenOffice.org, Mozilla, etc.) mantienen su propio repositorio, frecuentemente con software a la medida. Sistemas similares se utilizan también en ambientes corporativos para el desarrollo de software no-libre. (Fernandez et al., 2011)

Sin embargo, no todos los proyectos tienen la capacidad para mantener un sitio que soporte foros, listas de correo y espacio para los componentes que se irán produciendo, depurando, evaluando y liberando. Un espacio virtual con esas

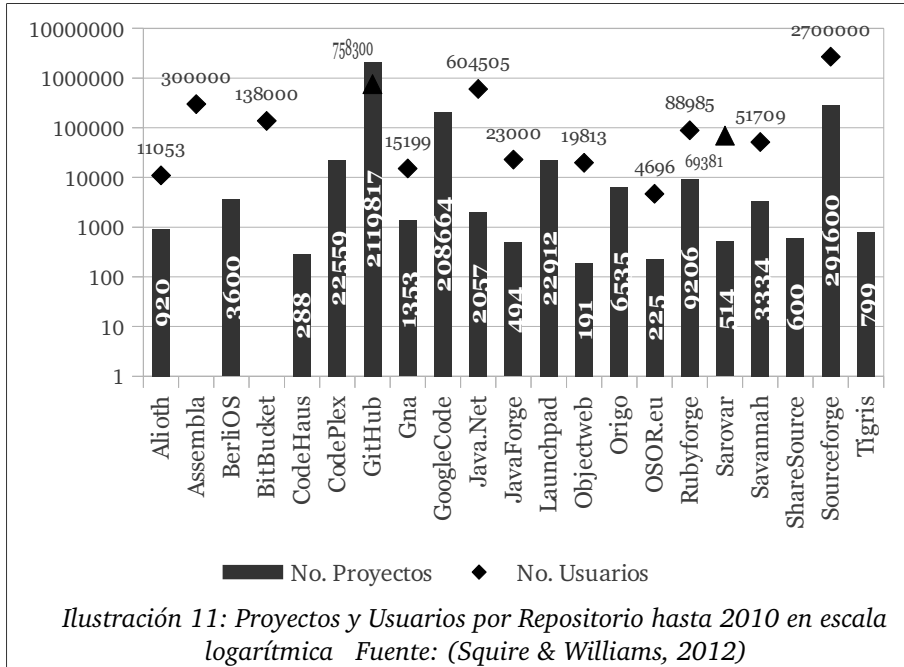
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

características requiere el pago de un dominio, una administración definida a largo plazo y garantías de mantenimiento y permanencia. En estos casos, los desarrolladores iniciales acuden a repositorios, que son sitios que ofrecen esta infraestructura. También sucede que los mismos repositorios inician el desarrollo de un programa, invitando a otros programadores a colaborar colectivamente en un momento dado.

Estos repositorios actúan hospedando proyectos para que los desarrolladores puedan colaborar en línea y trabajen en red. El número de proyectos y de usuarios varía, lo que depende del tipo de proyectos que hospedan, ya que pueden especializarse en tipos de programas, si éstos son libres o no, lenguajes de programación, sistemas, etcétera.(Fernandez et al., 2011)

El gráfico que se presenta a continuación muestra 21 repositorios que hasta 2010 se encontraban en pleno funcionamiento. Las diferencias de los repositorios en tamaño, según cantidad de desarrolladores y/o proyectos, son amplias. Esto es así porque cada repositorio cuenta ya con una cierta popularidad entre las comunidades y, como se ha mencionado, sus propias características de especialización en programas, lenguajes, o servicios, así como espacio de alojamiento para más o menos proyectos.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

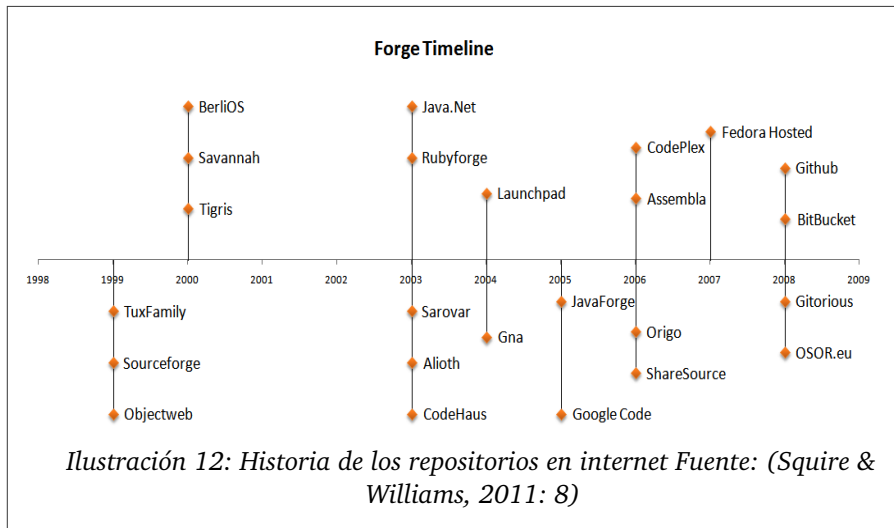


Los repositorios se hicieron populares entre los desarrolladores en 1999. En ese momento, las principales plataformas eran Sourceforge, TuxFamily y Objectweb. Con los años, han aparecido y desaparecido repositorios y otros han continuado, por la generación de nuevas comunidades y la desaparición de otras, por la capacidad para obtener recursos o no, etcétera. Como dice Fernández, el diseño y estructura de estas plataformas ha cambiado realmente poco desde finales de los 90, simplemente se ha ajustado mejor para el trabajo en colaboración:

La arquitectura básica y servicios de estos

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

repositorios ha evolucionado muy poco durante la última década, resolviendo algunas de las necesidades de los equipos de desarrollo y probando una buena adaptación para escalar bien hacia las decenas de miles de proyectos y cientos de miles de desarrolladores y cientos de millones de líneas de código. (Fernandez et al., 2011)



Los programas hospedados, a veces no sólo están en un repositorio: pueden dividirse los componentes entre diversos sitios, o cambiarse de uno a otro a través del tiempo de desarrollo. La migración y/o división de componentes responde a dinámicas específicas de cada equipo de desarrollo; sin embargo, a veces, estas divisiones responden a conflictos dentro del equipo de programadores que generan la bifurcación de las propias aplicaciones. En algunas ocasiones, las divisiones generadas por conflictos dan lugar a versiones diferentes de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

programas, teniendo como base el mismo código fuente. Como lo define Fernández M.(2011):

Un fork o bifurcación consiste en la creación de una nueva versión y comunidad de un proyecto de desarrollo de software libre ya existente empleando el código fuente del proyecto original [...] El famoso glosario “Jargon File” comenta que esto es debido tanto a la pérdida de recursos como a la acritud y al conflicto generado entre los miembros de la comunidad.(Fernández M., 2011)

Las situaciones de bifurcación son posibles por la propia naturaleza del *software* libre y abierto: la libertad de acceder y usar el código fuente. La bifurcación puede ocurrir por diferencias en el equipo sobre la arquitectura, la forma de organización y/o el tipo de licencia que se quiere seleccionar. Es una amenaza para los programas en desarrollo, pero también un elemento que lleva a la innovación y la diversidad de aplicaciones libres y abiertas. De hecho, en algunos repositorios, como GitHub, se estimula la bifurcación, ofreciendo servicios de seguimiento entre las divisiones. Esto sucede porque muchas bifurcaciones han resultado exitosas, como se menciona a continuación:

Forks notorios y de resultado han sido el proyecto X.org, sustituyendo a Xfree, Mozilla Firefox de Mozilla, Xvid de Open Divx así como un gran número de distribuciones de BSD. Es novedad, por ejemplo, el fork de OpenOffice en un nuevo proyecto

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

llamado Libreoffice por disconformidad de miembros destacados de la comunidad con el manejo de Oracle de la comunidad desde la adquisición de Sun por parte de la misma. (idem: 10)

La bifurcación y la migración constante es un problema cuando se quiere estudiar la dinámica de producción en las comunidades y redes de FLOSS, pues entre los repositorios hay diferencias en la terminología para etiquetar y clasificar procesos y actores, además de que el mismo proyecto puede, repentinamente, variar su nombre.

Contar la historia de un proyecto de software a través del tiempo requiere que seamos capaces de seguir al proyecto a través de las forjas y de todos estos cambios. (Squire & Williams, 2012)

Los repositorios contienen más o menos información sobre los proyectos que alojan, lo que hace posible ver los datos y obtener estadísticas sobre los mismos. Sin embargo, el análisis de datos de estos repositorios, en general, es complicado por la cantidad de datos alojados y la clasificación de los mismos. Sourceforge, uno de los repositorios más populares en la red recaba datos que, posteriormente, envía al laboratorio de cómputo de la Universidad de NotreDame. La Universidad depura y clasifica los datos para presentarlos de forma más accesible a través de consultas de base de datos.⁴⁶

⁴⁶ El esquema de los datos de Sourceforge. Net, alojados en el sitio de la Universidad de NotreDame puede consultarse en: http://zerlot.cse.nd.edu/mediawiki/index.php?title=ER_diagrams

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

La cantidad de datos es enorme y, si tomamos en cuenta que estos datos están recabados cada dos meses, la extensión y complejidad para el análisis crece de manera importante. El fenómeno de producción de FLOSS ha suscitado el interés creciente de investigadores y científicos en todo el mundo, por lo que el acceso a los datos y la posibilidad de utilizarlos para un análisis formal se ha convertido ya en un asunto de importancia y estudio dentro de la comunidad científica.⁴⁷

La complejidad en la presentación de datos en los repositorios y el interés por estudiar el fenómeno han llevado a la creación de otros sitios, calificados como repositorios de repositorios, que exponen los datos de varios sitios de desarrollo, intentando hacerlo de un modo más depurado y comprensible para el estudio de la producción de *software* libre.⁴⁸ A continuación, presento el esquema de datos del repositorio de repositorios FLOSSmole, que fue utilizado para esta tesis. En este trabajo, se utilizaron las bases de datos, alojadas en el sitio referido, de los siguientes repositorios: *GitHub*, *RubyForge*, *Launchpad*, *ObjectWeb*, *Savannah* y *Tigris*.⁴⁹

Los procedimientos que se utilizaron para manejar y seleccionar los datos se explicarán en el siguiente capítulo, donde, además,

⁴⁷ Ver <http://flosshub.org/biblio>

⁴⁸ Ver <http://flossmole.org/> y <http://flossmetrics.org/>

⁴⁹ El esquema de los datos de estos repositorios, alojados en el sitio Flossmole.org puede consultarse completo en <http://flossmole.org/content/database-schema-0>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

se presentan los resultados preliminares del análisis de redes y de las motivaciones de los actores para colaborar.

Capítulo 4 Las redes de producción de *software* libre y abierto. Interacción, motivaciones y perspectivas

Los capítulos precedentes han servido de marco explicativo del surgimiento, clasificación y descripción de los actores e infraestructura de desarrollo del *software* libre y abierto. Como se vio, las prácticas de intercambio y producción abiertas a la participación colectiva, además de las estrategias de defensa a la privatización, derivan en que se le clasifique como un bien común. Estas prácticas de intercambio de conocimientos entre desarrolladores son producto de formas de organización social previa a la emergencia de Internet y, por ello, se convirtieron en una estructura tradicional de la cultura *hacker* en la red de redes.

En el espacio virtual, la colaboración pudo ampliarse y hacer más eficientes sus procesos de revisión y aportación voluntaria dentro de plataformas de apoyo y servicios para el desarrollo de código. Los repositorios se convirtieron en la infraestructura básica para el desarrollo colectivo de aplicaciones libres y abiertas, pues facilitan la división del trabajo, los procesos de revisión y control, así como la liberación temprana de los programas a una base de usuarios dispuesta a experimentar

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

sobre ellos y dar retroalimentación a los equipos de desarrolladores.

Tomando en cuenta todo lo que se ha expuesto en esta tesis, en este capítulo desglosaré y comentaré los hallazgos encontrados desde el marco teórico-metodológico seleccionado para responder a las preguntas rectoras de la investigación. Finalmente, como este fenómeno de producción y distribución de *software* libre y abierto está inserto en un contexto histórico, económico, político y social, también expondré los retos y perspectivas a futuro de este tipo de organización del trabajo y las posibilidades de su replicación en otros sectores pues, como dice Mas i Hernández (2005):

El modelo de producción del software libre ha demostrado tener un éxito considerable y ser altamente productivo en el desarrollo de programas tan complejos como Linux o Apache. Más allá del software, proyectos como Wikipedia o Open Directory muestran que este modelo es exportable a otros tipos de creaciones intelectuales. (Mas i Hernández, 2005: 93)

Este capítulo se dividirá en cuatro apartados principales que cubren la guía metodológica, la descripción de las dinámicas de producción de FLOSS en los repositorios, la perspectiva de actores clave y una visión general de los retos a los que se enfrentan estas aplicaciones en el contexto tecnológico de inicios del siglo XXI.

4.1 Guía metodológica

Para mostrar la interacción entre los desarrolladores y usuarios de *software* libre y abierto en la producción del mismo, se eligió utilizar el análisis de redes sociales bajo la perspectiva del interaccionismo estructural, pues permite visualizar la estructura de las comunidades y redes que participan en el fenómeno observado. Los datos analizados fueron recabados del sitio Flossmole.org sobre siete repositorios de internet, a saber: *GitHub*, *Launchpad*, *ObjectWeb*, *RubyForge*, *Tigris* y *Savannah*.

El análisis de los datos de los repositorios se complementa con un análisis interpretativo de las motivaciones para la colaboración, basado en una entrevista presencial a Richard Stallman y entrevistas recabadas en línea, artículos y textos seleccionados de algunos personajes reconocidos en las comunidades y redes de FLOSS, como: Lawrence Lessig, Sir Tim Berners-Lee, Mark Shuttleworth, Eric Raymond, Karl Fogel y Eben Moglen.

Para este análisis, utilicé principalmente: estadística descriptiva y análisis de redes sociales, dos entrevistas a profundidad⁵⁰ e investigación documental para seleccionar entrevistas en línea, artículos y textos de personajes destacados del fenómeno FLOSS, y observación participante.

⁵⁰ Las entrevistas se realizaron a Richard Stallman y a “La Morsa”, aunque la segunda sirvió para profundizar en el conocimiento de la producción de software, no se tomó en cuenta para el análisis de motivaciones, ya que el entrevistado indicó nunca haber participado colectivamente en un proyecto de software libre.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En este apartado, describiré los procedimientos utilizados para construir las gráficas visuales con los datos recabados, y cómo éstos fueron base del trabajo posterior para refinar y profundizar en la observación y análisis de las interacciones.

4.1.1 *Software libre a la luz del análisis de redes sociales (ARS)*

El análisis de los datos y el desarrollo de los gráficos de redes fue posible por el uso de aplicaciones especiales para manejo de bases de datos y construcción de redes. Me fue necesario destinar un tiempo considerable en el aprendizaje para el uso y manejo de estos programas, lo que, a su vez, se convirtió en parte del trabajo de observación participante.

En el caso los datos de desarrolladores y programas en los distintos repositorios⁵¹, se utilizaron principalmente dos programas de administración de bases de datos, que ayudan a organizar, seleccionar, depurar y relacionar los mismos: *MySQL* y *Microsoft Access*, el primero es una aplicación libre y abierta, mientras que el segundo no. La complejidad para aprender el lenguaje computacional necesario para utilizar solamente el programa libre, *MySQL*, en un tiempo restringido a la entrega de los avances formales de la tesis, fue decisivo para utilizar el *software* privativo.

⁵¹ Como se mencionó previamente, los repositorios fueron *GitHub*, *Launchpad*, *ObjectWeb*, *RubyForge*, *Tigris* y *Savannah*. Más adelante, en este capítulo, se describirán los mismos y se explicará el motivo de esta selección.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Pude notar prácticamente que, aunque el programa privativo me facilitaba la mayoría de los procesos para la vinculación de los datos, al mismo tiempo, restringía las posibilidades de lo que yo podía obtener de los mismos datos; es decir, sólo me facilitaba aquellos procesos comunes que estaban incluidos como posibilidad para realizar diferentes consultas. Si yo quería realizar consultas más complejas, debía aprender a utilizar el lenguaje *SQL* de cualquier modo, lo que me hubiera llevado a usar la aplicación *MySQL* desde el inicio. Lo anterior es una muestra de cómo, si bien es cierto que el *software* privativo en muchos casos es “más amigable” (amigable en el sentido de que contiene algunos procesos que pueden hacerse más fácilmente sin tener que aprender un lenguaje lógico específico), sí es más restrictivo.

Para la generación visual y el análisis de las redes, se utilizó el programa libre *Pajek*, lo que fue posible gracias al Mtro. Alejandro Arnulfo Ruiz León del Laboratorio de Redes de la UNAM y profesor de la misma materia en el posgrado de la Facultad de Ciencias Políticas y Sociales.

4.1.1.1 Observaciones preliminares

Los datos de los repositorios se obtuvieron del “repositorio de repositorios” FLOSSmole, que es una de las más grandes bases de datos sobre producción de *software* libre y abierto en internet. Este sitio⁵² permite la descarga de datos libremente después de solicitar vía correo electrónico el acceso. Los datos que se descargan están relativamente organizados, para efectos

⁵²<http://flossmole.org/>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

de esta investigación, en listas de desarrolladores y listas de proyectos, así como un tercer listado que relaciona los primeros a los segundos. La base de datos se maneja a través del administrador *MySQL*, lo que permite el manejo de grandes cantidades de datos.

Recabé los datos que corresponden, en el caso de los seis repositorios analizados, a los últimos meses (noviembre y diciembre) de 2011. Sólo analicé para esta tesis un corte de tiempo por la gran cantidad de datos a observar y analizar. Un análisis en el tiempo sobre la dinámica de las redes permitiría observar a profundidad los cambios en las interacciones, lo que sería conveniente para lograr conclusiones más generalizadas. Sin embargo, este trabajo es un análisis inicial que muestra la posibilidad de observación y descripción de estas redes en varios niveles, proponiéndose como base para un estudio más exhaustivo a largo plazo.

La mayoría de las redes que generé con los datos obtenidos muestran una gran cantidad de desarrolladores con un sólo proyecto, y proyectos con un sólo desarrollador; es decir, parecería que los desarrolladores, en general, trabajan solos y no en colaboración. Sin embargo, es importante mencionar que, al estudiar a profundidad las bases de datos y sus categorías, encontré que:

- Los datos a los que se tiene acceso cuando se consultan “proyectos”, como se define en los repositorios, se refieren a una parte o componente de un programa o aplicación; es decir, cada programa contiene en el proceso de desarrollo una variedad de proyectos. Así, un proyecto

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

puede referirse a la interface de una aplicación, al desarrollo o mantenimiento del foro de retroalimentación, a las diversas traducciones, a un componente para su compilación, etcétera.

- Los componentes del programa se dividen en proyectos, y éstos no se desarrollan en un mismo momento del tiempo o tienen distintos grupos de trabajo que, a veces, trabajan vinculados, y a veces no. El desarrollo y mantenimiento de un programa o aplicación puede extenderse por amplios períodos de tiempo, p.e. *GNU-Linux*.

Lo anterior implica que, al tener un corte de dos meses en el tiempo, no se puede saber hasta hacer un análisis de varios períodos, si un componente, referido como X proyecto, es desarrollado por solamente una persona en cada momento, o bien la colaboración es por transferencia del trabajo realizado hacia aquel programador que continúa con otra parte del mismo componente.

En todos los repositorios, existe un número considerable de desarrolladores que tienen más de 2 proyectos, lo que confirma lo anteriormente mencionado, pues es muy probable que los participantes en un componente de programa, también participen en otro componente del mismo.

Así pues, al observar los gráficos y analizar los datos debe entenderse que:

- a) los desarrolladores y proyectos forman parte de una red amplia de desarrolladores de *software* libre y abierto,

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- b) los desarrolladores, a su vez, forman parte de la red de programadores de cada repositorio, pues han elegido éste para interactuar con pares⁵³,
- c) los proyectos son partes de los componentes de aplicaciones o programas que, por ahora, no se ven vinculados, pues sólo se muestra un corte en el tiempo; sin embargo, lo están.

