



Universidad Nacional
de Mar del Plata

Universidad Nacional
de General Sarmiento 

Edición

4^{ta}

**Maestría en Economía
y Desarrollo Industrial**

Mención en la Pequeña y Mediana Empresa

*"Restricciones en la difusión de tecnologías abiertas. La
difusión de software libre en la Argentina"*

Estudiante: Verónica Robert

Director de Tesis: Gabriel Yoguel

Fecha de Defensa: 28 de Abril de 2006

Jurado:

Gabriel Yoguel

Fernando Porta

Ricardo Ferraro

2002-
2003

FORMULARIO "E" TESIS DE POSGRADO

Este formulario debe figurar con todos los datos completos a continuación de la portada del trabajo de Tesis. El ejemplar en papel que se entregue a la UByD debe estar firmado por las autoridades UNGS correspondientes.

1.1. Niveles de acceso al documento autorizados por el autor

El autor de la tesis puede elegir entre las siguientes posibilidades para autorizar a la UNGS a difundir el contenido de la tesis:

- a) Liberar el contenido de la tesis para acceso público.
- b) Liberar el contenido de la tesis solamente a la comunidad universitaria de la UNGS:
- c) Retener el contenido de la tesis por motivos de patentes, publicación y/o derechos de autor por un lapso de cinco años.

a. Título completo del trabajo de Tesis:

"Restricciones en la difusión de tecnologías abiertas. La difusión de software libre en la Argentina"

b. Presentado por (Apellido/s y Nombres completos del autor):

Verónica Robert

c. E-mail del autor: vrobot@gmail.com

d. Estudiante del Posgrado (consignar el nombre completo del Posgrado):

"Maestría en Economía y Desarrollo Industrial con mención en la PyME."

e. Institución o Instituciones que dictaron el Posgrado (consignar los nombres desarrollados y completos):

Universidad Nacional de General Sarmiento y Universidad Nacional de Mar del Plata.

f. Para recibir el título de (consignar completo):
"Magister en Economía y Desarrollo Industrial. Mención en la Pequeña y Mediana Empresa."

a) Grado académico que se obtiene: *Magister*

b) Nombre del grado académico: *"Economía y Desarrollo Industrial con mención en la Pequeña y Mediana Empresa."*

g. Fecha de la defensa: 28 Abril de 2006

h. Director de la Tesis (Apellidos y Nombres):

Gabriel Yoguel

i. Tutor de la Tesis (Apellidos y Nombres):

j. Colaboradores con el trabajo de Tesis:

k. Descripción física del trabajo de Tesis (cantidad total de páginas, imágenes, planos, videos, archivos digitales, etc.): *150 páginas*

l. Alcance geográfico y/o temporal de la Tesis:

Análisis de la difusión del SL en la Argentina a partir de una encuesta realizada durante el período 2003-2004

m. Temas tratados en la Tesis (palabras claves):

Tecnologías abiertas, Conocimiento, Bienes club, Difusión, Apropiación, Propiedad intelectual

n. Resumen en español (hasta 1000 caracteres):

En los últimos años ha surgido el interés de sociólogos, antropólogos y economistas por el fenómeno del software libre (SL) o software de fuente abierta (free software u *open source software*). El SL, a diferencia del propietario, se caracteriza por otorgar al usuario la libertad de usar el software sin restricciones, incluyendo la inspección,

modificación y redistribución de su código fuente sin el pago de licencias; desafiando 'leyes' que la economía considera centrales, entre ellas la racionalidad de los individuos objetivada en la búsqueda de la maximización de la utilidad.

Los objetivos generales de esta investigación son (i) comprender el fenómeno de SL desde una perspectiva histórica caracterizada por la evolución tecnológica, organizacional e institucional, y (ii) analizar el proceso de difusión del SL identificando restricciones asociadas al desarrollo de competencias. Un enfoque centrado en la difusión permite participar de la discusión sobre las formas de protección y apropiación, lo que está asociado a la cuestión de las motivaciones, y a la vez contribuye a comprender el fenómeno desde una perspectiva evolutiva caracterizada por el cambio, la incertidumbre, el aprendizaje y la adaptación.

o. Resumen en portugués (hasta 1000 caracteres):

Em recentes de anos, sociólogos, antropólogos e economistas mostrou o interesse pelo fenômeno do software livre (SL) ou *open source software* (OSS). O SL, ao contrário do proprietário, é caracterizado por oferecer a operadores o liberdade de usar o software sem restrições, incluindo a inspeção, modificação e redistribuição de seu código fonte sem qualquer pagamento de licenças; desta maneira, o fenômeno desafia de 'leis de economia' como o objectified racionalidade de los agentes na procura do maximization de utilidade.