4.1.1.2 Procedimientos

1. Observé los datos organizados y alojados en el sitio FLOSSmole para seleccionar seis repositorios, a saber: *GitHub*, *Launchpad*, *ObjectWeb*, *RubyForge*, *Tigris* y *Savannah*. Quedaron fuera del análisis los datos de *Sourceforge.net*, *UDD*, y *Google Code*, debido a que son repositorios con una cantidad enorme de datos que no permiten un manejo fácil con computadoras personales de memoria *RAM* estándar. Otros repositorios quedaron fuera, como *Alioth*, puesto que no otorga datos de desarrolladores, impidiendo la vinculación con proyectos, así como los datos de *FreshMeat* y de la *Free software Foundation*, puesto que éstos sitios son más bien directorios de programas; es decir, sitios que informan y

⁵³ Cabe mencionarse que algunos desarrolladores trabajan en varios repositorios, lo que convertiría a éstos en actores vinculantes entre repositorios. Algunos de estos actores se identificaron azarosamente durante el análisis. Un estudio más exhaustivo sería necesario para presentarlos de manera sistematizada.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

llevan con vínculos hacia el lugar donde se encuentran alojados los proyectos, pero no los alojan.

2. Descargué y sistematicé los datos de todos los desarrolladores y proyectos para cada uno de los seis repositorios en un corte de tiempo (noviembre-diciembre 2011)
3. Comparé y complementé los datos entre repositorios, generando gráficos de estadística comparativa.
4. Por cada repositorio, vinculé desarrolladores a proyectos para generar cuadros de datos en el programa Access, que me permitió relacionar y trabajar los datos.
5. Para identificar redes hacia el interior de los repositorios, busqué y seleccioné y aquellos desarrolladores que trabajaron en 3 proyectos o más, dentro de proyectos que involucraban a 2 o más programadores. Esto me permitió reducir los datos y encontrar aquellos que, para ese momento del tiempo, parecían estar más vinculados por su participación en diversos componentes de programa.
6. Generé tablas de relación entre proyectos y desarrolladores con las características mencionadas las preparé para su utilización en Pajek⁵⁴ como datos para redes de dos modos.⁵⁵

⁵⁴ La preparación del archivo para Pajek se realizó en hojas de cálculo, luego se trasladaron los datos a archivos .txt (bloc de notas)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

7. Dentro de Pajek, generé los gráficos visuales de redes con los que trabajé en la descripción de las dinámicas de las distintas interacciones presentadas.

En cuanto a la entrevista realizada y las que fueron recabadas en línea de personajes destacados para este estudio, así como conferencias y documentos recuperados de y sobre el fenómeno FLOSS, analicé, buscando en el discurso, las motivaciones para la cooperación, comparando los resultados con estudios previos. El procedimiento fue el siguiente:

1. Realicé una entrevista a profundidad y recolecté dos más, además de notas, artículos y libros de personajes reconocidos en las comunidades de FLOSS, como son: Richard Stallman – fundador de la Fundación de *software* Libre, desarrollador del proyecto *GNU* y creador del concepto Copy-left. Es el principal defensor del *software* libre en el mundo-, Linus Torvalds – creador del kernel Linux, base de la mayoría de los sistemas operativos libres; miembro directivo de la Linux Foundation-, Lawrence Lessig –creador de las licencias Creative Commons-, Sir Tim Berners-Lee –inventor del protocolo que hace funcionar internet: www (world wide web), así como de la liberación de los protocolos http y html-, Mark Shuttleworth – fundador de Canonical, empresa de la que depende el mantenimiento y apoyo para la distribución libre Ubuntu-Linux-, Eric Raymond –miembro directivo de la Fundación Linux, conocido además de por sus habilidades como

⁵⁵ Una red de dos modos es aquella cuyos nodos corresponden a dos tipos de actores (entidades) distintas; en este caso, los nodos corresponden tanto a desarrolladores como a proyectos. Las líneas que vinculan a los nodos indican la participación de un programador en un proyecto específico.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

desarrollador, por sus libros “La Catedral y el Bazaar” y “El diccionario *hacker* (Jargon File)”, Karl Fogel –socio de Open Tech Strategies, LLC., que apoya a organizaciones en la utilización y participación en proyectos de fuente abierta, y colaborador en QuestionCopyright.org para promover el uso de licencias libres. Ha participado en muchos proyectos libres populares, como GnuEmacs, Launchpad, entre otros--, Eben Moglen –profesor en la Universidad de Columbia; fundador del Centro de la ley por el *software* Libre.

2. Seleccioné estudios académicos previos sobre las motivaciones de los desarrolladores para trabajar en colaboración (K. R. Lakhani & Wolf, 2005; E. Von Hippel, 2007; Weber, 2004)
3. Seleccioné frases y resultados de los estudios y entrevistas y las clasifiqué en cuatro categorías conceptuales: creatividad, identidad social, identidad política y reproducción/sustentabilidad, con las que generé un esquema cognitivo.

4.2 Dinámicas y procesos en las redes y comunidades de producción digital de *software* libre. Análisis comparativo de seis repositorios en internet

Desde 2004, el sitio de internet FLOSSmole⁵⁶ recoge y comparte datos sobre los proyectos de *software* libre y abierto para su descarga gratuita en diferentes formatos. Estos datos están integrados y organizados en una gran base, administrada por los creadores de este gran repositorio de repositorios

⁵⁶Flossmole.org

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

(Howison, Conklin y Crowston) sobre una plataforma *MySQL*. Según la información proporcionada en su página de internet, para marzo de 2012, este sitio contaba con cerca de 1 Terabyte de datos desde 2004, hasta entonces, que incluyen a más de 500,000 proyectos diferentes y sus desarrolladores. (Ver Howison, Conklin, & Crowston, 2006)

Este sitio no sólo aloja los datos de los repositorios e índices de FLOSS, sino también vincula artículos que hablan sobre el fenómeno desde diferentes puntos de vista. Squire y Williams (2012) realizaron un estudio de 24 repositorios, clasificándolos según el tipo de artefactos y servicios que ofrecen.

Un repositorio, como se mencionó en el capítulo 3, es un sitio de Internet que ofrece la infraestructura virtual, tanto espacio para alojamiento como servicios de apoyo, para el desarrollo de un proyecto de *software*. Entre los servicios que ofrece se encuentran aplicaciones para búsqueda de fallas, listas de correo, foros, canales de charla, direcciones URL, entre otras.

Los repositorios pueden especializarse en algunos tipos de proyectos de software (sólo FLOSS, ciertos lenguajes de programación, ciertos lenguajes hablados). (Squire & Williams, 2012: 3)

Los autores del artículo referido analizaron los repositorios, así como estudios sobre los mismos y encontraron que la mayoría de éstos utilizaron datos del sitio *Sourceforge.net*⁵⁷. Sus

⁵⁷ Sourceforge.net es uno de los más conocidos repositorios de *software* en internet. Ver www.sourceforge.net

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

primeras observaciones indican que otros repositorios, también públicos, casi no han sido estudiados.

El aliciente para usar Sourceforge como fuente de datos es muy fuerte. Sucede esto a pesar de la existencia de otras múltiples fuentes de datos de proyectos auto-soportados y repositorios disponibles públicamente, a pesar de la existencia de colecciones públicas hechas para investigación de datos de Sourceforge, y a pesar de las advertencias sobre las fallas potenciales de usar estos datos para empezar. (Squire & Williams, 2012: 3)

El estudio que hicieron buscó documentar las razones por las que existe esta preferencia en el uso de los datos de *Sourceforge*, pero también cuáles serían las cualidades necesarias con las que debían contar éste y otros repositorios para ser utilizados en estudios académicos. Calificaron cuatro características básicas de los repositorios: a) que fuera fácil encontrar la lista de proyectos, b) que contaran con un gran número de proyectos para seleccionar, c) que se expusieran suficientes meta-datos de cada proyecto, y/o d) que estuvieran expuestos muchos artefactos para cada proyecto. Bajo estos criterios, encontraron que *Sourceforge* cumple con estos requisitos en lo general, pero también lo hacen *Google Code*, *Launchpad*, *Rubyforge*, y *CodePlex*.

Como parte de sus hallazgos, realizaron un diagrama que indica el tipo de servicios que otorgan, en lo general, los repositorios. El tamaño de cada una de las etiquetas del

4.2.1 Características de los repositorios

4.2.1.1 GitHub

Este repositorio fue fundado en 2008 por Chris Wanstrath, PJ Hyett y Tom Preston-Werner. Sus oficinas se ubican en San Francisco, EUA, pero tienen equipos colaborando en todo el mundo.⁵⁸ Según los datos proporcionados en su sitio, para finales de marzo de 2012, contaban con 1,478,382 personas trabajando en 2,493,988 de programas y aplicaciones. Su característica particular es que utilizan el sistema de control de versiones 'git'.

Este repositorio ofrece espacio y apoyo de manera gratuita a proyectos de *software* libre y abierto, pero también tienen paquetes con diferente costo a empresas privadas e individuos para desarrollar aplicaciones y programas privativos.

Entre sus servicios se encuentran wikis (foros de desarrollo), páginas web para cada proyecto, seguimiento gráfico de los proyectos y bifurcaciones de los mismos (mismas que fomentan y apoyan para posteriormente fusionar los mejores componentes y/o complementar los programas desarrollados), aplicaciones para seguimiento y comunicación entre el equipo.

Entre sus proyectos desarrollados o en mantenimiento están *Ruby on Rails*, *Iron Ruby*, *jQuery*, *Moodle*, *Diaspora*.

⁵⁸ <https://github.com>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En el período noviembre-diciembre analizado en esta tesis, el total de desarrolladores trabajando en este repositorio fue de 75,961 en 109,013 proyectos. El número de desarrolladores con más de dos proyectos fue de 11,651 (15% del total) en 50,102 proyectos (46% del total). De estos desarrolladores, aquellos que se encontraban colaborando con al menos otro desarrollador fueron 2,136.

4.2.1.2 Launchpad⁵⁹

Es una plataforma administrada por el equipo desarrollador de la distribución Ubuntu⁶⁰. Auspiciada por Canonical y fundada en 2004, ofrece servicios de identificación de fallas, hospedaje y revisión de código fuente, traducciones, listas de correo, seguimiento de preguntas y respuestas, construcción y hospedaje de paquetes Ubuntu, etcétera. En un principio, esta plataforma operó con código cerrado, pero en 2009 liberó su código fuente.

Según los datos proporcionados en marzo de 2012 en su página de internet, para ese momento se encontraban trabajando en el desarrollo de 26,903 proyectos; 1,895,556 traducciones; reparando cerca de 962,167 fallas; contestando a 190,787 preguntas y/o peticiones de soporte.

⁵⁹ <https://launchpad.net/>

⁶⁰ Ubuntu es una de las distribuciones, basada en el kernel Linux, más populares entre los usuarios de FLOSS y es producto de la empresa Canonical, dirigida por Mark Shuttleworth.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Entre sus proyectos más populares, además de la distribución Ubuntu, se encuentran *MySQL*, *BlueBream*, *GnomeDo*, *Drizzle*, *Inkscape*, *Bazaar*, *Launchpad*.

En el período noviembre-diciembre analizado en esta tesis, el total de desarrolladores trabajando en este repositorio fue de 6,021 en 8,400 proyectos. El número de desarrolladores con más de dos proyectos fue de 337 (6% del total) en 2,229 proyectos (27% del total). De estos desarrolladores, aquellos que se encontraban colaborando con al menos otro desarrollador fueron 29.

4.2.1.3 ObjectWeb⁶¹

Este repositorio fundado en 2007, ahora conocido como OW2, es el resultado de la fusión de ObjectWeb y Orientware. El primero inició sus operaciones en 2002, producto de los esfuerzos conjuntos entre las empresas europeas *INRIA*, *Bull* y *France Telecom*. Orientware, por su parte, fue generada en 2004 por la Universidad de Pekin, la Universidad de Aeronáutica y Astronáutica de Beijing, la Universidad Nacional de Tecnología de Defensa, La Compañía de Ingeniería de *software* CVIC, Ltd. y el Instituto de *software*, así como la Academia China de las Ciencias.

Según su propia definición, “el consorcio es una organización independiente sin fines de lucro, abierta a la participación de compañías, organizaciones públicas, academias e individuos”.

⁶¹<http://www.ow2.org/>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Entre sus objetivos y de manera destacada, promueven el uso y desarrollo de programas y aplicaciones de fuente abierta. Se autodefinen como una plataforma industrial que facilita la interacción entre los productores de código de fuente abierta y consumidores del mismo.

En el período noviembre-diciembre analizado en esta tesis, el total de desarrolladores trabajando en este repositorio fue de 828 en 146 proyectos. El número de desarrolladores con más de dos proyectos fue de 391 (47% del total) en 94 proyectos (64% del total). De estos desarrolladores, aquellos que se encontraban colaborando con al menos otro desarrollador fueron 85.

4.2.1.4 RubyForge⁶²

Desde sus inicios en 2003, este repositorio se ha especializado en hospedar y apoyar proyectos con base en el lenguaje de programación Ruby específicamente. Esta comunidad, para mantener el repositorio, fue apoyada por SunMicrosystems (ahora parte de Oracle) con *hardware*, Defender Hosting con la bancha ancha para el servidor principal, InfoEther, Inc. con soporte administrativo, GForge Group con la aplicación para la plataforma de control de versiones, y Bytemark. Como administradores de la plataforma están Kurt Dresner, Bruce Williams, Gregory Brown y Ben Bleything.

Hasta marzo de 2012, su página registraba 96,821 usuarios. Y 9,373 proyectos divididos en bases de datos, ambientes de escritorio, aplicaciones educativas y de oficina, juegos e internet,

⁶² rubyforge.org

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

multimedia, programas científicos, seguridad, desarrollo de *software*, terminales y aplicaciones para el sistema, entre otros.

En el período noviembre-diciembre analizado en esta tesis, el total de desarrolladores trabajando en este repositorio fue de 6,916 en 9,188 proyectos. El número de desarrolladores con más de dos proyectos fue de 1,052 (15% del total) en 4,426 proyectos (48% del total). De estos desarrolladores, aquellos que se encontraban colaborando con al menos otro desarrollador fueron 587.

4.2.1.5 Savannah⁶³

Apoyado por la Fundación de *software* Libre desde el año 2000, este repositorio se caracteriza fundamentalmente por el fomento a la producción de *software* libre específicamente. Sólo aceptan hospedar aplicaciones y desarrollos que sean 100% libres; es decir, que no contengan ningún componente privativo o semi-privativo en el código fuente.

Es la plataforma oficial de desarrollo, mantenimiento y distribución del sistema *GNU*, iniciado y encabezado por Richard Stallman. Para proyectos que no tienen que ver con este sistema, el sitio ofrece otra dirección electrónica: <http://savannah.nongnu.org> Utilizan los sistemas de control de versiones *git* y *cg* principalmente.

Los datos presentados en su página en marzo de 2012 registraban 55,663 usuarios y 3,428 proyectos, de los cuales

⁶³<http://savannah.GNU.org/>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

2,931 no pertenecían al proyecto *GNU*. Entre sus proyectos más populares se encuentra Emacs, QEMU.

En el período noviembre-diciembre analizado en esta tesis, el total de desarrolladores trabajando en este repositorio fue de 280 en 233 proyectos. El número de desarrolladores con más de dos proyectos fue de 47 (17% del total) en 157 proyectos (67% del total). De estos desarrolladores, aquellos que se encontraban colaborando con al menos otro desarrollador fueron 43.

4.2.1.6 Tigris⁶⁴

Apoyado por CollabNet desde 2001, este repositorio se enfoca en el desarrollo de proyectos colaborativos de fuente abierta. Fomentan y apoyan proyectos que se basen en ingeniería de *software* de código abierto, susceptibles de convertirse en herramientas amigables y accesibles para desarrollo de aplicaciones.

El movimiento de software de código abierto ha producido una gran cantidad de herramientas de desarrollo de software útiles y poderosas, pero también ha evolucionado el proceso de desarrollo de software que funciona mejor bajo condiciones donde el proceso de desarrollo normal falla.

Entre sus proyectos más populares se encuentran *Subversion*, *Subclipse*, *TortoiseSVN*, *RapidSVN*, *Scarab*, *ArgoUML*, *SubEtha*,

⁶⁴www.tigris.org/

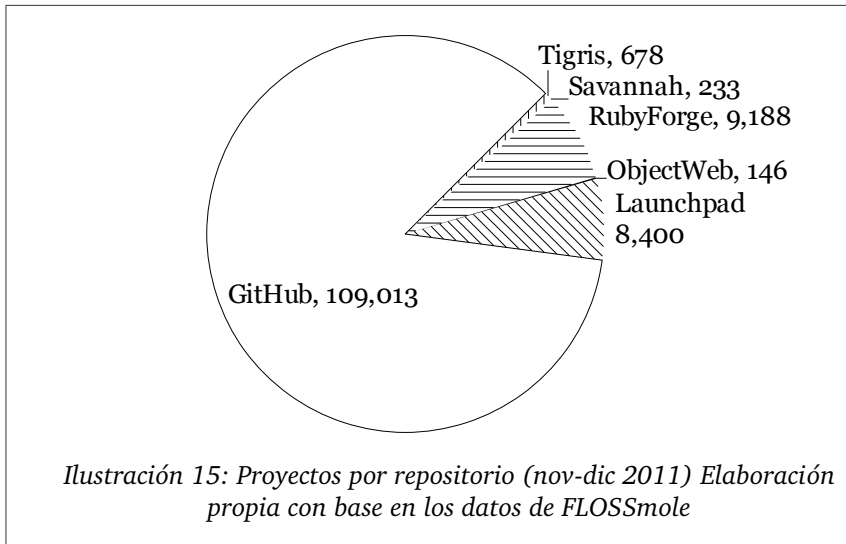
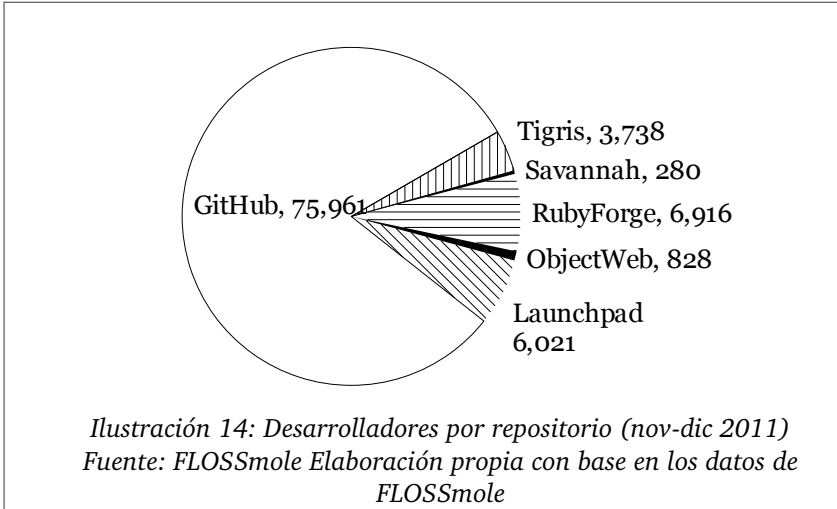
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

eyebrowse, midgard, cowiki, antelope, scon, build-interceptor, phing, maxq, aut, current, ReadySET, GEF, Axion, Style, SSTree.

En el período noviembre-diciembre analizado en esta tesis, el total de desarrolladores trabajando en este repositorio fue de 3,738 en 678 proyectos. El número de desarrolladores con más de dos proyectos fue de 74 (2% del total) en 185 proyectos (27% del total). De estos desarrolladores, aquellos que se encontraban colaborando con al menos otro desarrollador fueron 67.

Nombre	Clave	Dependencia	Fecha de fundación	Particularidad
GitHub	GH	<i>GitHub.Inc.</i>	2008	Aloja proyectos FLOSS, pero también privativos a través de suscripciones pagadas
Launchpad	LP	<i>Canonical Ltd.</i>	2004 privativo / 2009 libre	Aplicaciones libres, semi-libres y abiertas, principalmente desarrolladas por la comunidad Ubuntu.
ObjectWeb	OW	<i>OW2 Consortium</i>	2007	Para software abierto (OS) principalmente
RubyForge	RF	<i>Ruby Central, Inc.</i>	2003	Desarrollo de aplicaciones con el lenguaje Ruby
Savannah	SV	<i>Free Software Foundation</i>	2000	Principalmente para desarrollo, mantenimiento y distribución de GNU. Sólo software libre
Tigris	Tig	<i>CollabNet, Inc.</i>	2001?	Enfocada en desarrollo de proyectos colaborativos de software de fuente abierta.
<i>Características de los 7 repositorios estudiados. Elaboración propia</i>				

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



De los seis repositorios, Github tiene más desarrolladores y proyectos, lo que puede deberse a que su sistema de control de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

versiones, “git”, es cada vez más utilizado por los equipos de desarrollo. El sitio, además, aloja proyectos de, y es apoyado por, varias empresas reconocidas como *RedHat* y *Mozilla*, aunque también entre sus usuarios y patrocinadores están *Microsoft*, *Facebook*, *Twitter*, *VMware* y *Linkedin*. Estos patrocinios y apoyo, a pesar de que podrían generar dudas en las comunidades por la participación de empresas como *Microsoft*, otorgan seguridad y garantía de permanencia a mayor plazo por el soporte financiero que reciben.

Savannah se muestra como el más pequeño de los repositorios, con sólo 280 desarrolladores y 233 proyectos, pero la interacción entre desarrolladores y proyectos es mayor. Esto no es extraño si tomamos en cuenta que *Savannah* es el repositorio donde se desarrollan los proyectos de la Fundación de *software* Libre⁶⁵, en especial, el proyecto *GNU*, que es una de las tareas principales de esta organización.

Lo que muestra el comparativo siguiente es, para el momento del tiempo seleccionado, el número de desarrolladores totales trabajando en cada repositorio, el número de desarrolladores con más de 2 proyectos en cada repositorio y, finalmente, el número de desarrolladores con más de 2 proyectos, trabajando al menos con otro desarrollador en cada repositorio.

⁶⁵ La Fundación de *software* Libre está dirigida por Richard Stallman, cuenta con una organización definida y un fuerte grupo de donantes para sus diferentes proyectos; entre estos, y como prioridad, se encuentra el desarrollo y mantenimiento del sistema *GNU*.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Datos de desarrolladores y proyectos en 6 repositorios (nov-dic 2011)

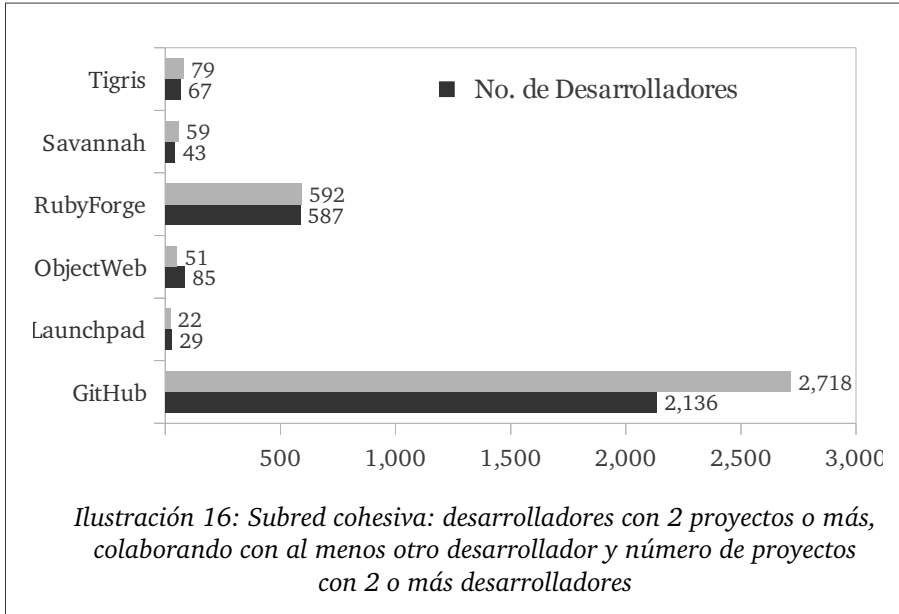
Nombre	Total de registros	Total actores	Total de proyectos	No. de actores	% con respecto al total	No. de Proyectos	% con respecto al total	No. de actores	% con respecto al total	No. de Proyectos	% con respecto al total
Fresh Meat	41,628	21,511	39,782	3,749	17	19,802	50	314	1	322	1
GitHub	133,074	75,961	109,013	11,651	15	50,102	46	2,136	3	2,718	2
Launchpad	26,072	6,021	8,400	337	6	2,229	27	29	.5	22	.3
ObjectWeb	1,228	828	146	391	47	94	64	85	10	51	35
RubyForge	12,389	6,916	9,188	1,052	15	4,426	48	587	8	592	6
Savannah	555	280	233	47	17	157	67	43	15	59	25
Tigris	4,224	3,738	678	74	2	185	27	67	2	79	12

**Los datos corresponden a noviembre-diciembre de 2011*

Elaboración propia con base en los datos de FLOSSmole

Los componentes más cohesionados (desarrolladores con más de 2 proyectos colaborando con al menos otro desarrollador) o subredes encontradas en los repositorios estudiados muestran una equivalencia relativa en el número de proyectos con el número de desarrolladores. La gráfica se presenta a continuación.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



Se puede decir con respecto a los gráficos mostrados que:

- El repositorio que, en proporción, tiene más desarrolladores con más de 2 proyectos es ObjectWeb con 391, que corresponden al 47% del total de desarrolladores en ese repositorio para ese momento del tiempo. Le siguen Savannah con 17% (47) de su total, GitHub y RubyForge con 15% (11,651 y 1052), Launchpad con 6% (337) y Tigris con 2% (74).
- De estos desarrolladores, los que actúan en colaboración con al menos otro desarrollador son 91% en Savannah y Tigris (43 y 67 desarrolladores, respectivamente), 56% en

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

RubyForge (587), 22% en ObjectWeb (85), 18% en GitHub (2,136), y 9% en Launchpad (29).

Lo anterior indica que no necesariamente un repositorio mayor tiene, en proporción, más interacciones entre usuarios; de hecho, parece indicarse completamente lo opuesto. Sin embargo, hay que recordar que aunque parezcan intercambios aislados, no necesariamente esto es así, puesto que, como se ha dicho, los proyectos son partes de componentes de un programa.

Es posible que en el momento del tiempo estudiado, los distintos repositorios se encontraban en diferentes momentos del proceso de producción: unos en el proceso de desarrollo de componentes individuales y otros en el proceso de unificación de componentes, o bien, una mezcla de ambos procesos, según el número de programas desarrollados o mantenidos.

La colaboración parece organizarse casi de modo espontáneo, pero no es así, pues todos los proyectos y los repositorios que los alojan tienen reglas de procedimientos y formas de trabajar. Más que pensar en formas emergentes de auto-organización, como lo expresa Weber (2004), “el orden emerge de manera orgánica o endógena de las propias interacciones locales entre individuos.” (p. 132)

Lo que sucede en estas redes de producción no es muy distinto de cualquier otro proceso de producción; es decir, se pueden ver la división del trabajo y los flujos de comunicación e intercambio entre los distintos grupos en diferentes momentos del tiempo.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Alguien decide comenzar el proyecto y comienza a escribir el código del proyecto... y cuando llega al punto de ser algo útil... lo muestra a los demás y otros comienzan a usarlo y... algún día alguien ofrece un cambio para mejorarlo, y el primero, que es automáticamente el jefe del proyecto, decide aceptar o no ese cambio. Cuando el proyecto crece más, cuando tiene más que un contribuidor regular, pueden decidir otro arreglo para su trabajo colaborativo [...] Pueden hacer lo que quieran, no hay reglas para eso. Cada grupo decide cómo actuar. (Stallman, 2011, entrevista realizada para este trabajo)

Lo interesante de este fenómeno es que la mayoría del trabajo es voluntario y responde a reglas que rebasan el puro interés lucrativo, ajustándose más, aunque no únicamente, a valores que tienen que ver con la solidaridad y el apoyo mutuo.

Los incentivos y motivaciones que pueden llevar a una persona a contribuir en el desarrollo del software libre son variados, y en general, al contrario de lo que se puede pensar, los de carácter económico no son los principales. (i Hernández, 2005: 93)

4.2.2 Las redes de colaboración y sus dinámicas de intercambio en cada repositorio

Como se ha mencionado, la organización en la producción de *software* libre y abierto se da en varios niveles. La forma en la que los datos están disponibles en los repositorios no permite aún, o no de manera sencilla, hacer una fusión entre repositorios a través de los programas y/o aplicaciones desarrollados en común. Tampoco es posible generar esta gran red a través de los desarrolladores que pueden estar trabajando al mismo tiempo en varios de estos sitios, sea en un mismo programa dividido o bifurcado, o bien, en distintas aplicaciones que pueden tener un fin de integración en un solo sistema.

Sin embargo, el análisis de la dinámica en cada una de estas plataformas permite describir las dinámicas de producción en diferentes momentos. Se puede decir que todos los programas desarrollados en todos los repositorios pasan por momentos en los que parece casi no haber interacción entre desarrolladores, y otros en los que existe un intercambio mayor. Esto es así porque, como se dijo, los proyectos son componentes separados de todo un programa, aplicación o sistema.

[...] pero en proyectos de fuente abierta, los desarrolladores principales trabajan en lo que son en efecto subtareas paralelas separables e interactúan entre sí muy poco. [...] mientras que la codificación sigue siendo un actividad solitaria esencialmente, las operaciones realmente grandiosas vienen de cosechar la atención y el poder mental de comunidades

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

enteras. (Raymond, 1999: 11 y 20)

La división del trabajo es necesaria para hacer efectiva y eficiente la producción; por lo tanto, la mayoría de los proyectos se trabajan por uno o dos desarrolladores, se revisan y complementan por otros, se arreglan cuando es necesario volviendo al trabajo aislado, se documentan por distintas personas del equipo; se integran a otros componentes, interactuando con los desarrolladores de esos proyectos, etcétera.

La modularidad es parte de la definición de fuente abierta porque la organización de los archivos fuente en pedazos dentro de un programa de software aísla tareas sobre la base de la funcionalidad. Un desarrollador voluntario se dedica a un arreglo específico que considera necesario dentro de los parámetros establecidos por el módulo. Selecciona un tema en el que está más calificado para trabajar. Ya que la modularidad lleva al programador a dedicarse en un solo componente del código, es la base para especificar, solicitar y organizar contribuciones de los voluntarios. (Booth, 2010: 19)

Aunque los datos de los repositorios son del mismo corte en el tiempo, noviembre-diciembre de 2011, la diversidad de proyectos y sus momentos en el proceso, permiten documentar visualmente la diversidad de redes más o menos cohesionadas que se pueden producir a lo largo del proceso de producción. En

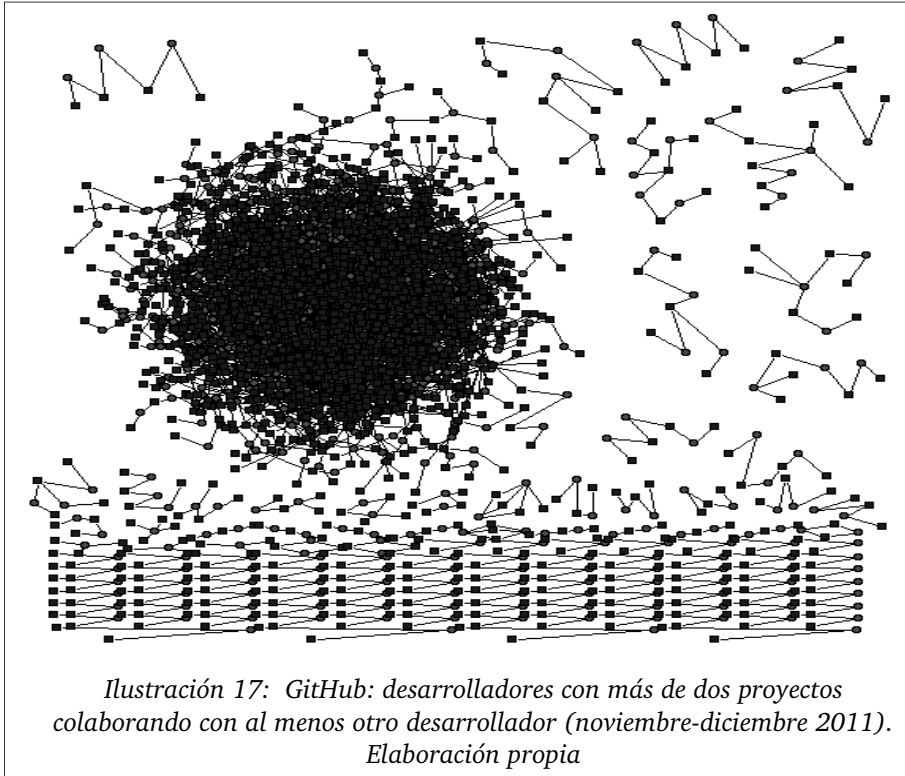
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

un estudio a mayor profundidad y con mayores recursos, será posible escoger un repositorio y dar seguimiento a la dinámica interna de sus programas y desarrolladores a lo largo del tiempo.

Por lo pronto, aquí se muestran distintas redes que representan la dinámica en ese momento del tiempo de cada uno de los repositorios seleccionados. Como se ha dicho, y es importante destacar de nuevo, no se pretende comparar las redes resultantes, puesto que nos parece que no son comparables sino complementarias para describir las dinámicas de estas comunidades de producción de FLOSS.

En las ilustraciones, los actores y proyectos son los nodos de las redes y están representados los primeros por círculos y los segundos por cuadrados. Las líneas corresponden al trabajo que une a un desarrollador con un proyecto. Estas redes son de dos modos; es decir, sus nodos son de dos tipos: desarrolladores y proyectos, por lo que se ven los vínculos de los actores hacia los proyectos, y no de los primeros entre sí. Se asume que si varios desarrolladores están vinculados a un proyecto, entonces, están interactuando entre sí.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



Esta primera red corresponde al repositorio GitHub. La cantidad de datos analizados es muy grande y no permite ver todos los nodos y flujos; sin embargo, es posible notar la existencia de varios pequeños componentes o equipos trabajando separadamente. Los equipos más pequeños se muestran en la parte baja del gráfico; otros, más grandes de 3-5 desarrolladores, colaborando en 4-5 proyectos se ven en la parte media y alta a la derecha. Hay una subred muy conectada, que corresponde al componente más grande, que se visualiza en el lado izquierdo de la ilustración. La imagen de esta red muestra

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

cómo los proyectos se vinculan a través de desarrolladores, o bien los desarrolladores interactúan por ser parte de proyectos compartidos. En los equipos medianos se aprecian desarrolladores que trabajan en varios proyectos; unos proyectos los vinculan con ciertos desarrolladores y otros proyectos con distintos actores de esta red.

Esta no es una red conexas, puesto que no hay vínculos entre todos los nodos (actores y proyectos) de la red. Sin embargo, podemos dividirla en dos grupos o clusters: el primero con 4,654 nodos que representa el 96% de actores y proyectos, que se encuentra vinculado en una gran red, y el segundo, más disperso, con 200 nodos entre desarrolladores y proyectos, que representa el 4%. El número de vínculos entre actores y proyectos es de 2.3634; es decir, que, en promedio, cada actor se vincula a 2.3 proyectos. Finalmente, la densidad⁶⁶ que proporcionan estos números es muy baja, ya que corresponde a solo un 0.0009880.

La red de GitHub muestra diferencias en los procesos de los diversos proyectos que desarrollaban para noviembre-diciembre de 2011. En general, aún en la subred más conectada al interior del repositorio, se muestran equipos trabajando de manera aislada, y los que están unidos, en realidad, lo están a partir de lo que se puede denominar como “cadenas de colaboración” (un desarrollador unido a un proyecto, por el que interactúa con otro desarrollador, el cual está unido al primero y a otros actores a través de proyectos distintos, y así de manera encadenada). En

⁶⁶ Son los vínculos existentes entre los vínculos posibles en toda la red. La mayor densidad posible es 1

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

el centro de la subred más cohesionada, se pueden encontrar algunas conexiones más densas, a las que podríamos denominar como “constelaciones encadenadas” (dos o más desarrolladores unidos a dos o más proyectos que los vinculan con dos o más diferentes actores, y así sucesivamente).

Resumiendo, en la red de GitHub se muestran momentos del proceso de producción de FLOSS que involucra de manera mayoritaria a equipos, trabajando en 'cadenas de colaboración', más que en 'constelaciones encadenadas'.

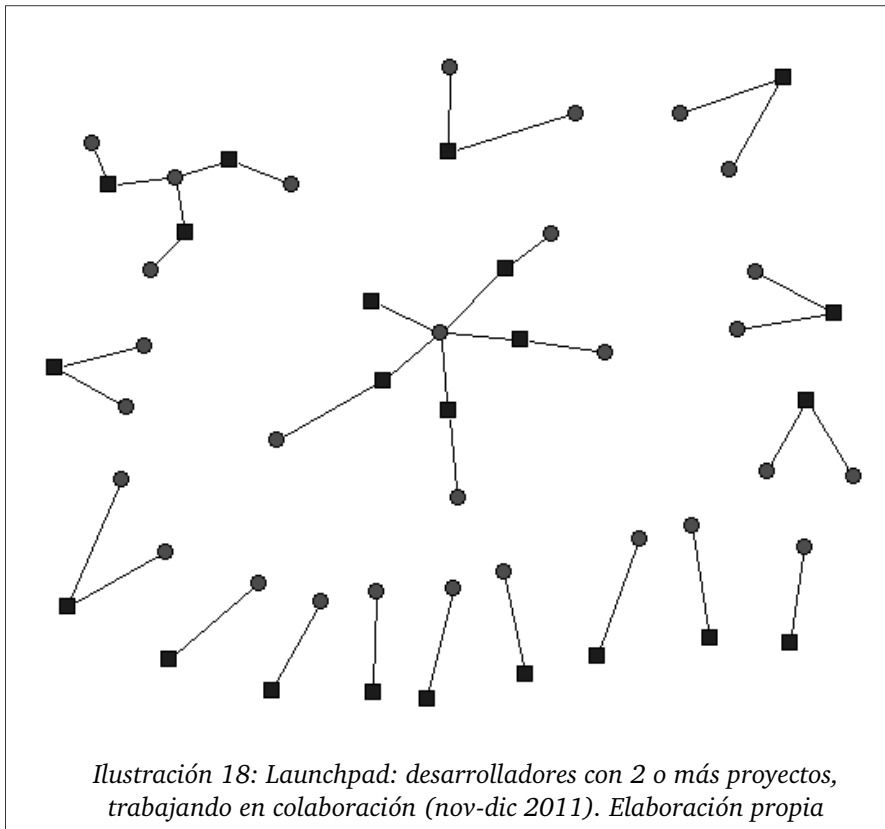
Se puede inferir que cuando se muestran cadenas de colaboración, se está trabajando en un momento del proceso en el que cada desarrollador trabaja en un componente del equipo y, si acaso, junto con otro desarrollador, aportan soluciones específicas. Esta sería la parte del proceso en el que se está desarrollando código. Por otro lado, cuando se ven 'constelaciones encadenadas', se infiere que se está en una parte del proceso de producción en el que varios desarrolladores revisan componentes en búsqueda de fallas o mejoras, por ejemplo, cuando se lanzan las versiones beta. Este momento del proceso lo ejemplifica Raymond (1999):

Linus objetó que la persona que entiende y arregla un problema no es necesariamente, incluso usualmente, la persona quien por primera vez lo caracteriza. “Alguien encuentra un problema”, dice, “y alguien más lo entiende. Iré más allá al decir que encontrarlo es el mayor reto.” [...] El punto clave es que ambas partes del proceso (encontrar y arreglar)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

tienden a suceder rápidamente. (p.8)

A continuación, se presenta la red de Launchpad que muestra otra forma de colaborar, según el momento del proceso en que se encuentran los proyectos en esta red. Las diferencias se presentan obvias.