Os objetivos gerais desta investigação são: entender o fenômeno de FS numa perspectiva histórica, caracterizado pelo organizacional tecnológico e evolução institucional, e (ii) analisar o processo de difusão do FS identificando restrições associadas a desenvolvimento de competências. Um approach baseou no processo de difusão permite participar na conversa nas formas de proteção e apropriação, que é associada à pergunta das motivações, e, ao mesmo tempo, contribui para entender o fenômeno de uma perspectiva evolutiva, caracterizado pela mudança, a incerteza, o aprende e a adaptação.

p. Resumen en inglés (hasta 1000 caracteres):

In recent years sociologists, anthropologists and economists has shown increasing the interest by the phenomenon of the free software (FS) or open source software (OSS). The FS, as opposed to the proprietary, is characterized by offering to users the freedom of using the software without restrictions, including the inspection, modification and redistribution of its source code without any payment of licenses; In this way, the fenomenon challenges ‘economics laws’ like the individual rationality objectified in the search of the utility maximization.

The general objectives of this investigation are (i) to understand the phenomenon of FS in a historic perspective, characterized by the technological, organizacional and institutional evolution, and (ii) to analyze the process of diffusion of the FS identifying associated restrictions to competences development. A aproach based on the diffusion process allows to participate in the discussion on the forms of protection and appropriation, which is associated to the question of the motivations, and, at the same time, contributes to understand the phenomenon from an evolutionary perspective, characterized by the change, the uncertainty, the learning and the adaptation.

rAprobado por (Apellidos y Nombres del Jurado):

Firma y aclaración de la firma del Presidente del Jurado:

Firma del autor de la tesis:

A Leandro, por darme buenas ideas.

Mis agradecimientos:

A los programadores y otros trabajadores informáticos que participaron en el trabajo de campo; a SADIO y en especial a Gabriel Baum, Hector Monteverde, Alfredo Pérez, Federico Heinz, Enrique Chaparro y Javier Blanco, por sus charlas y discusiones durante las JAIIO 2004; a Pablo Lavarello y Miguel Zanabria, por sus lecturas y comentarios de versiones preliminares de este trabajo; a todas aquellas personas que conciente o inconscientemente me ayudaron a que terminara esta tesis; y, especialmente, a Analía Erbes, José Borello, Dario Milesi, Sonia Roitter, Gabriel Yoguel y, en general, al grupo de investigación del Instituto de Industria de la Universidad Nacional de General Sarmiento, por darme un espacio para trabajar y enseñarme tantas cosas.

Especialmente, quiero agradecer a mi director, Gabriel Yoguel, por sus valiosos comentarios y por las fructíferas discusiones, pero sobre todo, por haber leído cada una de las versiones de esta tesis con el mismo entusiasmo de la primera vez.

ÍNDICE

1	Introducción	10
1.1	Problemática	10
1.2	Objetivos	13
1.3	Hipótesis	15
1.4	Justificación	16
1.5	Metodología	19
1.6	Secciones de la investigación.....	22
2	Qué es el software libre: evolución histórica y definiciones	25
2.1	Sendero evolutivo del SL y la informática	25
2.2	Definiciones	39
2.3	Otros tipos de software	45
2.4	Software libre vs. software propietario.....	47
2.4.1	El uso privado	47
2.4.2	El uso en la administración pública	49
2.4.3	El uso académico y científico	51
2.4.4	Algunos argumentos adicionales a favor de la promoción del SL.....	52
2.5	Síntesis y conclusiones	55
3	Algunas notas sobre la economía del software libre.....	57
3.1	A título de introducción o la economía digital ¿el fin de la era de la escasez?	57
3.2	Cambio organizacional e institucional en el nuevo contexto tecnológico.....	60
3.2.1	Cambio tecnológico y organizacional.....	61