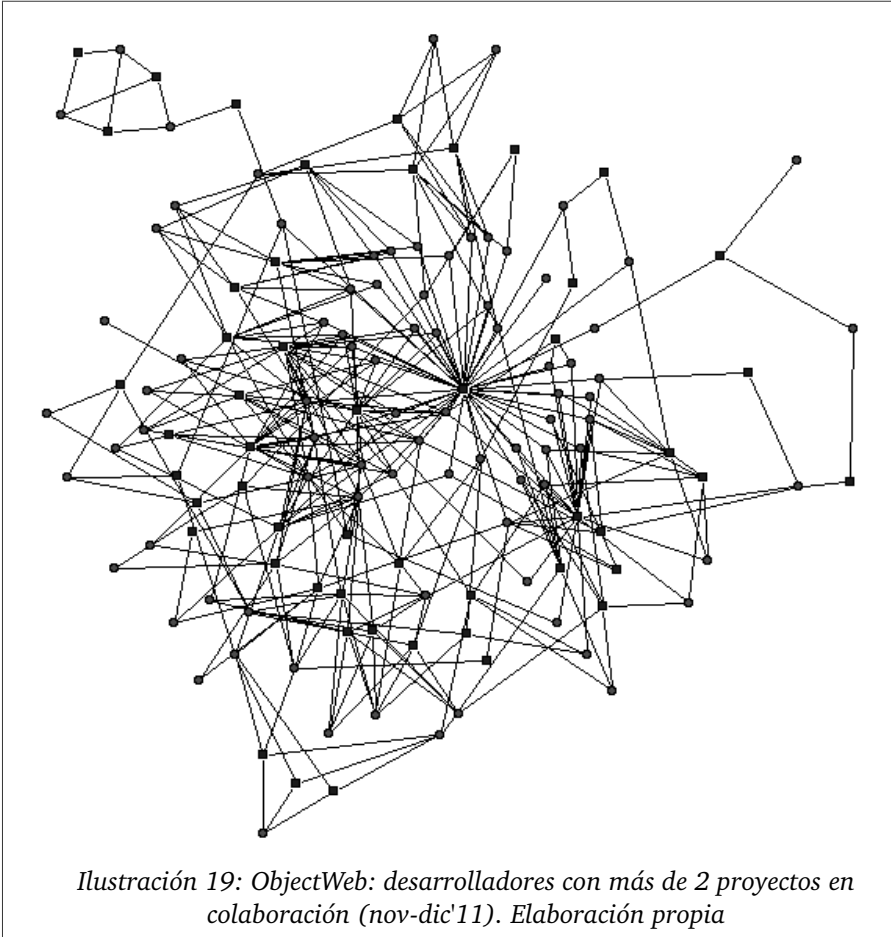


Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

En este caso, vemos los equipos aislados, incluso aún más separadamente que en la subred de GitHub. Dos pequeñas estrellas se muestran en el centro y arriba a la izquierda, que sugieren más que el momento de revisión entre pares, un momento del proceso en el que el líder o administrador de un equipo organiza al mismo en varias tareas. Lo que se muestra es un actor predominante que se une a otros desarrolladores a través de proyectos, pero estos programadores/usuarios no están unidos entre sí más que por el actor central. En este repositorio, la selección de proyectos y desarrolladores se encontraban en un momento del proceso de desarrollo de código y organización en la división del trabajo especializado. La dispersión de los nodos no permite sacar medidas sobre redes.

El repositorio ObjectWeb muestra diferencias con respecto a los anteriores. El gráfico se presenta a continuación. Esta red se encuentra en el momento más fuerte de la colaboración. Los proyectos involucrados están siendo trabajados por mínimo tres desarrolladores y cada uno de estos actores, en general, está vinculado a mínimo dos proyectos.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



Todos los nodos están conectados de alguna forma, por lo que es una red conexas. Su orden, es decir, la cantidad de nodos es de 136 (85 desarrolladores y 51 proyectos); los vínculos existentes son 348, lo que resulta en una densidad de $.0802768$ ⁶⁷.

⁶⁷ La densidad de la red se mide dividiendo los vínculos existentes entre los

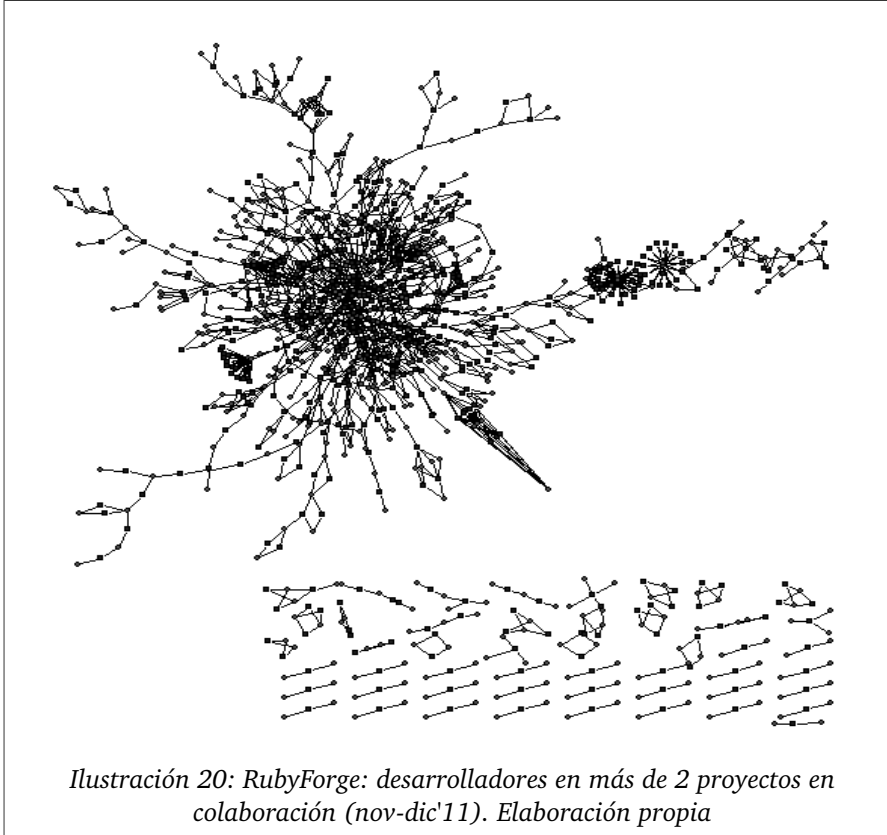
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Por lo que se muestra en esta plataforma es que los proyectos se encuentran en un momento del proceso de producción donde existen, tanto cadenas de colaboración, como constelaciones colaborativas. Hay líderes de proyecto que organizan las tareas y/o revisan las mismas, y hay desarrolladores que, al mismo tiempo que desarrollan código en colaboración con otros, revisan el trabajo de los demás. Cabe recordar que desde este repositorio, como los que veremos a continuación, el tipo de proyectos que se desarrollan son abiertos y/o libres, quedando fuera aquellos privativos que, por motivos de secrecía en el código fuente, permiten menor interacción entre usuarios del repositorio y equipos desarrolladores.

A continuación, se muestra el gráfico de la colaboración dentro del repositorio RubyForge. Éste tiene similitudes con GitHub, ya que hay una separación entre componentes: una subred conexas y varios equipos separados. Sin embargo, en este repositorio la sub-red no se conecta sólo con cadenas de colaboración lineales, sino con constelaciones colaborativas y estrellas.

vínculos posibles, así la densidad mayor es igual a 1, y es donde todos los nodos están conectados con los otros nodos de manera directa.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



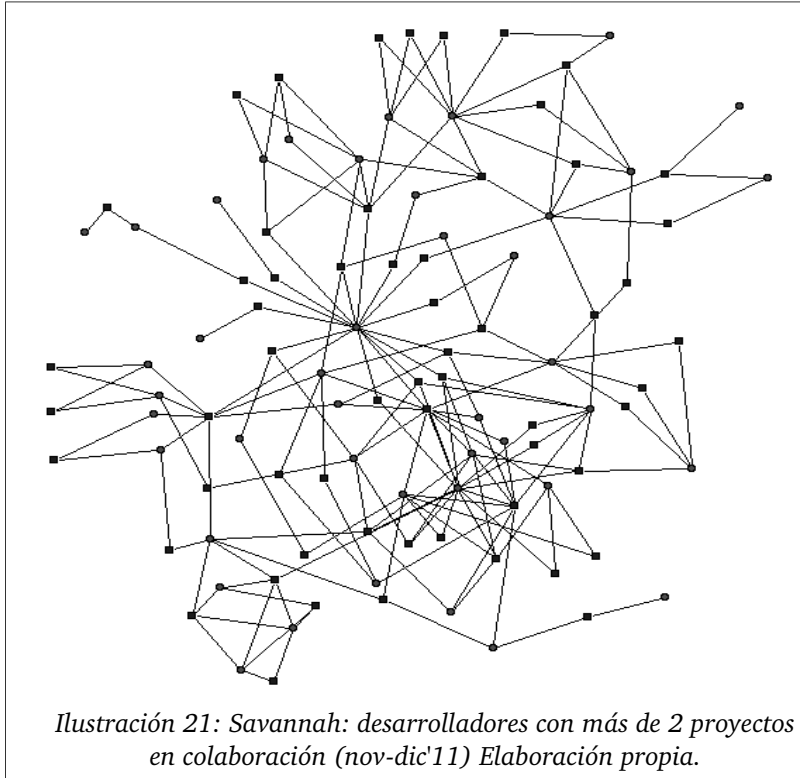
El gráfico de RubyForge muestra diversos momentos del proceso de producción. Visualmente, permite observar el trabajo aislado de desarrollo de código o documentación (triadas y cadenas de colaboración unidas o no a la subred); el trabajo colaborativo de organización y división del trabajo (estrellas), y el proceso de revisión y mejora (constelaciones colaborativas). La cooperación que se muestra y la interacción entre desarrolladores es lógica si recordamos que éste es un

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

repositorio cuya característica base es desarrollo de código para aplicaciones con el mismo lenguaje de programación; es decir, hay un elemento (el lenguaje de programación Ruby) común entre los proyectos y una habilidad común (conocer y manejar este lenguaje) entre los desarrolladores.

La red que se ilustra ahora corresponde a la interacción percibida en el repositorio Savannah. La conexidad de esta subred es similar a la de ObjectWeb, ya que los vínculos son mayoritariamente cadenas de colaboración, aunque también existen constelaciones que unen grupos de desarrollo entre sí. Nuevamente, se encuentra una lógica en las dinámicas de interacción de esta red, si se toma en cuenta que hay un elemento común: Savannah es el repositorio de desarrollo del sistema y aplicaciones para *GNU*.

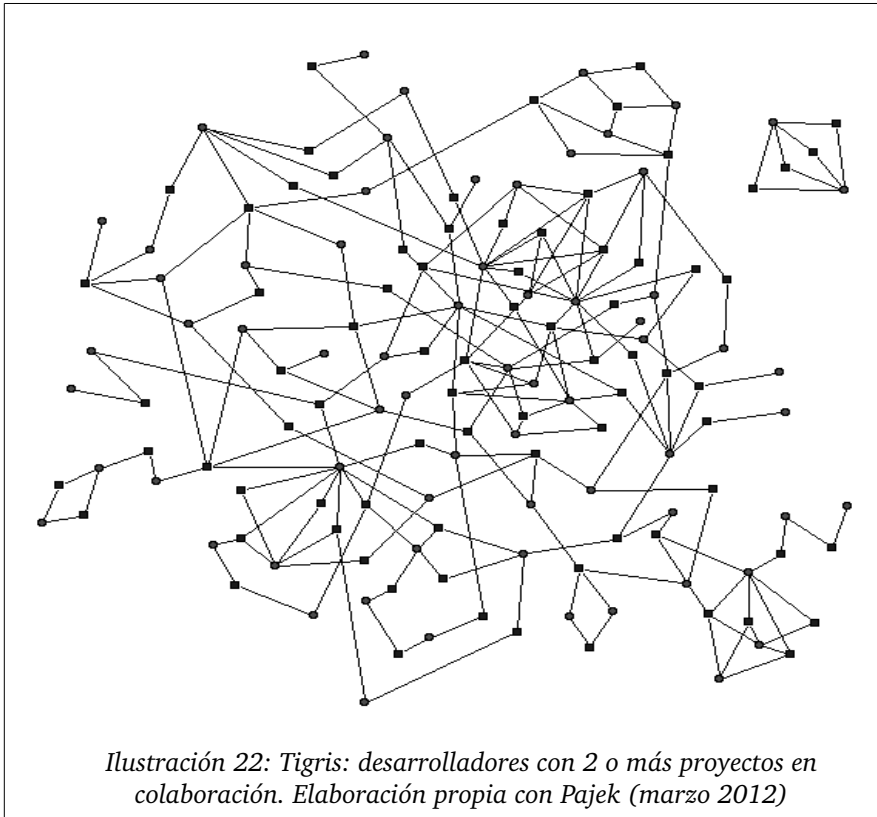
Software Libre y abierto: comunidades y redes de producción digital de bienes comunes



Finalmente, el repositorio Tigris muestra la interacción entre desarrolladores y usuarios de manera similar a Savannah y Object Web. La cantidad de nodos de esta subred es de 146 (67 desarrolladores y 79 proyectos) y el número de vínculos es de 215. Su densidad es de 0.0406197. El grado de centralización promedio es de 0.02466, lo que indica que hay un equilibrio relativo en la cantidad de vínculos de cada nodo, aunque hay actores más conectados a otros. La formación de redes en este repositorio resulta lógica ya que este repositorio tiene como

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

característica principal alojar proyectos que se trabajen de modo colaborativo; es decir, fomenta la interacción constante entre los actores.



Como puede verse, la morfología de las redes de producción de *software* libre es muy diversa y no está determinada por el tamaño de la colaboración, si acaso podría decirse que hay más similitudes en las dinámicas internas entre aquellos repositorios que sólo alojan proyectos libres y/o abiertos. Sin

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

embargo, como se mencionó, deben verse las distintas dinámicas como diferentes momentos en el proceso de producción, más que como características de una forma de organización en los repositorios. Lo más probable es que la interacción más fuerte y constante se dé en los equipos de desarrollo de FLOSS, lo que es lógico porque son los que otorgan acceso al código fuente, permitiendo con esto mayor libertad en la colaboración, tanto para desarrollar y mejorar el código, como para crear bifurcaciones y nuevos programas a partir de una sola propuesta.

La visualización de estas redes ha permitido la descripción de ciertas interacciones percibidas entre desarrolladores y usuarios de FLOSS. Lo que denota el análisis anterior es que la dinámica de la cooperación en estas comunidades se presenta como un sistema complejo, no es lineal ni determinista. El que exista mayor o menor cooperación en cualquier momento del proceso no implica éxito o fracaso en el producto final.

La producción de FLOSS se presenta aquí como una forma de organización flexible, en la que el siguiente paso no está predeterminado, sino que se define conforme a las situaciones emergentes y condiciones del producto en cada momento. El trabajo en colaboración dentro de estas redes, en general, es voluntario y responde más al ejercicio de la creatividad y, si acaso, a la búsqueda de reconocimiento en las comunidades *hackers* que a valores impuestos por el mercado. La razón por la que estas redes y comunidades trabajan con ese nivel de incertidumbre, tanto en el tipo de colaboración con la que cuentan y las formas de reconocimiento que reciben es la pregunta que se intentará responder en el siguiente apartado, a

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

través del análisis del discurso de actores clave en este fenómeno.

4.3 Las comunidades y redes de producción digital de *software* libre y abierto desde la perspectiva de sus actores

4.3.1 Motivaciones de los actores⁶⁸

Como se ha mencionado, las razones para la colaboración no persiguen de manera principal la remuneración económica, aunque tampoco está excluida. Estudios y argumentos de los propios actores dentro de las comunidades y redes de producción de *software* libre y abierto muestran que, entre las principales motivaciones de los desarrolladores para contribuir, se encuentran algunas de índole ideológica y otras de carácter técnico. Entre las primeras, se encuentran nociones e ideas sobre la libertad, el acceso libre al conocimiento, la solidaridad y la justicia social. Por otro lado, también son importantes la búsqueda de reconocimiento y pertenencia a una comunidad, la diversión y sensación de reto al resolver un problema específico, la posibilidad de aprender-colaborando y de obtener algún beneficio, como puede ser una oportunidad laboral o un pago monetario. Un estudio de Lakhani y Wolf (2005), arrojó los siguientes resultados sobre las motivaciones de los desarrolladores para la colaboración en la producción de FLOSS:

⁶⁸ Como se mencionó anteriormente, este análisis se basó en los datos arrojados por una entrevista realizada, dos entrevistas recabadas en línea y artículos y textos seleccionados de personajes reconocidos en las comunidades de FLOSS.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Motivación	% de respuestas
El desarrollo de código es intelectualmente estimulante	44.9
Me gusta trabajar con este equipo de desarrolladores	20.3
Para mejorar mis habilidades de programación	41.3
Necesito el código para algo personal	58.7
Me da status profesional	17.5
Creo en que el código fuente debe estar abierto	33.1
Siento una obligación personal en contribuir al uso de FLOSS	28.6
No me gusta el software privativo y quiero vencerlo	11.3
Me da reputación en la comunidad FLOSS	11

Respuestas obtenidas en estudio de Lakhani

Fuente: (K. R. Lakhani & Wolf, 2005)

A continuación, he seleccionado y presento algunas de las motivaciones enunciadas por los actores y recogidas en artículos, conferencias y entrevistas. En este trabajo, he clasificado las frases en cuatro categorías, que corresponden a cuatro a cuatro conceptos de un esquema cognitivo (Ver Strauss & Quinn, 1997), a saber, creatividad, identidad social, identidad política y reproducción.

- a) Motivaciones de carácter técnico, donde se engloba el aprendizaje y mejora en las habilidades de programación, la diversión por el desarrollo de código y/o por la resolución de problemas. En el discurso expuesto sobre este punto, se puede definir un núcleo conceptual que tiene que ver con la creatividad, la sensación que emerge de identificar “formas de hacer”

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Linus Tovarlds explica que, como muchos otros hackers, empezó a trabajar en el desarrollo de Linux simplemente por la diversión que implicaba hacer algo interesante [...] Enfrentarse a un reto y encontrar una buena solución produce un estado de satisfacción similar al que experimenta un científico cuando realiza un descubrimiento. (i Hernández, 2005: 136)

El lado del código abierto es extremadamente importante para la creatividad, para lo académico, para las ciencias de la computación y los sitios-web. Como disciplina, necesitas plataformas de fuente abierta en las que la gente pueda probar y desarrollar nuevas ideas. (“Entrevista a Tim Berners-Lee” en Powell, 2006)

[Crear] arte y belleza [...] Los desarrolladores hablan de escribir código no sólo como un problema de ingeniería, sino también como un objetivo estético, una cuestión de estilo y elegancia que hace de la codificación un acto de auto-expresión. (Resultado de encuesta de 2001 por Boston Consulting Group, citada en Weber, 2004: 135)

Crear código elegante y limpio es parte de la experiencia del programador que ve su trabajo como vocación [...] Los programadores consistentemente describen [la solución de problemas y el compartir esta solución] como un sentimiento primordialmente de eficacia más que deberse a motivaciones sociales per se. (Resultado de encuesta de 2001 por Boston Consulting Group, citada en Weber, 2004: 135, 137)

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Lo que me parece realmente importante del software libre es que te permite desarrollar tecnología en tus propios términos. (“Entrevista a Mark Shuttleworth,” n.d.)

Si un desarrollador de software tiene una necesidad que no cubre ningún programa es probable que escriba el programa que necesita o, si encuentra un problema en una aplicación de terceros que quiera resolver, que colabore en el proyecto solucionando el problema específico. (i Hernández, 2005: 136)

Todo buen trabajo de software inicia intentando resolver un problema personal del propio desarrollador. (Raymond, 1999: 4) Para resolver un problema interesante, inicia por encontrar un problema que es interesante para ti. (ídem: 20)

La innovación requerida de los usuarios vanguardistas: Los usuarios vanguardistas encaran necesidades que se generalizarán en el mercado, pero las encaran antes que los demás; ellos están posicionados para beneficiar de modos significativos si pueden encontrar respuestas a estas necesidades. (Von Hippel, 1988: 107)

El mundo del software libre es un excelente campo de aprendizaje para profesionales de las tecnologías de la información. Ofrece la posibilidad de estudiar el código fuente de miles de programas informáticos, participar en su desarrollo y corrección de errores, o simplemente mejorar nuestros conocimientos generales siguiendo la evolución de

las diferentes comunidades. (i Hernández, 2005: 137)

Un hacker puede intentar entrar furtivamente en un determinado servidor, pero sólo para estudiar las medidas de seguridad del sistema como medio de aprendizaje, nunca va a perder su valioso tiempo en dañar ningún ordenador. (Gómez Sánchez, 2008: 134)

[Contribuir en OS] es la forma en la que uno se convierte en un mejor programador. [...] Para aprender cómo un programa en particular trabaja (Respuestas sobre motivaciones realizadas por Shapiro y David, citado en Booth, 2010: 21)

[La gente coopera] porque es natural, porque desean tener un buen resultado y saben que colaborar es la manera de producirlo [...] A veces uno es capaz de escribir todo el programa y puede hacerlo, pero... por fin reconocerá que el programa podría ser mucho mejor con la participación de otros... porque él tiene tiempo limitado. No puede trabajar más de 13 horas al día, entonces, si otros trabajan 10 horas más cada día, añadiendo cosas útiles, en horario el programa sería mucho mejor con la ayuda de los otros. (Stallman, 2011: entrevista realizada para este trabajo)

- b) Motivaciones de carácter individual-colectivo, entre las que emana la búsqueda de reconocimiento y prestigio dentro de la comunidad, así como la necesidad de sentir pertenencia a la misma. El núcleo conceptual al que referimos este tipo de motivaciones es la identidad social,

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

la forma en la que los actores buscan mirarse a sí mismos y que se les mire, la necesidad humana de pertenencia grupal:

En el software libre, tal como describe Eric S. Raymond, la economía del regalo en este entorno tiene como objetivo el ganar reputación entre los otros miembros de la comunidad. (i Hernández, 2005: 136) Esto se puede traducir en oportunidades laborales. (idem: 139)

Los buenos programadores saben qué escribir, los mejores saben qué re-escribir (y reusar). (Raymond, 1999: 4) La “función de utilidad” que los desarrolladores de Linux maximizan no es económica, sino lo intangible de sí mismos, la satisfacción de su ego y la reputación entre otros hackers. (ídem: 22)

[...] El sentido de quién es quién [en la comunidad Mozilla] está basado en su reputación informal dentro de la comunidad, su historial de contribuciones, ese tipo de cosas. (Brian Behlendorf, miembro fundador de la Fundación del software Apache y miembro del Consejo de la Fundación Mozilla, citado en Booth, 2010: 30)

Es mejor ser descrito como hacer por otros que describirse así a uno mismo. Los hackers se consideran a sí mismos como una élite (una meritocracia basada en la habilidad), una en la que los nuevos miembros son bienvenidos. Existe, sin embargo, una cierta satisfacción del ego al ser identificado como hacker (pero si tu dices ser uno y no lo

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

eres, rápidamente serás etiquetado como 'bogus'.(E. S. Raymond, 1992: 123)

Satisfacción del ego, Reputación (Resultado de encuesta de 2001 por Boston Consulting Group, citada en Weber, 2004: 136)

La gente debe sentir que su conexión a un proyecto y su influencia sobre el mismo es directamente proporcional a sus contribuciones. (Fogel, 2005: 17)

El deseo de devolver algo a la comunidad [...] Proveer alternativas al uso de software privativo (Respuestas sobre motivaciones realizadas por Shapiro y David, citado en Booth, 2010: 21)

Interactuar con programadores que piensan de manera similar a uno. (Respuestas sobre motivaciones realizadas por Shapiro y David, citado en Booth, 2010: 21)

Muchos desarrolladores creen más fuerte que nunca que sus valores alrededor de la cooperación y el intercambio de conocimientos en la producción son las razones fundamentales de porqué hacen lo que hacen. (Weber, 2004: 116)

Torvalds dijo que la decisión de liberar Linux “no fue una decisión agonizante que tomé, pensando larga y profundamente; fue una decisión natural dentro de la comunidad de la que sentí que yo quería ser parte.”

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

(Entrevista con Linus Torvalds, First Monday 3(3), March 1998, en www.firstmonday.dk/issues/issue3_3/torvalds/index.html, citado en Weber, 2004: 140)

Y sobre todo una sociedad necesita incentivar el espíritu de cooperación entre sus ciudadanos. Cuando los propietarios de software nos dicen que ayudar a nuestro vecino de una manera natural es «piratería», están contaminando el espíritu cívico de nuestra sociedad. (Stallman, 2004: 68)

Estamos en desacuerdo [el movimiento de código abierto y el de software libre] en los principios básicos, pero estamos mas o menos de acuerdo en las recomendaciones prácticas. Así que trabajamos juntos en muchos proyectos específicos. No pensamos en el movimiento open source como enemigo. El enemigo es el software propietario. (Stallman, 2004: 76)

Pero no es obligatorio distribuir el programa, puedes escribir un programa y usarlo. En unos casos diría que si el programa es muy importante, diría que tienes un deber social de distribuir el programa a los demás, pero si no es tan importante, no diría nada. Entonces, este programa libre, pero privado no es un bien común, pero, de ser publicado, de ser disponible al público, el programa libre es un bien común. (Stallman, 2011: entrevista realizada para este trabajo)

- c) Motivaciones de carácter ideológico, donde emerge la identidad política de los actores y los adhiere o separa de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

grupos al interior de la propia red de desarrolladores. En esta categoría aparecen las nociones y argumentos sobre la libertad, la propiedad, los valores sociales, los bienes comunes, etcétera.

El movimiento de software libre no era un movimiento tecnológico; era la cara de una lucha por la libertad de pensamiento en el ámbito tecnológico. Tomó ventaja de la realidad tecnológica para poner sobre la mesa un escrutinio más profundo de sus posibilidades políticas. (Moglen, 2004: 4)

Cuando Richard Stallman creó la Free software Foundation en 1984 e inició el proyecto GNU su mayor motivación era la creencia de que el software libre era necesario para preservar la libertades de los usuarios y era bueno para la sociedad. (i Hernández, 2005: 136)

El internet es un recurso global público que debe mantenerse abierto y accesible [...] Los individuos deben tener la habilidad de dar forma a sus propias experiencias en Internet [...] FLOSS promueve el desarrollo de Internet como un recurso público. (El Manifiesto Mozilla, citado en Booth, 2010: 23)

La ética hacker cree en que el intercambio de información es un bien positivo poderoso y que es un deber ético compartir su conocimiento al escribir software libre y facilitar el acceso a la información y a los recursos de cómputo siempre que sea posible [...] Muchos hackers se

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

suscriben a esta ética por este motivo y mucho actúan escribiendo y regalando software libre. (E. S. Raymond, 1992: 123)

[Derrotar a] el enemigo común, Identidad y sistemas de creencias (resultado de encuesta de 2001 por Boston Consulting Group, citada en Weber, 2004: 136)

[...] es lo mejor del software libre, porque el software privativo anima todo lo malo de la naturaleza humana, mientras el SL anima lo bueno... Lo bueno y lo malo existen en todos, pero un sistema social puede animar uno u otro, y así podemos tener más o menos de lo bueno y de lo malo. (Stallman, 2011: entrevista realizada para este trabajo)

La importancia del software libre y abierto está más allá de los negocios o el código eficiente [...] por código abierto quiero referirme a los valores que cruzan tanto los contextos técnicos como legales que promueven un mundo donde las estructuras –códigos- de gobierno son fundamentalmente libres. (Lessig, 2001: 270)

La sociedad también necesita libertad. Cuando un programa tiene un propietario, los usuarios pierden la libertad de controlar una parte de sus propias vidas. (Stallman, 2004: 68)

Tienes derecho a poder cooperar abierta y libremente con otras personas que usan software. Tienes derecho a poder aprender cómo funciona el software, y a enseñar a tus

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

estudiantes con él. Tienes derecho a poder contratar a tu programador favorito para arreglar- lo cuando se rompa. (Stallman, 2004: 69)

Unos pocos van más lejos y argumentan que toda la información debe ser libre y que todo control privativo de ésta es malo. Esta es la filosofía detrás del proyecto GNU. (E. S. Raymond, 1992: 123)

Los usuarios deben ser libres de modificar el software que utilizan. (80% de las respuestas sobre motivaciones realizadas por Shapiro y David, citado en Booth, 2010: 21)

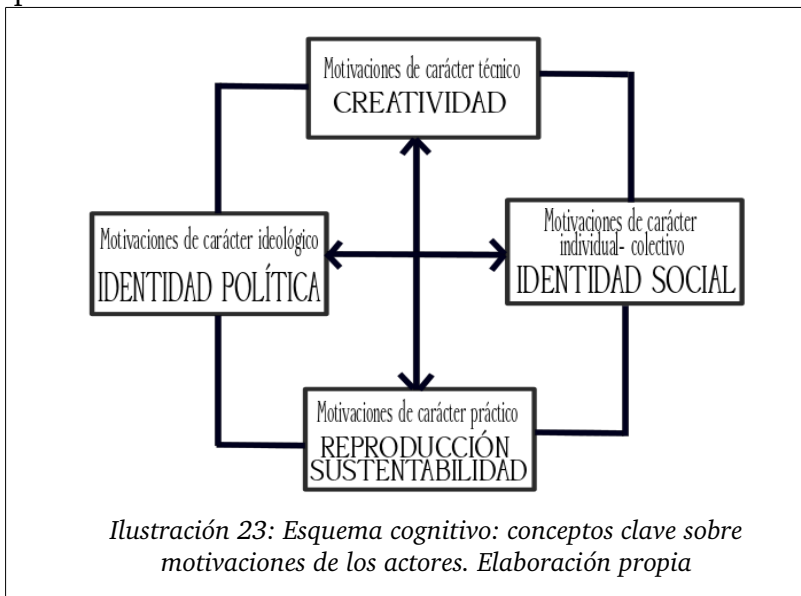
- d) Motivaciones de carácter práctico, que tienen que ver con la reproducción de la vida cotidiana, los aspectos materiales que retribuyen el trabajo realizado, las posibilidades financieras para la sustentabilidad, etcétera. El concepto clave al que se unen estas motivaciones es al de reproducción/sustentabilidad.

Son habituales los casos donde una empresa – como pueda ser Red Hat o Novell – tiene desarrolladores colaborando en un proyecto libre o una universidad que ha conseguido apoyo financiero para desarrollar o mejorar un proyecto libre [...]En el caso de las empresas es común que estos desarrolladores sean contratados de las propias comunidades, es decir, voluntarios que ya estaban contribuyendo al proyecto y que ahora se podrán dedicar a tiempo completo como parte de su trabajo.(i Hernández,

2005: 138)

Durante años, hasta que una beca lo hizo innecesario, yo me ganaba la vida realizando mejoras a medida sobre software libre que yo había escrito. Cada mejora se añadía a la versión estándar publicada y así, eventualmente, se ponían a disposición del público en general. Los clientes me pagaban para que trabajase en las mejoras que ellos querían, en lugar de en las características que yo habría considerado como la máxima prioridad. (Stallman, 2004: 68)

El esquema que resulta de la clasificación realizada en esta tesis muestra las conexiones entre estos cuatro conceptos clave, lo que permite aproximarse a una explicación preliminar de los comportamientos de estos actores.



Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Cada concepto alimenta a otro y obtiene también retroalimentación. La creatividad de los desarrolladores, aplicada en las contribuciones que realizan, responde a la búsqueda de reconocimiento dentro de comunidades *hackers* específicas (según la ideología política de cada actor y sus motivaciones de carácter práctica; es decir, la forma en la que busca sustentar su propio trabajo). El trabajo desarrollado, entonces, puede resultar en la obtención de prestigio, dependiendo de sus habilidades y presencia en los espacios de cooperación colectiva.

Según las observaciones realizadas para este análisis, se puede decir que las diferentes motivaciones de los actores no aparecen de manera aislada; es decir, la existencia de nociones dentro de una categoría conceptual del esquema propuesto no excluye la existencia de argumentos que sustenten otras categorías. Más concretamente, sostengo que todos los actores manejan de manera más o menos conciente el mismo esquema cognitivo, aunque sólo se enuncien algunas ideas en el discurso que pertenecen a una o dos categorías. Propongo como hipótesis de trabajo que la preminencia de una categoría sobre otra puede explicar las diferencias de comportamiento en los distintos grupos de desarrolladores que conviven en internet.

4.3.2 Respuestas de las comunidades y redes de FLOSS al mercado digital, en tanto defensa de la producción colectiva y de alianzas estratégicas

La forma en que las comunidades de desarrollo de *software* libre y abierto responden al mercado es distinta. Sin embargo,

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

existe una lucha en común para hacer que el desarrollo abierto y/o libre de código sea sustentable. El fenómeno FLOSS ha demostrado que puede sostenerse en el tiempo y, no sólo eso, que puede proporcionar importantes beneficios económicos, convirtiéndose en una forma alternativa de insertarse en el mercado.

La capacidad de las aplicaciones libres y abiertas para ganar el mercado de *software* en el mundo fue descrita para la empresa Microsoft en un documento conocido como “el documento Halloween”. En éste, se hablaba del fenómeno de código abierto como una amenaza para la empresa:

[...] la calidad comercial puede ser lograda, incluso excedida por los proyectos de software de fuente abierta (OSS) [...] Linux y otros partidarios de OSS están utilizando de manera progresiva un argumento más creíble de que el software abierto es al menos tan robusto –si no es que más– que las alternativas comerciales. (citado en Weber, 2004: 126)

Desde finales de los 90, sobre todo en Estados Unidos, el fenómeno FLOSS emergió como una promesa financiera. *Red Hat* y *VA Linux* en 1999 se incorporaron al índice *Nasdaq* de la bolsa de valores.⁶⁹

⁶⁹ La síntesis histórica de lo que ha sucedido desde entonces puede verse en: “Revolution OS” (<http://video.google.com/videoplay?docid=7707585592627775409>) y en “The Code Linux” (<http://video.google.com/videoplay?docid=->

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Aunque, las cosas han cambiado mucho desde entonces, especialmente para VA Linux, Red Hat hoy en día tiene más capitalización bursátil que algunas de las empresas tradicionales de software propietario. (i Hernández, 2005)

Otras empresas emergieron, vendiendo OSS como producto a la medida, o bien cobrando por la distribución y los servicios. Muchas de ellas, la gran mayoría, como *Cygnus* o *Canonical*, ideológicamente son más afines a la Iniciativa *Open Source* (fuente abierta) que a la Fundación de *software Libre*, pues aunque apoyan y financian el desarrollo de programas libres, también participan, y permiten en sus aplicaciones y/o distribuciones, el uso de *software* privativo.

El *software libre* no quiere decir que sea gratuito; por lo que ofrece oportunidades de negocios en varios sentidos, ya sea por la redistribución física de aplicaciones y programas, o por el servicio de instalación, consultoría, apoyo y seguimiento, como lo han hecho ya algunas empresas. Otra forma de sustentabilidad es el financiamiento a través de donaciones, como lo recibe la *Fundación de software Libre*.

Algunas compañías, incluyendo Intel, Motorola, Texas Instruments y Analog Devices, han unido esfuerzos para financiar el desarrollo continuado del compilador GNU para el lenguaje C. Mientras, el compilador GNU para el lenguaje Ada está siendo financiado por la Fuerza Aérea de EEUU, que cree

[3498228245415745977](#))

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

que ésta es la manera más efectiva de conseguir un compilador de alta calidad. (La financiación de la Fuerza Aérea se acabó hace algún tiempo; el Compilador GNU de Ada está ahora en servicio, y su mantenimiento se financia comercialmente). (Stallman, 2004: 69)

En general, todos los programas en desarrollo y mantenimiento reciben donaciones de distintas fundaciones, empresas y personas. Aunque ésta no es una práctica común en México, sí lo es en otras partes del mundo, como Estados Unidos, Inglaterra o Francia. Las donaciones muchas veces son deducibles de impuestos y, además, otorgan beneficios especiales al donador.

Para defender la producción de FLOSS, muchos grupos lo que hacen es difundir su uso, buscando expandir la base de usuarios. Para Stallman, lo importante es difundir la filosofía sobre la libertad y no sólo el uso de programas abiertos:

[...] hay muchos en la comunidad, la mayoría, que piensan buscar a más usuarios... a toda costa buscar más usuarios; es decir, buscar éxito y sí sirve algo. Es útil tener a más usuarios, pero sólo tener a más usuarios no lograría la meta última de libertad, porque tener a muchos usuarios que no conocen la filosofía es diluir la filosofía, es perder las ideas más importantes. Para que un movimiento logre un trabajo de largo plazo, el requisito principal es no olvidar la meta. Y el problema entonces de lo que

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

hacen ellos es que nos hace olvidar la meta, por lo tanto mi esfuerzo es para difundir la filosofía, con pensar la debilidad en lo que ellos hacen.

Para lograr este tipo de defensa, la *Fundación de software Libre* (FSF) realiza a lo largo del año varias campañas para difundir los principios de libertad de uso, distribución, modificación y mejora, entre otras como:

- Defectuoso desde el diseño (Defective-by-Design): es una campaña en contra de las restricciones administrativas digitales, esto es, elementos integrados a las aplicaciones que restringen las acciones que puede ejecutar el usuario, p.e. que éste no pueda hacer cierto tipo de descargas o copiar texto, etcétera.
- Sistema Operativo *GNU*, que apoya el desarrollo y mantenimiento del proyecto *GNU*, a través del repositorio (como se vio) Savannah
- Los siete pecados de Windows, que busca difundir las prácticas ocultas detrás del sistema operativo Windows 7.
- Fin a las patentes de *software* difunde el peligro de la utilización de patentes de *software* en el mundo
- Campaña de Documentos Libres, que busca extender las nociones sobre la libertad hacia los manuales y documentación de las aplicaciones y programas

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Contra ACTA (Anti-Countering-Trade-Agreement)

Una parte importante de la defensa de la producción colectiva tiene que ver con la protección de los programas a través de licencias. Éste es uno de los logros más importantes y de mayor impacto de Richard Stallman, pues le dio la vuelta al derecho de autor con su concepto *Copy-left*. Como lo comentó en entrevista:

[...] el otro caso posible es que alguien haga una versión privativa del programa libre de otro. En este caso, la primera pregunta es si la licencia libre lo permite o no, porque las licencias libres con izquierda de autor lo prohíben, pero las licencias libres sin izquierda de autor lo autorizan... Si este proyecto lleva una licencia que autoriza hacerlo privativo, no hay qué hacer, excepto buscar mejorar la versión libre, para que la versión privativa no tenga mucho éxito... Pero si la versión privativa es ilegal, los tenedores del derecho de autor de este programa libre pueden demandar. (Stallman, 2011 en entrevista para este trabajo).

Sin embargo, no existen aún instancias dentro de las redes de *software* libre que se dediquen a hacer seguimiento de los programas que se licencian o patentan. La única manera de detectar la inclusión de código libre en un programa privativo es a partir de los propios usuarios y desarrolladores; es decir, a través de que alguien se dé cuenta al utilizar un programa privativo que esto está sucediendo, y lo denuncie.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Antes de terminar este apartado, deseo retomar algunos de los puntos más sobresalientes de este análisis. La dinámica de las redes de producción de *software* libre y abierto se ha estudiado en esta tesis bajo la perspectiva del interaccionismo estructural y el análisis de redes sociales, así como de la Teoría del Esquema y el Conexionismo. Con estas herramientas teórico-metodológicas, he podido hacer una exploración de las motivaciones de los actores para contribuir en este tipo de redes y una descripción de la colaboración en un momento del tiempo dentro de 6 repositorios con distintos proyectos de desarrollo en proceso. Así pues, de este análisis se puede concluir que las redes de producción de FLOSS:

- presentan una organización flexible en la que el siguiente paso no está predeterminado, sino que se define conforme a las situaciones emergentes y condiciones del producto en cada momento.
- su morfología es muy diversa y no está determinada por el tamaño de la colaboración, si acaso podría decirse que hay más similitudes en las dinámicas internas entre aquellos repositorios que sólo alojan proyectos libres y/o abiertos.
- Por otro lado, el esquema motivacional de los actores se compone de elementos de carácter técnico, práctico e ideológico en mayor o menor grado: la forma en la que se mezclan estos elementos inclina a mayor o menor colaboración, compromiso y/o espera de retribuciones de algún tipo.