3.2.2	Las bases antropológicas de la comunidad <i>hacker</i> base del cambio institucional.....	70
3.3	La cuestión de las motivaciones	76
3.3.1	La visión de la economía convencional.....	76
3.3.2	Algunas regularidades empíricas sobre la cuestión de las motivaciones..	80
3.3.3	Breve digresión sobre la hipótesis de racionalidad.....	83
3.3.4	SL, conocimiento y procesos de aprendizaje. El SL como un bien club ..	84
3.4	Síntesis y conclusiones	90
4	El proceso de difusión de SL en la Argentina. una aproximación teórica.....	92
4.1	El SL en los países en desarrollo: la relevancia del debate.....	92
4.2	Límites en la difusión de SL	93
4.3	Algunos modelos de difusión de tecnologías competitivas en presencia de economías de escala en la adopción.....	95
4.4	El concepto de percolación y adopción de tecnologías libres.....	100
4.5	Síntesis y conclusiones	104
5	Difusión de software libre en la Argentina. Límites y restricciones.....	105
5.1	Introducción: rasgos característicos.....	105
5.2	La difusión de SL entre programadores argentinos: un análisis descriptivo ..	105
5.3	Receptividad y conectividad	111
5.4	Relación entre las trayectorias tecnológicas, uso de redes y difusión de SL ..	117
5.5	Síntesis y conclusiones	120
	Bibliografía	123
	Anexos	129

1 INTRODUCCIÓN

1.1 Problemática

En los últimos años, luego de que su uso y desarrollo se generalizara, y de que se instalara la reflexión sobre el tema en ámbitos tecnológicos, ha surgido el interés de sociólogos, antropólogos y economistas por el fenómeno del software libre (SL) o software de fuente abierta (*free software* u *open source software*)¹. Este fenómeno aparece, particularmente ante la economía, como un hecho anómalo que desafía ‘leyes’ que esta disciplina considera centrales, entre ellas la racionalidad de los individuos objetivada en la búsqueda de la maximización de la utilidad.

Sin embargo, la popularidad que el SL alcanzó en el último tiempo², así como la participación de grandes empresas³ productoras de software y hardware en su desarrollo guiada estrictamente por la lógica del mercado, condujo a que muchos autores dejaran de lado su perplejidad y empezaran a intentar explicar y justificar qué posición adoptar frente a esta nueva forma de producción, distribución, y organización del trabajo en un área relativamente nueva como la informática. Asimismo, la notoriedad que adquirió el fenómeno abrió una discusión más general sobre las formas de apropiación -privada y

¹ A pesar de las importantes diferencias entre software libre y software de fuente abierta (ver sección 2), en este trabajo se utilizan las siglas SL (software libre) para hacer referencia indistintamente al software libre como al de fuente abierta. A los objetivos de la investigación dichas diferencias no son relevantes.

² Linux es el ejemplo de software libre más conocido y está desafiando la posición de Microsoft (MS) en el mercado de sistemas operativos (OS) para servidores. Actualmente Linux representa más del 25% de los servidores y la tasa de crecimiento es de 25% al año versus el 10% que logran los productos MS. Por otra parte, Apache representa el SL de mayor utilización, siendo líder en el mercado de servidores web con una participación superior al 60% (Dalle y Jullien, 2003).

³ Empresas como Oracle, IBM, Sun, Hewlett Packard tiene proyectos en los que usa y desarrollan SL y otras empresas como Red Hat, Suse y Mandrake, tiene modelos de negocios basados en la comercialización de distribuciones de Linux (Lerner y Tirol, 2000).

pública- de los beneficios del progreso tecnológico, la determinación de estándares, el espacio del dominio público y el rol de los derechos de propiedad intelectual (DPI).

El SL, a diferencia del propietario, se caracteriza por otorgar al usuario la libertad de usar el software sin restricciones, incluyendo la inspección, modificación y redistribución de su código fuente sin el pago de licencias. El SL vuelca voluntariamente al dominio público este conjunto de libertades porque no concibe al software como un producto terminado, desarrollado sobre la base de un diseño rígido por un grupo delimitado de personas que detentan (personalmente o a través de una organización) la propiedad de su producción intelectual, sino como un producto que evoluciona constantemente con nuevas contribuciones realizadas por un grupo amplio y abierto de desarrolladores que participan con distinto grado de involucramiento.

La cuestión sobre las motivaciones que llevan a la participación en desarrollos libres provocó numerosas discusiones e hipótesis centradas en la búsqueda de lucro, la búsqueda de prestigio y en una problemática más amplia referida al papel de los DPI en la nueva economía. A diferencia de lo que ocurre con las formas tradicionales de propiedad intelectual, donde el reconocimiento está relacionado con el lucro a través del usufructo de los DPI, en el SL, el prestigio lo otorga la aceptación de pares dentro de una comunidad, y el lucro está vinculado con el desarrollo de modelos de negocios basados en la provisión de servicios. Por otra parte, la discusión llevada a cabo en una esfera política o institucional se refiere a la utilidad y aplicabilidad de la normativa sobre derechos de propiedad intelectual y al crecientemente limitado espacio del dominio público en el actual paradigma tecno-productivo (Freeman y Perez, 1988) donde el conocimiento, la información y los activos intangibles ocupan un lugar cada vez más medular.

En el fenómeno de software libre pueden verse cambios tecnológicos, organizacionales e institucionales que, interrelacionados, caracterizan el pasaje de una economía basada en la producción industrial a una economía con creciente importancia de las actividades de servicios y basadas en conocimiento (Langlois, 2003). Para la difusión del SL fueron imprescindibles (i) avances drásticos en las tecnologías informacionales, tales como la aparición de Internet y la WEB (David 2005); (ii) cambios organizacionales

caracterizados por la organización de la producción por proyectos y en red con fuerte interacción en vínculos horizontales que favorecen los procesos de aprendizaje (Stolpper 1989, Lundvall 1999, Malerba y Orsenigo 2000, Nonaka y Takehuchi 1995, Casalet, Cimoli y Yoguel, 2005), y (iii) cambios institucionales asociados a la discusión sobre las formas de protección de la propiedad intelectual y el espacio de dominio público en una economía caracterizada por el rol del conocimiento y la información (Barlow, 1994).

Los distintos trabajos que reflexionan el fenómeno pueden ser agrupados en tres grandes categorías de acuerdo a si basan su análisis en las tres dimensiones planteadas. A pesar de que en muchos casos haya superposición de estas dimensiones de análisis, la siguiente agrupación está basada en el foco central de cada grupo de estudios. Los trabajos que analizan la dinámica de las comunidades, la organización en red y el impacto que tuvo sobre estas formas organizacionales el surgimiento de la comunicación electrónica y los estándares de comunicación se enmarcan en la discusión tanto sobre el cambio tecnológico como el organizacional (Benkler, 2002; Dalle y David 2003, Raymond 1999a). Los estudios enfocados en la comparación entre los dos modelos de desarrollo (libre/fuente abierta y propietario), en la discusión sobre el desarrollo de la ingeniería de software y formas más flexibles de organización y en el análisis de los efectos de los nuevos modelos sobre la calidad de los productos y sobre la construcción de competencias de los agentes, analizan fundamentalmente el cambio organizacional (Fowler, 2003; Benkler 2002; Garcia 2001, Raymond 1999a y b;). El conjunto de trabajos que discute las formas de distribución del software libre y sus formas de licenciamiento, el rol los derechos de propiedad intelectual en la nueva economía, la creciente limitación del dominio público, la base ética del encerramiento (*enclosure*) del conocimiento, entre otras cuestiones, va en el plano del cambio institucional (Zietlyn 2003; Dalle y Julien 2003, West 2003; Raymond 1999c y Stallman 1983 y 2002). Un último conjunto de artículos, enmarcados metodológica y teóricamente en la economía convencional⁴, busca entender el fenómeno como el resultado de la optimización de agentes racionales que

⁴ Por economía convencional se entiende la rama *mainstream* de las ciencias económicas, asociada frecuentemente al pensamiento neoclásico.

actuando individualmente, guiados por la búsqueda del beneficio personal, encuentran suficientes motivaciones para participar en desarrollos libres. Este último grupo no está asociado directamente con ninguna de las tres dimensiones arriba mencionadas, sino con realizar un esfuerzo por englobar dentro su marco teórico las conductas de estos agentes que *a priori* parecen irracionales⁵ y que entrarían en contradicción con las ‘leyes’ económicas (Lerner y Tirol, 2000; 2002; Baldwin y Clark, 2002; Hann et al, 2002; Varian, 2001). La pretensión de universalidad de la economía convencional anula la posibilidad de entender el fenómeno como el resultado de un proceso evolutivo e histórico que depende de un sendero previo, marcado por discontinuidades y acumulación, tanto en la esfera tecnológica como en la organizacional e institucional.