4.4 El futuro del *software* libre y abierto

Muchos son los retos que enfrenta hoy el *software* libre y abierto, siendo uno de los principales la sustentabilidad de los proyectos a largo plazo. Muchos programas han logrado superar la prueba de permanencia a través del uso extendido por la calidad misma del *software*, como es el servidor Apache, el buscador Google, el sistema operativo *GNU*. La difusión en las preferencias de los usuarios sobre este tipo de programas y sistemas ha logrado mayor financiamiento y, con ello, mayor poder de permanencia, a través del mantenimiento constante. Sin embargo, no es un panorama generalizado aún.

Otro reto planteado tiene que ver con los usuarios, en tanto su papel en el proceso de producción. La expansión en el uso del *software* libre y abierto es necesaria para lograr mayores financiamientos, incluso, formas de contribución y actores involucrados, pero aún muchos desarrolladores no quieren o no han logrado simplificar y adecuar sus canales de comunicación con los usuarios 'legos'. Muchos estudios demuestran que la retroalimentación de los usuarios en el proceso de producción lleva a la mejora de los productos y es una fuente de innovación. Si bien es cierto que el uso de *software* libre es una invitación al usuario a ser más activo y reflexivo al usar su ordenador, también es cierto que, en muchas ocasiones, algunas aplicaciones resultan complicadas, incluso, inaccesibles si no se tienen conocimientos avanzados en programación.

Las redes de producción enfrentan otro problema al hablar de la sinergias y alianzas con empresas, puesto que es necesario

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

definir la relación que se tendrá entre la producción de *software* de manera colectiva, convirtiéndolo en un bien común, y las necesidades mercantiles sobre los beneficios monetarios que necesitan las empresas.

[sobre] las incertidumbres sobre la relación entre la empresa de negocios y la comunidad desarrolladora de software de fuente abierta. El discurso intelectual y práctico que rodea el asunto de los bienes comunes ha dejando una ambigüedad importante y realmente básica sobre cómo esta relación debe estructurarse. (Weber, 2004: 217)

Una de las interrogantes más importantes en estos momentos tiene que ver con la transición hacia los servidores remotos de muchas aplicaciones, las así llamadas “nubes”. El uso de estas aplicaciones presenta incertidumbres y confusión para muchos usuarios de *software* libre y abierto y la cuestión tiene que ver con el manejo y el control de la información. Para Stallman, la situación se plantea de la siguiente manera:

Hay muchas maneras de usar el internet en la informática, y cada manera plantea sus asuntos éticos. [...] Las cosas que son capaces de hacerse en tu computadora, debes hacerlas en tu computadora con un programa libre para tener el control de ella. Hay otras cosas [que se hacen en internet] como la comunicación. Cuando te comunicas con otros, no es tu propia informática. No puedes hacerlo dentro de tu computadora. Es otro asunto [...] Hay cosas para

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

las cuales el uso del internet, no lleva a abusos ni peligros, entonces usémoslos, pero para tu informática, debes rechazar cualquier oferta de hacerla en el servidor de otro, porque así él tiene el control de ella y tú no.

El asunto sobre las aplicaciones en servidores remotos lleva a reflexiones sobre el control de la información y la seguridad de ésta. Una forma de abordar el tema es, como lo hace Stallman, separando las actividades que se realizan en el ordenador, realizando sólo aquellas que tienen que ver con la comunicación en este tipo de aplicaciones-en-red. El asunto no es tan sencillo cuando estamos siendo bombardeados en todos los medios informáticos sobre los beneficios de “migrar” nuestra información a “las nubes”⁷⁰.

El movimiento de nube-abierta se convertirá en una de esas actividades cuyo negocio en realidad consiste

⁷⁰ El término “nube” se refiere a las aplicaciones que se manejan en Internet donde parte de los componentes se encuentran en un servidor remoto y no en la computadora personal; esto implica que se necesita conexión a Internet para poder acceder, mediante contraseñas, a la parte de la aplicación que está en línea. Ejemplo de estas aplicaciones son los servicios de respaldo en línea, como Dropbox. Una especie de aplicación en nube sería por ejemplo “gmail” o cualquier correo electrónico que guarda nuestros correos recibidos en un sitio virtual al que accedemos cuando entramos o nos conectamos a Internet. Actualmente, muchos proveedores de *software* proponen aplicaciones que se manejen en línea, lo que supuestamente evitaría la piratería y haría que nuestros ordenadores no tuvieran que tener tanta memoria RAM, pues las aplicaciones arrancarían a través de servidores remotos con gran capacidad de almacenaje.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

en habilitar a los usuarios para dejarlo. (Laisné, Aigrain, Bollier, & Tiemann, 2010: 11)

La transición a estos programas debe plantearse en cuanto al tipo de servicios de Internet. Aigrain (2012) los clasifica como 1) *aplicaciones de software como servicio que sólo recentralizan y que incluyen funcionalidades de valor agregado que fácilmente podrían correr en el servidor bajo el control del usuario, y 2) servicios como máquinas globales de búsqueda que, según las condiciones actuales de la tecnología, requieren de procesos de centralización de los datos. (Laisné et al., 2010: 16)*

En el primer caso, el consejo es evitar el uso; en el segundo, se proponen políticas públicas, vigilancia de las comunidades y organizaciones públicas para su regulación y mantenimiento como alternativas libres.

Nuevamente, la propuesta de las redes de FLOSS sobre la reflexión de nuestra forma de interactuar con los ordenadores puede darnos alguna respuesta. Al final todo depende de la seguridad que busquemos para nuestra información y quién queremos que tenga el control sobre ella. Si ponemos sobre la mesa el asunto de la seguridad la reflexión está sobre lo que estamos dispuestos a mostrar al mundo o no. Siguiendo la regla 17 de (E. Raymond, 1999: 18):

Un sistema de seguridad sólo es seguro mientras es secreto. Cuídate de pseudo-secretos.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Parte de los beneficios que otorga el *software* libre y abierto, como la libertad de modificación, en el imaginario colectivo de la mayoría de los usuarios 'legos' no se ve como beneficio. Existe una gran cantidad de información errónea sobre la utilización de FLOSS, que lo presenta como *software* para desarrolladores; es decir, que sólo puede ser utilizada por gente que sabe de programación.

Si bien es cierto que en el principio esto pudo haber sido así, cada vez hay más aplicaciones que dejan sin sustento este argumento. El problema principal para la adopción generalizada de los programas libres tiene que ver con la forma en que los usuarios, en su mayoría, se relacionan con su ordenador, y esto es resultado del tipo de aplicaciones que aprendieron a utilizar de inicio y lo que éstas les permitían, incluso, exigían para desarrollar sus propias habilidades.

Una lección importante de estas redes de desarrollo de *software* libre es que no se necesita ser programador para desarrollar código. Es deseable y mucho más fácil hacerlo si se tienen estos conocimientos, sí, pero las habilidades pueden adquirirse si así se desea. Las aplicaciones libres de manera primordial lo que hacen es invitarnos a pensar, a reflexionar sobre lo que queremos hacer y cómo queremos hacerlo, y no nos restringen a aceptar tan sólo las formas de hacer predeterminadas por el fabricante.

Conclusiones

La expansión del movimiento de *software* libre y abierto en países europeos y sudamericanos, en contraste con el desconocimiento casi generalizado del mismo en México, me motivó a usar y buscar conocer el origen y desarrollo de este tipo de programas y aplicaciones, sus fundamentos y formas de organización en la infraestructura virtual. La observación de las formas de producción y organización, así como de la cooperación voluntaria de miles de usuarios anónimos para desarrollar, incluso ayudar a resolver problemas, me llevó a estudiar el tema académicamente, buscando responder en esta tesis de maestría, principalmente, dos preguntas: 1) ¿Cómo y por qué funcionan las redes de producción, distribución e intercambio de *software* libre y abierto, en tanto interacción entre usuarios y desarrolladores? Y, tomando en cuenta el debate sobre los bienes comunes, ¿cómo se inserta la producción de *software* libre y abierto en la dinámica del capitalismo del siglo XXI?

Para responder a estas preguntas, en un primer momento de la investigación, se describió el contexto histórico y social en el que emergió y permanece el fenómeno del *software* libre y abierto (FLOSS por sus siglas en inglés). El conocimiento de la historia del *software* libre y abierto fue necesario para

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

comprender las transformaciones y cambios del fenómeno estudiado, en tanto que éste es producto de procesos sociales. En este sentido, este trabajo mostró cómo el cierre de los códigos de programación devino de un proceso en el que emergieron intereses mercantiles. Esto se ejemplificó con la narración de lo sucedido con el sistema operativo Unics. También a través de la historia, se describió cómo la lucha por la libertad en el *software* emergió casi al mismo tiempo en que se iniciaron las restricciones en el mismo. Una de las figuras trascendentes en esta fase, y aún ahora, es Richard Stallman, a través del movimiento denominado *copy-left* y la Fundación de *software* Libre.

Para esta tesis se propuso mirar el *software* libre y abierto y su proceso de producción, mantenimiento y distribución, como nuevo bien común del conocimiento, ya que, en principio, son las propias comunidades de desarrollo las que así lo definen en sus prácticas, a través de reglas formales e informales, como se mostró con los trabajos de Charles M. Schweik en el apartado 2.5 de este estudio. En este sentido, la inserción del *software* libre y abierto en el mercado global no sólo toma en cuenta el *software* como producto, sino también como proceso, en tanto que es resultado de la acción colectiva de un gran número de personas.

Apoyándome en trabajos previos sobre la producción de FLOSS, describí la interacción entre desarrolladores y usuarios, lo que facilitó la descripción de la forma en la que los usuarios inciden en la innovación y mejoramiento de las aplicaciones. En el análisis del discurso para esta tesis de Richard Stallman, Karl

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Fogel y, principalmente, Eric Raymond, destacó la importancia que los desarrolladores dan a los usuarios en partes fundamentales del proceso de producción, tanto para retroalimentación, como para proponer componentes. Esta interacción se hizo visible en las redes de producción en los momentos en los que los diagramas desarrollados para este trabajo mostraban mayores interacciones, dando a entender un proceso de revisión entre pares o retroalimentación de usuarios sobre un programa integrado.

Posteriormente, se plantearon las herramientas teórico-metodológicas para abordar el análisis de las redes y las motivaciones de los actores. En tanto que la producción de FLOSS se organiza en la infraestructura virtual dentro de repositorios, esto es, sitios de Internet que dan alojamiento y servicios para desarrollo de programas y aplicaciones, se buscó la obtención de datos sobre colaboración en el desarrollo de proyectos de *software* libre y abierto de estas plataformas virtuales.

Elegí recabar los datos del sitio *Flossmole.org*, ya que es una plataforma que aloja con cierta sistematización los datos de varios repositorios. El tipo de datos que se pueden encontrar en el sitio se divide en los diferentes repositorios y va desde listados de desarrolladores y proyectos, tipos de proyectos y licencias, tiempos de inicio, trabajo en los espacios de intercambio, número de problemas resueltos y a resolver, etcétera.

Del universo de datos expuesto en el sitio *Flossmole.org*, seleccioné los listados de desarrolladores y proyectos de seis

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

repositorios de *software* en Internet en un corte de tiempo bimestral: noviembre-diciembre de 2011. Estos datos se recogieron sólo con la petición de vincular desarrolladores a proyectos. Fue mi labor depurar los datos duplicados y crear una base para relacionar, delimitar y seleccionar. A partir de los datos y tras el trabajo con los mismos, pude mostrar el tamaño de la colaboración en cada uno de los repositorios en ese corte de tiempo. Con los mismos datos expuse visualmente las distintas dinámicas en red que pueden emerger en diferentes momentos del proceso de desarrollo de proyectos de *software* libre y abierto.

Para complementar el análisis de redes, recopilé documentos de, y realicé una entrevista a, personajes clave del fenómeno observado para examinar sus motivaciones para cooperar, percepciones y perspectivas sobre el futuro del *software* libre y abierto. A continuación expongo las conclusiones a las que se llegó con este planteamiento de análisis.

La inserción del *software* libre y abierto en la dinámica del capitalismo del siglo XXI, sucede en el contexto de la emergencia de las tecnologías de información y comunicación, así como de la digitalización de los mercados. El presente estudio partió de la conceptualización del *software* libre y abierto como una forma alternativa de organización para la producción, difusión, distribución y mejoramiento de un producto colectivo. Tras la observación de la dinámica en los repositorios, puedo decir que la organización del trabajo para el desarrollo de FLOSS no es del todo diferente a la organización en el desarrollo de *software* privativo, en tanto que en ambos, como en cualquier proceso

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

laboral, existe una división del trabajo que implica momentos en los que los procesos se desarrollan más de manera individual y otros en los que se necesita de la colaboración en equipos. La diferencia está en que el producto del trabajo es colectivo, puede compartirse libremente y, al mismo tiempo, generar ganancias con mercados de servicios a su alrededor.

Un proyecto exitoso de software libre y abierto implica no sólo o no necesariamente que éste se inserte en el mercado y genere ganancias a partir de donaciones, venta o los servicios emanados a su alrededor, sino también que cumpla una función de utilidad o resolución de problemas para la comunidad de desarrolladores y usuarios, que haga avanzar el lenguaje de programación para generar nuevas aplicaciones, que genere interés por parte de la comunidad hacker, o bien, que se convierta en un elemento lúdico o de reto para los propios desarrolladores y usuarios. En sí, un proyecto exitoso se basa más en la generación de comunidades y redes de desarrollo a su alrededor que de tener simplemente un producto terminado, susceptible de generar ganancias. El reto más importante para un proyecto de software libre y abierto es generar interés.

La propuesta libre y abierta, a diferencia de la privativa, cuando los proyectos de desarrollo son exitosos, permite la sustentabilidad de los productos, pues el interés de la comunidad en su mantenimiento los mantendrá vivos y en evolución. Pero, además, reparte las ganancias económicas, cuando las hay, entre una base más amplia de actores; es decir, el trabajo de los desarrolladores no puede enajenarse por una sola persona o entidad, puesto que el código fuente está expuesto y, según el tipo de licencia, más o menos protegido en

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

contra de su clausura o cierre. Además, por la forma en que puede distribuirse y re-diseñarse, permite que un mayor número de personas tengan acceso a aplicaciones y programas. Si bien es cierto que se necesita conocimiento técnico y habilidades de programación para manipular el código, el que éste se encuentre 'a la vista' posibilita que se adapte y utilice de diferentes maneras.

El uso de software privativo implica la dependencia del usuarios hacia los desarrolladores de sus programas, puesto que el usuario simplemente aprende a entender la interfaz de las aplicaciones; es decir, se le entrena para que sepa dónde debe oprimir para hacer limitadas funciones, pero no se le enseña a resolver problemas específicos que van más allá de la interfaz. Por ejemplo, un usuario común de procesadores de texto privativos sabe cómo imprimir, cómo cambiar el tamaño de las fuentes o su tipo, incluso puede saber cómo hacer plantillas de documentos, etcétera, pero no sabe lo que sucede cuando la aplicación se congela y se cierra inesperadamente. Este tipo de usuario no sabe cómo resolver el problema si la aplicación se cierra inesperadamente cada que intenta 'salvar' su documento, independientemente, de que tampoco sabe si su aplicación contiene alguna orden oculta que envía información sobre el usuario a algún servidor cada vez que se conecta a Internet.

Considero mediante este estudio que el uso abierto y libre de aplicaciones y programas es necesario para fomentar el conocimiento tecnológico y las habilidades técnicas de la población en general. A diferencia del uso de software privativo, que sólo necesita de entrenamiento más o menos básico, el software libre y abierto permite a los usuarios entender los

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

procesos que ocurren al usar ciertas aplicaciones; además de que la interacción con desarrolladores y otros usuarios invita y facilita el aprendizaje del uso de comandos, y con ello fomenta la independencia del usuario en el uso de sus aplicaciones.

La apertura de los códigos de programación posibilita el aprendizaje y, con ello, la innovación tecnológica en cada país, lo que podría llevar, si las condiciones materiales y la forma de intercambio lo permiten, a la disminución de la brecha digital y de la dependencia tecnológica de los países pobres y en vías de desarrollo.

El *software* libre y abierto ha demostrado su capacidad de sustentabilidad a través de alianzas entre las comunidades específicas y empresas o Fundaciones, p.e. *Red Hat*, *Cygnus*, *Canonical*, *FSF*. A partir de estas alianzas y apoyos, ha incrementado a lo largo del tiempo sus comunidades y redes de producción, así como su base de usuarios, lo que puede comprobarse con la utilización del sistema *GNU-Linux*, y aplicaciones derivadas, en los sistemas del departamento de defensa estadounidense, así como varios laboratorios, gobiernos y empresas. Los datos del apartado 1.4 del 1^{er}. capítulo de esta tesis, mostraron también esta aseveración, a través de los ejemplos sobre el uso del servidor Apache y de los navegadores *Firefox* y *Google Chrome*.

A partir del análisis de los datos de los seis repositorios estudiados, se observó que el desarrollo de *software* libre y abierto se realiza a través de la colaboración en redes de desarrolladores y proyectos sobre una infraestructura virtual que

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

se divide, por intereses y motivaciones compartidas entre los miembros de las distintas comunidades. Estas comunidades, así como los espacios virtuales en los que interactúan, cuentan con instituciones, en tanto reglas formales e informales⁷¹. Además de los estudios realizados por otros investigadores sobre las reglas dentro de la organización de los proyectos, en esta tesis se pudo observar que cada repositorio tiene reglas específicas sobre el tipo de proyectos que aloja para su desarrollo (ver subapartado 4.2 del 4º capítulo en esta tesis). Asimismo, cada equipo de desarrolladores, en lo general, organiza la división del trabajo y los tiempos de entrega, así como de liberación y revisión entre pares.

Finalmente, la producción y uso del *software* libre y abierto está sustentada sobre una infraestructura cultural, la cultura *hacker* –descrita en el apartado 3.3.1 del 3º. capítulo en este estudio–, que preexiste a, y sobrevive en el contexto virtual. Esta cultura *hacker* tiene reglas tácitas de comportamiento, así como formas específicas de entender y dar sentido a sus acciones. Un claro ejemplo de esta cultura *hacker* se mostró en el capítulo 3 con la presentación de las reglas en el desarrollo de proyectos que Eric Raymond desarrolló en su libro *La Catedral y el Bazaar* (1999). Es en el sentido que los propios actores dan a sus acciones que se buscó entender las motivaciones para la cooperación.

La Teoría del Esquema y el Conexionismo, que pretende la comprensión de los elementos cognitivos interconectados que

⁷¹Ver la descripción de los trabajos de Charles M. Schweik en el capítulo 3 de esta tesis.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

orientan la acción de los individuos, permitió una aproximación al estudio de las motivaciones de los actores desde un ángulo diferente al de la explicación de la acción colectiva de las teorías basadas en la acción racional. En este trabajo, he presentado una aproximación a las motivaciones de los actores, y propuesto un modelo de análisis con base en la Teoría del Esquema y el Conexionismo. Se observó que las razones que tienen los actores para participar en proyectos libres y abiertos, dan forma a las interacciones dentro de las redes de colaboración y por esto es que son importantes para la investigación.

Según el análisis realizado, las motivaciones expresadas por los actores para colaborar en este tipo de programas y aplicaciones corresponden a cuatro tipos, que se identificaron con cuatro conceptos o elementos cognitivos, a saber: motivaciones de carácter técnico, que tienen que ver con el ejercicio de la creatividad; motivaciones de carácter individual-colectivo, que tienen que ver con la identidad social y el sentido de pertenencia; motivaciones de carácter práctico, que se identificaron con la reproducción o sustentabilidad; y motivaciones de carácter ideológico, que son parte de la identidad política.

Las diferentes motivaciones de los actores no aparecen de manera aislada, sino que se retroalimentan entre sí. Dicho de otro modo, la existencia de nociones dentro de una categoría conceptual del esquema propuesto no excluye la existencia de argumentos que sustenten otras categorías. Todos los actores manejan de manera más o menos consciente el mismo esquema cognitivo, aunque sólo se enuncien algunas ideas en el discurso que pertenecen a una o dos categorías.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Sostengo, a partir de lo observado, que las pautas identificadas como motivaciones de los actores, esquematizadas en el modelo antes expuesto, se compone de elementos de carácter técnico, práctico e ideológico y la forma en la que se mezclan estos elementos inclina a mayor o menor colaboración, compromiso y/o espera de retribuciones de algún tipo. Un ejemplo de cómo se expone el esquema en el discurso y la acción individual lo podemos encontrar en los argumentos de Stallman quien, por la forma en la que expresa sus ideas sobre la libertad de los usuarios, muestra que las motivaciones de carácter ideológico son las más fuertes en su esquema cognitivo, lo que lo lleva a actuar en consecuencia, defendiendo ideológicamente el uso de *software* libre. En contraposición, están los argumentos de Eric Raymond quien, muestra más en su discurso motivaciones de carácter técnico, que tienen que ver con la creatividad y la resolución de problemas específicos. Es justamente este interés mayor por lo técnico lo que lo lleva a encabezar, junto con otros, el movimiento de fuente abierta, que pone el énfasis en la apertura de códigos para el mejoramiento de las aplicaciones, más que para defender la libertad de los usuarios.

Se propone como hipótesis de trabajo que la preeminencia de una categoría sobre otra puede explicar las diferencias de comportamiento en los distintos grupos de desarrolladores que conviven en Internet, y esto puede ser fundamental para las formas de organización de la producción y las dinámicas de interacción en las redes, así como para explicar los conflictos que surgen dentro del movimiento de *software* libre y abierto y,

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

más específicamente, dentro de las propias comunidades al momento de desarrollar un programa.

El esquema desarrollado para el análisis en este trabajo se presenta como una aportación y, al mismo tiempo, una propuesta de modelo de análisis para un estudio cualitativo a mayor profundidad. Un análisis, a partir del modelo aportado, podría ayudar a explicar qué tanta influencia tiene la forma en la que los individuos dan sentido a sus acciones en la configuración y resultados de la producción misma, en la organización social de ésta y en la construcción social de este tipo de procesos.

Con base en la teoría del interaccionismo estructural y el marco teórico-metodológico del Análisis de Redes Sociales fue posible identificar y describir las relaciones entre desarrolladores en seis repositorios de internet durante noviembre y diciembre de 2011. Estos marcos teóricos fueron de utilidad para la observación y análisis de los datos en un nivel más amplio de lo que los estudios neo-institucionales me permitían. El Análisis de Redes Sociales posibilita el estudio a diversos niveles de agregación, por lo que resulta útil y, tal vez, podría ser complementario a los estudios neo-institucionales sobre las reglas. En el caso del marco teórico del esquema y el conexionismo, creo que su utilidad radicó, para este estudio, en que permite ver las diferencias en la colaboración, y no sólo explicar su existencia o ausencia en un momento dado.

La visualización de las redes de producción permitió observar distintos modos de organización en diferentes proyectos. El análisis de los datos y la observación dentro de los repositorios

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

permitió definir que lo que se denomina “proyecto” en estas plataformas se refiere a componentes o partes de un programa, por lo que se pudo determinar que el trabajo de los desarrolladores se da de manera aislada y en colaboración más o menos amplia, según momentos del proceso de producción. Si bien es cierto que los datos mostraron que la mayor parte del trabajo en los repositorios se da de manera aislada, este estudio mostró que este trabajo aislado corresponde a momentos del proceso, pues en el desarrollo de código, la colaboración se da en el ensamblaje del mismo y no en la generación de los distintos componentes o proyectos. Las formas que presentaron las redes de desarrollo indicaban no sólo formas de organización distintas, sino momentos del proceso diferentes de los distintos programas divididos en proyectos.

La producción de *software* libre y abierto se presentó como una forma de organización flexible, en la que el siguiente paso no está predeterminado, sino que se define conforme a las situaciones emergentes y condiciones del producto en cada momento. La observación y el análisis en este trabajo mostraron que los datos de colaboración en los repositorios están muy desagregados; es decir, muestran el trabajo en proyectos, que son componentes de programas y aplicaciones. Debido a esta desagregación, el trabajo parece desarrollarse de manera aislada por parte de los desarrolladores, no se muestra de manera mayoritaria el trabajo en colaboración. Sin embargo, la cooperación sucede y lo hace de modo importante cuando se observan los proyectos integrados; es decir, cuando se unen los componentes que son parte de una aplicación completa, pues es aquí cuando emergen las redes colaborativas.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Una de las principales aportaciones de este trabajo al estudio del fenómeno de *software* libre y abierto es la representación visual de sus redes de producción. Pocos estudios se encontraron que hablen sobre la colaboración en redes para el desarrollo de FLOSS; aquellos que lo mencionan (Bría Ramírez, 2007; del Rivero Herrera & Ricapito, 2008), no basan su estudio en ellas ni las presentan como diagramas visuales.

La morfología de las redes, emanada de las formas de organización de los proyectos en los repositorios, mediante la diagramación, se mostró diversificada; es decir, distinta en las distintas plataformas, aunque con similitudes. En el caso de los repositorios que aceptaban programas privativos, se veía una sub-red conexas y varios equipos trabajando de manera aislada; mientras, que en los repositorios que sólo aceptan el desarrollo de aplicaciones libres y abiertas, las redes estudiadas estaban completamente conectadas. Esto resulta lógico porque sólo los programas libres y abiertos otorgan acceso al código fuente, y por tanto, tienen mayor libertad para generar y aceptar equipos colaborativos, tanto para desarrollar y mejorar el código, como para crear bifurcaciones y nuevos programas a partir de una sola propuesta. Sin embargo, como se mencionó, deben verse las distintas dinámicas como diferentes momentos en el proceso de producción, más que como características de una forma de organización en los repositorios.

Uno de los principales obstáculos para la utilización de los datos fue la cantidad de los mismos. Su manejo en bases de datos resultó complicado, como se explicó en la guía metodológica del capítulo 4. Debido a esto, solamente se pudo elegir un corte temporal de cada uno de los repositorios. Un

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

estudio que permitiera ver y comparar más cortes temporales e identificar los programas que integran a los proyectos existentes, podría proporcionar una descripción más completa de las dinámicas de las redes con un nivel de agregación mayor.

La trascendencia de este tipo de estudios tiene que ver con que posibilitan la identificación de mejores prácticas y si éstas son susceptibles de trasladarse a otro tipo de formas de producción, como puede ser la producción de conocimiento o las plataformas educativas de enseñanza-aprendizaje. Además ofrecen un conocimiento más amplio sobre la tecnología que, en materia de programación, se está desarrollando a nivel mundial. La forma de inserción de México en la producción de *software*, a partir de este tipo de investigaciones, podría recapitularse, con base no sólo en la reflexión sobre nuestras capacidades y necesidades, sino también en la posibilidad de generar redes de desarrollo de habilidades y conocimiento técnico que se basen en una producción más horizontal y equitativa, y que aprovechen el intercambio a nivel mundial a través de Internet.

En tanto que existen pocos estudios en México sobre este fenómeno, esta tesis aporta una nueva línea de investigación desde los Estudios Políticos y Sociales para ampliar la base de conocimiento sobre esta temática. Creo que este trabajo explica las dinámicas de desarrollo de FLOSS, pero deja abierta la puerta para un análisis de las mismas a lo largo del tiempo; asimismo, propone un estudio más amplio y profundo sobre las motivaciones de los actores, y cómo éstas influyen en la configuración de las diferentes comunidades, a partir del esquema desarrollado.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Finalmente, espero que esta investigación sea de utilidad para la sociedad, en tanto que difunde formas alternativas de interacción con la tecnología computacional, que no motivan a la dependencia de los usuarios, sino que los invitan a un uso más racional y personalizado de sus aplicaciones.

Referencias:

- Amin, S. (1997). *Capitalism in the Age of Globalization*. Londres: Zed.
- Amin, S. (2000). La economía política del siglo XX. *Globalización. Revista mensual de Economía, Sociedad y Cultura*, (junio), 1-8.
Recuperado de <http://www.rcci.net/globalizacion/2000/fg129.htm>
- Ashton, T. S. (1965). *La Revolución Industrial*. México, D.F.: Fondo de Cultura Económica.
- Barker-Plummer, D. (2011). Turing Machines. *Stanford Encyclopedia of Philosophy*. Recuperado de <http://plato.stanford.edu/entries/turing-machine/>
- Bergin, T. J., & Gibson, R. G. (1996). *History of programming languages*. Addison Wesley.
- Bernal, J. D. (1997). La ciencia en la Historia. *Revolucion Industrial*. México: Nueva Imagen/Universidad Nacional Autónoma de México.
- Berry, D. (2005). The Commons as an Idea-Ideas as a Commons. *Free software magazine*, (1). Recuperado de http://www.freesoftwaremagazine.com/articles/commons_as_ideas?page=0,1
- Bijker, W. E., Law, J., & Latour, B. (1992). *Shaping technology / building society*. (W. E. Bijker & J. Law, Eds.). Massachusetts: The MIT Press.
- Bollier, D. (2002). Reclaiming the Commons. *Boston Review*, verano. doi:10.1353/nhr.2006.0074
- Bonaccorsi, A., & Rossi, C. (2002, November). Why open source software can succeed. *LEM Working papers*. Pisa, Italia. doi:10.1111/j.1467-629X.1980.tb00220.x
- Booth, D. R. (2010). *Peer Participation and Software. What Mozilla has to teach government*. *Digital Media*. London, England: MIT Press.
- Borge Holthoefter, J. (n.d.). Álgebra de boole: del silogismo aristotélico a los circuitos integrados. *A parte rei*, 25.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Bourdieu, P. (2000). *Las estructuras sociales de la economía*. Barcelona: Anagrama.
- Boyd, R., & Richerson, P. J. (2009). Culture and the evolution of human cooperation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1533), 3281-8. doi:10.1098/rstb.2009.0134
- Brown, J. S., & Vincent, T. L. (2008). Evolution of cooperation with shared costs and benefits. *Proceedings. Biological sciences / The Royal Society*, 275(1646), 1985-94. doi:10.1098/rspb.2007.1685
- Bría Ramírez, M. B. (2007). “Reseña de ‘copy-left. Manual de Uso’ de VV.AA.” *Revista Athenea Digital*, No. 11((primavera, 2007)), 243-247. Recuperado de <http://redalyc.uaemex.mx/redalyc/src/inicio/ArtPdfRed.jsp?iCve=53701116>
- Buchan, N. R., Grimalda, G., Wilson, R., Brewer, M., Fatas, E., & Foddy, M. (2009). Globalization and human cooperation. *Proceedings of the National Academy of Sciences of the United States of America*, 106(11), 4138-42. doi:10.1073/pnas.0809522106
- Case, P. (2006). Aesthetics, performativity and resistance in the narratives of a computer programming community. *Human Relations*, 59(6), 753-782. doi:10.1177/0018726706066853
- Castells, M. (2001). *La galaxia Internet: Reflexiones sobre Internet, empresa y sociedad*. España: Plaza & Janés editores. Recuperado de <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:La+galaxia+internet#0>
- Castells, M. (2005). Economía Sociedad y Cultura. *La era de la Información*. Madrid: Alianza Editorial.
- Chandler, A. D. J., & Cortada, J. W. (2002). La era de la información. Continuidades y diferencias. *Una nación transformada por la información. Como la información ha modelado a EUA desde la época de la colonia hasta la actualidad*. Oxford University Press.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Chang, H.-J. (2010). Institutions and economic development: theory, policy and history. *Journal of Institutional Economics*, 1-26.
doi:10.1017/S1744137410000378
- Chase-Dunn, C. (2008). World-Systems Analysis. *Encyclopedia of Social Problems*. Sage Publications. Recuperado de <http://www.sage-reference.com/socialproblems/Article_n644.html>
- Christakis, N. A., & Fowler, J. H. (2010). *Conectados. El sorprendente poder de las redes sociales y cómo nos afectan* (pp. 1-32). México: Editorial Taurus.
- Ciuro, O. (2009). Industrias culturales e industrias creativas. *Knol. A unit for knowledge*. Recuperado de knol.google.com/k/industrias-culturales-e-industrias-creativas/
- Clippinger, J., & Bollier, D. (2005). A Renaissance of the Commons: How the New Sciences and Internet are Framing a New Global Identity and Order. In R. A. Ghosh (Ed.), *CODE: Collaborative Ownership and the Digital Economy*. Cambridge, MA: Massachusetts Institute of Technology.
- Copeland, B. J. (2008). The modern history of computing. *Stanford Encyclopedia of Philosophy*. Recuperado de <http://plato.stanford.edu/entries/computing-history/#Bab>
- Dasgupta, P. (2009). Trust and cooperation among economic agents. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1533), 3301-9. doi:10.1098/rstb.2009.0123
- De la Rúa, F. (2009). La perspectiva del interaccionismo estructural para el análisis de redes. *REDES-Revista hispana para el Análisis de redes sociales*, 17(12). Recuperado de <http://dialnet.unirioja.es/servlet/dcart?info=link&codigo=3165627&orden=243052>
- Dobb, M. (1996). La Revolución Industrial y el Siglo XIX. *Estudios sobre el desarrollo del Capitalismo*. México, D.F.: Siglo XXI editores.
- Ellul, J. (1964). *The technological society*. E.U.A: Vintage Books.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Elías, N. (1990). *La sociedad de los individuos*. Barcelona: Península.
- Entrevista a Mark Shuttleworth. (n.d.). Software Freedom Day.
- Fernandez, R., Gonzalez-Barahona, J. M., Hierro, J. J., Martinez, A., Polo, A., Reyes, M., & Soriano, J. (2011). The Networked Forge: New Environments for Libre Software Development. *IFIP Advances in Information and Communication Technology (AICT)*, 275(275), 299–306. Recuperado de <http://dl.ifip.org/index.php/AICT/article/view/19076>
- Fernández M., D. (2011). Uso de ODR en las comunidades de desarrollo de software libre. *UOC Repositorio institucional*. doi:10.1063/1.3037039.4
- Fernández Santillan, J. (1996). *Locke y Kant. Ensayos de filosofía política*. México, D.F.: Fondo de Cultura Economica.
- Fogel, K. (2005). *Producing open source software: How to run a successful free software project*. O'Reilly Media, Inc. Recuperado de <http://portal.acm.org/citation.cfm?id=1121560>
- Forester, T. (1992). *La sociedad de alta tecnología*. México, D.F.: Siglo XXI editores.
- Freeman, L. C. (2004). *The development of social network analysis. Social Networks* (Vol. 27). Vancouver, BC Canada: Empirical Press.
- Friedkin, N. E. (2004). Exchange Networks. *Encyclopedia of Social Theory*. Sage Publications. Recuperado de http://www.sage-reference.com/socialtheory/Article_n99.html
- Gilbert, M. (2006). Rationality in Collective Action. *Philosophy of the Social Sciences*, 36(1), 3-17. doi:10.1177/0048393105284167
- Goldstine, H. (1972). *The computer from Pascal to von Neuman*. Princeton University Press.
- Goldthorpe, J. H. (1998). Rational action theory for sociology. *The British Journal of Sociology*, 49(2), 167–192. JSTOR. Recuperado de <http://www.jstor.org/stable/591308>
- González Casanova, P. (2004). Comunidad: la dialéctica del espacio. *Temas, enero-marz(36)*, 4-15. Cuba. Recuperado de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- <http://www.temas.cult.cu/revistas/36/36004.pdf>
- González Soto, A. P., Gisbert, M., Guillén, A., Jiménez, B., Lladó, F., & Rallo, R. (1996). Las nuevas tecnologías en la educación. In S. et Al. (Ed.), *Redes de comunicacion, redes de aprendizaje. EDUTEC'95* (pp. 409-422). Palma: Universitat de les Illes Balears.
- Gómez Sánchez, R. (2008). Software libre vs. software propietario: programando nuestro futuro. *Historia Actual Online*, 2(2), 125. Recuperado de <http://historia-actual.org/Publicaciones/index.php/haol/article/viewArticle/24>
- Harhoff, D., Henkel, J., & Von Hippel, E. (2003). Profiting from voluntary information spillovers : How users benefit by freely revealing their innovations. *Research Policy*, 32, 1752-1769. Recuperado de <http://web.mit.edu/evhippel/www/papers.htm>
- Harris, H. (2007). Networking. *Encyclopedia of Business Ethics and Society*. Sage Publications. Recuperado de http://www.sage-reference.com/ethics/Article_n573.html
- Helfrich, S. (2008). *Genes, bytes y emisiones: bienes comunes y ciudadanía* (pp. 1-338). México: Fundación Heinrich Böll.
- Hemetsberger, a., & Reinhardt, C. (2009). Collective Development in Open-Source Communities: An Activity Theoretical Perspective on Successful Online Collaboration. *Organization Studies*, 30(9), 987-1008. doi:10.1177/0170840609339241
- Hemetsberger, A. (2006). Learning and Knowledge-building in Open-source Communities: A Social-experiential Approach. *Management Learning*, 37(2), 187-214. doi:10.1177/1350507606063442
- Hess, C. (2008). Mapping the New Commons. “*Governing shared resources: connecting local experience to global challenges.*” Cheltenham, Inglaterra: Asociación Internacional para el Estudio de los Bienes Comunes. Recuperado de <http://ssrn.com/abstract=1356835>
- Hobsbawm, E. (2004). *En torno a los orígenes de la revolución industrial*. México, D.F.: Siglo XXI editores.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Howison, J., Conklin, M., & Crowston, K. (2006). FLOSSmole: A collaborative repository for FLOSS research data and analyses. *International Journal of Information Technology and Web Engineering*, 1(3), 17-26.
- ITU. (2011). *Measuring the information society 2011. Organizacija znanja* (Vol. 8). Geneva, Switzerland. doi:10.3359/oz0303157
- Illich, I. (2006). *Obras reunidas*. México, D.F.: Fondo de Cultura Economica.
- Jesiek, B. K. (2006). The Sociotechnical Boundaries of Hardware and Software: A Humpty Dumpty History. *Bulletin of Science, Technology & Society*, 26(6), 497-509. doi:10.1177/0270467606295002
- Laisné, J.-P., Aigrain, P., Bollier, D., & Tiemann, M. (2010). *2020 FLOSS Roadmap*. 2020FLOSSRoadmap.org.
- Lakhani, K., & Hippel, E. V. (2000). How open source software works: “free” user-to-user assistance. *Working Paper*, 4117, 1-39.
- Lakhani, K. R., & Wolf, R. G. (2005). Why hackers do what they do: understanding motivation and effort in Free/Open source software projects. In J. Feller, B. Fitzgerald, S. Hissam, & K.R.Lachani (Eds.), *Perspectives on free and open source software*. Boston, MA: MIT Press. Recuperado de <http://freesoftware.mit.edu/papers/lakhaniwolf.pdf>
- Latour, B. (1992). Where are the missing masses? The Sociology of a few mundane artifacts. In W. E. Bijker & J. Law (Eds.), *Shaping Technology/Building Society. Studies in sociotechnical change* (pp. 225-258). Cambridge, Massachusetts: The MIT Press.
- Lee, J. A. N. (2011). History of computing. *History of Computing*. Retrieved June 18, 2011, from <http://ei.cs.vt.edu/~history/>
- Lessig, L. (2004). *Free culture: How big media uses technology and the law to lock down culture and control creativity*. Creativity. Nueva York: The Penguin Press.
- Lessig, Lawrence. (2001). *The future of ideas*. *Duke Law Journal* (1a.

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- edició., Vol. 52, p. 368). Nueva York: Random House. Recuperado de <http://opensource.sfsu.edu/files/lessig-talk.pdf>
- Lizama M., J. (2002). Hackers: de piratas a defensores del software libre. *Revista Mexicana de Sociología*, 45(185), 91-108.
- Locke, J. (2005). *Ensayo sobre el gobierno civil*. México, D.F.: Ediciones Gernika.
- MacKenzie, D., & Wajcman, J. (1996). *The social shaping of technology*. (D. MacKenzie & J. Wajcman, Eds.). Buckingham: Open University Press.
- Marx, K. (2000). *La ideología alemana*. México, D.F.: Quinto Sol.
- Mathew, S., & Boyd, R. (2009). When does optional participation allow the evolution of cooperation? *Proceedings. Biological sciences / The Royal Society*, 276(1659), 1167-74. doi:10.1098/rspb.2008.1623
- Mattelart, A. (2002). *Historia de la sociedad de la información*. Barcelona: Editorial Paidós.
- Milberry, K., & Anderson, S. (2009). Open Sourcing Our Way to an Online Commons: Contesting Corporate Impermeability in the New Media Ecology. *Journal of Communication Inquiry*, 33(4), 393-412. doi:10.1177/0196859909340349
- Moglen, E. (2004). "Die Gedanken Sind Frei": Free Software and the Struggle for Free Thought. *Wizards of OS 3* (pp. 1-14). Berlin: Columbia University.
- Molina, J. L., Ruiz, A. A., & Teves, L. (2005). Localizando geográficamente las redes personales. *Redes: Revista hispana para el análisis de redes sociales*, 8(5). Recuperado de <http://dialnet.unirioja.es/servlet/articulo?codigo=1303357>
- Mounier, P. (2002). *Los dueños de la red*. España: Editorial Popular.
- North, D. (1993). *Instituciones, cambio institucional y desempeño económico*. México: Fondo de Cultura Económica.
- Obregón, D. (2002). La construcción social del conocimiento : Los casos de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Kuhn y de Fleck. *Bosque*, 3(6-7), 41-58.
- Olson, M. (1992). *La lógica de la acción colectiva*. México, D.F.: Limusa / Grupo Noriega Editores.
- Ostrom, E. (1990). Analyzing long-enduring, self-organized, and self-governed CPRs. *Governing the commons. The evolution of institutions for collective action* (pp. 58-102). Cambridge, Massachusetts: Cambridge University Press.
- Ostrom, E. (2000). Private and common property rights. *Encyclopedia of law and economics* (pp. 332-379). Recuperado de [http://valuefronteira.com/vf/images/textbooks2/private and common property rights.pdf](http://valuefronteira.com/vf/images/textbooks2/private%20and%20common%20property%20rights.pdf)
- Ostrom, E. (2005). Governing a Commons from a Citizen 's Perspective. *Political Theory*, 1-9.
- Ostrom, E., Hess, C., Bollier, D., Levine, P., Schweik, C. M., & Pradt Lougee, W. (2007). *Understanding knowledge as a commons. From theory to practice*. (E. Ostrom & C. Hess, Eds.) *Journal of the American Society for Information Science and Technology* (Vol. 59). Cambridge, Massachusetts / Londres: The MIT Press. doi:10.1002/asi.20747
- Paasi, A. (2002). Boundaries in a Globalizing World. *Handbook of Cultural Geography*. Sage Publications. Recuperado de <http://www.sage-reference.com/hdbk_culturegeo/Article_n25.html>
- Polanyi, K. (2004). *La Gran Transformación*. México: Casa Juan Pablos.
- Postmes, T., & Brunsting, S. (2002). Collective Action in the Age of the Internet: Mass Communication and Online Mobilization. *Social Science Computer Review*, 20(3), 290-301. doi:10.1177/089443930202000306
- Powell, S. (2006, October). Entrevista a Tim Berners-Lee. *Emerald Now*, 1-5.
- Puurtinen, M., & Mappes, T. (2009). Between-group competition and human cooperation. *Proceedings. Biological sciences / The Royal Society*, 276(1655), 355-60. doi:10.1098/rspb.2008.1060

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Raymond, E. (1999). The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3), 23-49. doi:10.1007/s12130-999-1026-0
- Raymond, E. S. (1992). *The Hackers' Dictionary. The Jargon File, Version 2.9.10*. Project Gutenberg. Recuperado de <http://www.manybooks.net/titles/anonetext92jargn10.html>
- Rheingold, H. (2002). *The Virtual Community: Homesteading on the Electronic Frontier*. Cambridge, MA: The MIT. Recuperado de <http://www.rheingold.com/vc/book/>
- Ritchie, D. M. (1996). The Evolution of the Unix Time-sharing System. *AT&T Bell Laboratories Technical Journal*, 63(6, Parte 2), 1577-93. Nueva Jersey. Recuperado de <http://cm.bell-labs.com/cm/cs/who/dmr/hist.html>
- Rose, C. M. (2003). Romans, roads, and romantic creators: traditions of public property in the information age. *Law and Contemporary Problems*, 89. Recuperado de [http://www.law.duke.edu/shell/cite.pl?66+Law+&+Contemp.+Probs.+89+\(WinterSpring+2003\)#H1N1](http://www.law.duke.edu/shell/cite.pl?66+Law+&+Contemp.+Probs.+89+(WinterSpring+2003)#H1N1)
- Sampedro Hernández, J. L. (2011). *Conocimiento y empresa: la industria de software en México* (p. 244). México, D.F.: Universidad Autónoma Metropolitana-Cuajimalpa / Plaza y Valdes Editores.
- Schmidt, A. (2006). Social Network Theory. *Encyclopedia of Governance*. Sage Publications. Recuperado de http://www.sage-reference.com/governance/Article_n503.html
- Schweik, C. (2007). Tragedy of the FOSS commons? Investigating the institutional designs of free/libre and open source software projects. *Public Policy*. Amherst, MA. Recuperado de http://www.firstmonday.org/issues/issue12_2/
- Schweik, C.M., English, R., Paienjtton, Q., & Haire, S. (2010). Success and abandonment in open source commons: selected findings from an empirical study of Sourceforge. net projects. *Proceedings of the Sixth International Conference on Open Source Systems (OSS 2010) Workshops* (pp. 91–101). Recuperado de

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- http://works.bepress.com/charles_schweik/17
- Schweik, C. M., Fernandez, M. T., Hamel, M. P., Kashwan, P., Lewis, Q., & Stepanov, a. (2008). Reflections of an Online Geographic Information Systems Course Based on Open Source Software. *Social Science Computer Review*, 27(1), 118-129. doi:10.1177/0894439308320793
- Schweik, Charles M., & Kitsing, M. (2010). Applying Elinor Ostrom's Rule Classification Framework to the Analysis of Open Source Software Commons. *Review Literature And Arts Of The Americas*, 2(1), 13-26. Recuperado de http://www.tnc-online.net/pic/2010032809124697.pdf?referer=http://works.bepress.com/charles_schweik/16/
- Solís Leree, B. (coord. . (2003). *El Reto de México ante la Cumbre Mundial de la Sociedad de la Información*. (C. Solís Leree, Beatriz, Ed.). México, D.F.: Fundaci ón Konrad Adenauer, A.C.
- Squire, M., & Williams, D. (2012). Describing the Software Forge Ecosystem. *45th Hawaii International Conference on System Sciences*. Maui, Hawaii.
- Stallman, Richard M. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. *Computer Networks*. Boston, MA: GNU Press /FSF. Recuperado de <http://www.fsf.org>
- Stallman, R.M. (2004). *Software libre para una sociedad libre*. Madrid: Traficantes de Sueños. Recuperado de <http://www.ie.unan.edu.ni/~lenin/libros/softlibre.pdf>
- Stewart, D. (2005). Social Status in an Open-Source Community. *American Sociological Review*, 70(5), 823-842. doi:10.1177/000312240507000505
- Strauss, C., & Quinn, N. (1997). *A Cognitive Theory of Cultural Meaning*. Cambridge University Press.
- Trejo Delarbre, R. (2003). Relatoría. VI Conferencia Internacional. El Reto de México ante la Cumbre Mundial de la Sociedad de la Información. <http://raultrejo.tripod.com>. México: Raúl Trejo Delarbre. Retrieved October 10, 2011, from <http://raultrejo.tripod.com/Internetensayos/Relatoriainferenciasenadoma>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- yo03.htm#1. Definiciones
- Trejo-Delarbre, R. (2006). *Viviendo en El Aleph. La sociedad de la información y sus laberintos*. Barcelona: Gedisa.
- Tropp, L. R. (2004). What Benefits the Group Can Also Benefit the Individual: Group-Enhancing and Individual-Enhancing Motives for Collective Action. *Group Processes & Intergroup Relations*, 7(3), 267-282. doi:10.1177/1368430204046111
- Van Vleck, T. (2011). Multicians.org. *Multics History*. Retrieved June 18, 2011, from www.multicians.org/history.html
- Van Wendel de Joode, R., De Bruijn, J., & Van Eeten, M. (2003). *Protecting the Virtual Commons; Self-organizing open source communities and innovative intellectual property regimes. Information Technology & Law Series, TMC Asser Press, The Hague*. Asser Press. Recuperado de <http://dlc.dlib.indiana.edu/dlc/handle/10535/25>
- Vercelli, A., & Thomas, H. (2008). Repensando los bienes comunes: análisis socio-técnico sobre la construcción y regulación de los bienes comunes. *Scientiae Studia*, 6(3), 427-442. doi:10.1590/S1678-31662008000300010
- Von Hippel, E. (2007). Horizontal innovation networks—by and for users. *Industrial and corporate change*, 16(2), 28. Oxford Univ Press. Recuperado de <http://icc.oxfordjournals.org/content/16/2/293.short>
- Von Hippel, E., & Von Krogh, G. (2003). Open source software and the “private-collective” innovation model: Issues for organization science. *Organization Science*, 14(2003), 209–223. JSTOR. Recuperado de <http://www.jstor.org/stable/4135161>
- Von Hippel, Eric. (1988). *The sources of innovation. October* (Vol. 132). New York: Oxford University Press. Recuperado de <http://stuff.mit.edu/people/evhippel/books/sources/Front Matter.pdf>
- Von Hippel, Eric. (2005). *Democratizing innovation*. Cambridge, Massachusetts /Londres, Inglaterra: The Johns Hopkins University Press. Recuperado de <http://web.mit.edu/evhippel/www/books.htm>

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

- Wallerstein, I. (2000). Globalization or the Age of Transition?: A Long-Term View of the Trajectory of the World-System. *International Sociology*, 15(2), 249-265. doi:10.1177/0268580900015002007
- Walljasper, J. (2010). *All that we share: How to Save the Economy, the Environment, the Internet, Democracy, Our Communities and Everything Else that Belongs to All of Us*. E.U.A: The New Press.
- Weber, S. (2004). *The Success of Open Source*. *Canadian Journal of Political Science/Revue*. Cambridge, MA: Harvard University Press.
- Weeda, R. (2004). History of Informatics: Ada Lovelace. Holanda. Recuperado de <http://www.cs.ru.nl/~hans/Geschiedenis/Papers/Renske.pdf>
- Wexelblat, R. L. (ed). (1981). *Programming Languages: history and fundamentals*. Academic Press.
- Winner, L. (1987). *La ballena y el reactor. Una búsqueda de los límites en la era de la alta tecnología*. España: Gedisa.
- Xalabander Plantada, R. (2006). Las licencias Creative Commons: ¿ una alternativa al copyright? *UOC, marzo*(002). Recuperado de <http://dlc.dlib.indiana.edu/dlc/handle/10535/5712>
- del Rivero Herrera, A., & Ricapito, L. R. (2008). Cibercultura y comunicación móvil. *Veredas*, 17, 143-162.
- i Hernández, J. M. (2005). *Software libre: técnicamente viable, económicamente sostenible y socialmente justo*. *Infonomía* (pp. 1-191). Barcelona: Infonomia. Recuperado de <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Software+libre:+técnicamente+viable,+económicamente+sostenible+y+socialmente+justo#0>

Índice de ilustraciones

Ilustración 1: Comparativo desarrollo TICs 2008-2010.....	28
Ilustración 2: Síntesis de la historia de la informática y la computación (1600-2000).....	55
Ilustración 3: Estructura de Linux por Ramesh Natarajan Fuente: http://www.thegeekstuff.com/2010/09/linux-file-system-structure/	60
Ilustración 4: Red histórica del desarrollo de sistemas operativos, a partir de Unics 1969-2008. Elaboración propia.....	64
Ilustración 5: Síntesis del desarrollo de Internet. Elaboración propia.....	72
Ilustración 6: Porcentaje de individuos entre 15 y 74 años, que usan Internet y computadora Fuente: Measuring the Information Society (http://www.itu.int/ITU-D/ict/publications/idi/2011/).....	73
Ilustración 7: Uso de sistemas operativos (agosto 2011) Elaboración propia con datos de StatCounter < http://gs.statcounter.com/ >.....	77
Ilustración 8: Uso de navegadores (porcentajes agosto 2011)Elaboración propia con datos de StatcounterFuente: Statcounter < http://gs.statcounter.com/ >.....	79
Ilustración 9: Uso de servidores 2003-2011 Elaboración propia con datos del sitio Netcraft < http://news.netcraft.com/ >.....	80
Ilustración 10: Nuevos bienes comunes del conocimientoFuente: Hess, 2008:20	107
Ilustración 11: Proyectos y Usuarios por Repositorio hasta 2010 en escala logarítmica Fuente: (Squire & Williams, 2012).....	165
Ilustración 12: Historia de los repositorios en internet Fuente: (Squire & Williams, 2011: 8).....	166
Ilustración 13: Frecuencia de etiquetas en las forjas, junio 2011 (Squire&Williams,2012) Fuente: http://flossmole.org/content/everything-you-ever-wanted-know-about-software-forges-code-forges-june-2011	184

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Ilustración 14: Desarrolladores por repositorio (nov-dic 2011) Fuente: FLOSSmole Elaboración propia con base en los datos de FLOSSmole.....	192
Ilustración 15: Proyectos por repositorio (nov-dic 2011) Elaboración propia con base en los datos de FLOSSmole.....	192
Ilustración 16: Subred cohesiva: desarrolladores con 2 proyectos o más, colaborando con al menos otro desarrollador y número de proyectos con 2 o más desarrolladores.....	195
Ilustración 17: GitHub: desarrolladores con más de dos proyectos colaborando con al menos otro desarrollador (noviembre-diciembre 2011). Elaboración propia.....	201
Ilustración 18: Launchpad: desarrolladores con 2 o más proyectos, trabajando en colaboración (nov-dic 2011). Elaboración propia.....	204
Ilustración 19: ObjectWeb: desarrolladores con más de 2 proyectos en colaboración (nov-dic'11). Elaboración propia.....	206
Ilustración 20: RubyForge: desarrolladores en más de 2 proyectos en colaboración (nov-dic'11). Elaboración propia.....	208
Ilustración 21: Savannah: desarrolladores con más de 2 proyectos en colaboración (nov-dic'11) Elaboración propia.....	210
Ilustración 22: Tigris: desarrolladores con 2 o más proyectos en colaboración. Elaboración propia con Pajek (marzo 2012).....	211
Ilustración 23: Esquema cognitivo: conceptos clave sobre motivaciones de los actores. Elaboración propia.....	224

Índice alfabético

Acción colectiva.....	12, 15, 126, 132 s., 135 s., 238, 245
Aigrain.....	235
Análisis de redes sociales.....	129, 136, 140, 142 s., 161, 173 s., 231
Análisis de Redes Sociales.....	14, 139, 143, 247
Anderson.....	40, 45
Annabi.....	40
ARPA.....	61, 68, 71
ARS.....	129, 136, 139 ss., 145, 174
Ashton(1965).....	101
AT&T.....	60 ss., 65, 67
Babbage.....	47 s., 55 s.
Benkler.....	99
Berners-Lee.....	71, 180
Berners-Lee,	173
Berry.....	51
Bien común	12, 14, 43 ss., 83, 85, 94 s., 101, 104 ss., 108, 123, 125, 128, 131 s., 135, 157, 171, 220, 233, 238
Bienes comunes	13, 40, 44, 83, 85, 92, 94, 98 s., 103 ss., 108, 110, 113, 117, 119 ss., 124, 126, 132, 221, 233, 237
Bijker.....	18
Bill Gates.....	55, 81
Bollier.....	40, 44, 94, 105, 121
Bonaccorsi.....	40, 43
Boole.....	49 s.
BSD.....	62, 65 s., 119, 167
Castells.....	60, 152
Chang.....	116
Copyleft.....	118 s.
Creative Commons.....	44, 112, 117 ss., 180
Crowston.....	40, 182
De Brujin.....	43

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

De la Rúa.....	143 s.
Dobb(1996).....	100
EDVAC.....	52
Elías.....	123, 140
Ellul.....	20 s.
ENIAC.....	52
Fernández Santillán(1996).....	102
FLOSSmole.....	125, 169, 175, 178, 181, 184, 194
Fogel.....	42, 146, 181, 239
Fogel y.....	173
Forrester.....	52
Freeman.....	70, 140
Fundación de Software Libre.....	75
Gao.....	40
Globalización.....	86 ss.
GNU-GPL.....	119
GNU-Linux.....	44, 58, 62, 66, 77, 177, 243
González Casanova,.....	151
Hacker....	41, 62, 81 s., 132, 148, 150 ss., 157, 171, 181, 217 s., 221, 244
Hackers.....	68, 81, 148 ss., 155 s., 212, 215, 218, 221, 225
Hardin.....	102
Hemetsberger.....	40, 42 s.
Hess.....	94
Hobsbawm.....	100
Howison.....	40, 182
Illich.....	23, 30
Interaccionismo estructural.....	129, 161, 173, 231, 247
Jacquard.....	47
Jesiek.....	56
Karatzogianni.....	40
Kuhn.....	25, 109
Laboratorios Bell.....	53 s., 59 ss., 65, 70
Lakhani.....	40, 43, 155, 213 s.
Latour.....	22, 49
Law.....	18

Software Libre y abierto: comunidades y redes de producción digital de bienes comunes

Leibniz.....	46, 50
Lessig.....	40, 43 s., 67, 108, 180
Lessig,	173
Linux...36 s., 41, 44, 58, 62 s., 66, 71, 75, 77 s., 152, 172, 177, 180, 215, 218 s., 226 s., 243	
Lizama Mendoza.....	149
Lovelace.....	48, 56
Madey.....	40
Marx.....	24, 139
Michaelides.....	40
Microsoft.....	37, 55, 67, 76, 79, 81, 149, 174, 193, 226
Milberry.....	40, 45
Minix.....	65 s.
Mounier.....	61
Multics.....	54, 59
Neoinstitucionalismo.....	124
Neoliberalismo.....	87
Olson.....	134 s.
Ostrom.....	33, 105, 107, 119, 121, 123 ss.
Pajek.....	145, 175, 179 s.
Pascal.....	46, 55
Patentes.....	90, 107, 113 ss., 229
Perens.....	155
Polanyi.....	105
Prosoft.....	93
Qureshi.....	40, 42
Raymond.....	41, 138, 151, 155, 159, 180, 203, 218, 223, 239, 244, 246
Raymond,.....	173
Reinhardt.....	40, 42
Rossi.....	43
Sampedro.....	51
Schmidt.....	141
Schweik.....	14, 44, 123, 125 ss., 238
Shweik.....	40
Solaris.....	66

**Software Libre y abierto: comunidades y redes de producción
digital de bienes comunes**

Sourceforge.net.....	74, 178, 182
Stallman...38, 58, 61 s., 66, 69, 97, 110, 115 s., 118 s., 146, 154 ss., 180, 189, 197, 217, 220 ss., 228, 230, 233 s., 238, 246	
Stanley Jevons.....	49
Steve Jobs.....	55
Teoría de la Acción Racional.....	12, 133
Teoría del Esquema y el Conexionismo.....	14, 136, 231, 244 s.
Torvalds.....	41 s., 62 s., 146, 159, 180, 219 s.
Trejo Delarbre.....	91
Turing.....	51
Unics.....	54, 59, 65, 238
Unión Internacional de Comunicaciones.....	27
Unión Internacional de Telecomunicaciones.....	25
Unix.....	36, 54 ss., 58 ss., 65, 67 ss.
Usenet.....	68 ss., 150 s.
Van Antwerp.....	40
Van Eccen.....	43
Van Krogh.....	40, 43
Van Wendel de Joode.....	40, 43
Vercelli & Thomas.....	109
Vercelli y Thomas.....	96
Vint Cerf.....	146
Von Hippel.....	40, 43
Wiebe.....	18
Winner.....	21, 23 s., 26
Wolf.....	40, 43, 213
Zuse.....	51