En gran parte de los trabajos, subyacen elementos normativos relacionados a distintos tipos de debates. En primer lugar, si el desarrollo y difusión del SL deberían ser promovidos por el Estado, ya que están asociados con creación de un espacio público y, por otra parte, significa una oportunidad para impulsar la industria de software en los países en desarrollo (PD) y reducir la dependencia tecnológica. En segundo lugar, y vinculado con la discusión sobre la difusión de estándares, la pregunta sobre cuál es el espacio para la imposición de estándares *de juri* y cuáles son los peligros de caer en efectos lock-in por la determinación de estándares *de facto* ineficientes o controlados por agentes privados en posiciones monopólicas. Finalmente, el debate sobre el rol de los derechos de propiedad intelectual en una economía caracterizada por la velocidad del cambio tecnológico, el conocimiento y el peso de los intangibles.

1.2 Objetivos

⁵ En cada hipótesis sobre las motivación para desarrollar SL subyace una forma de racionalidad. Para la teoría económica convencional los agentes son hiperracionales y toman sus decisiones en condiciones de información perfecta. Esta hipótesis de racionalidad instrumental conlleva a que los agentes estén dotados de conductas maximizadoras del bienestar o utilidad individual.

En el marco de esta problemática, el objetivo general de esta investigación es comprender el fenómeno de SL desde una perspectiva histórica caracterizada por la evolución tecnológica, organizacional e institucional. También forma parte del objetivo general de esta investigación analizar el proceso de difusión del SL e identificar sus restricciones. Un enfoque centrado en la difusión permite participar de la discusión sobre las formas de protección y apropiación, implica una hipótesis sobre las motivaciones y a la vez contribuye a comprender el fenómeno desde una perspectiva evolutiva caracterizada por el cambio, la incertidumbre, el aprendizaje y la adaptación. De esta investigación se desprenden recomendaciones de política relacionadas con las tres áreas identificadas más arriba (rol de Estado, determinación de estándares y derechos de propiedad intelectual).

Se busca conducir la investigación en el marco de los siguiente objetivos específicos:

- (i) Contribuir a la discusión sobre las motivaciones que llevan a la participación en desarrollos libres desde una perspectiva evolutiva y a partir de elementos de análisis tomados de la antropología, las ciencias cognitivas y la bibliografía sobre conocimiento desarrollada por la teoría evolucionista.
- (ii) Contribuir a la discusión sobre difusión y apropiación de conocimiento, en el marco de fuertes cambios institucionales que actualmente se debaten entre un endurecimiento de las formas de protección de la propiedad intelectual y el desarrollo de formas institucionales flexibles que pongan énfasis en eliminación de obstáculos a la innovación y al acceso a las fuentes de aprendizaje.
- (iii) Contribuir a la discusión sobre creación de competencias en el contexto de formas organizacionales caracterizadas por la utilización de herramientas de comunicación digital, la interacción en redes horizontales y la presencia de comunidades epistémicas o clubes definidos por el grado de desarrollo de capacidades cognitivas. Se busca interpretar la dinámica de desarrollo y difusión de SL desde el análisis de la circulación de conocimiento y la creación de competencias.

- (iv) Caracterizar el proceso de difusión de SL en una muestra de trabajadores informáticos argentinos, determinar sus restricciones sobre la base de un modelo que considere agentes heterogéneos en términos de capacidad de receptividad o adopción y de conectividad o vinculaciones.

Este trabajo parte de una serie de hipótesis sobre comportamiento, motivaciones y racionalidad, analiza los efectos de la difusión de SL sobre la creación de competencias y participa de la discusión sobre las formas de protección en el nuevo paradigma tecno-productivo.

1.3 Hipótesis

Las hipótesis centrales del trabajo están relacionadas con los objetivos específicos antes detallados:

H1. Las motivaciones para el desarrollo de software libre son tanto de tipo económico como no económico. Esta hipótesis está a mitad de camino entre dos extremos: la búsqueda del lucro y el altruismo. Las motivaciones están asociadas a reglas de comportamiento de una comunidad definida por el desarrollo de capacidades cognitivas (Yoguel et al 2005), habilidades específicas y pautas éticas comunes (Raymond 1999c). Los conceptos de comunidad epistémica (Cowan, et al 2000) y relaciones donáticas (Maus, 1960; Harris, 1995; Zeitlyn, 2003), así como la hipótesis de racionalidad comunicativa (Habermas 1988, Johnson y Lundvall, 1994) contribuyen a explicar el comportamiento de la comunidad de desarrolladores.

H2. La forma de apropiación a través de los derechos de propiedad intelectual en su definición tradicional pierde vigencia en el área del software (Barlow 1994; Raymond, 1999, Stallman 2002, Lessig 2001) para ser reemplazada por mecanismos de protección que no impongan ritmos a la actividad innovativa y que permitan maximizar los aprendizajes que surgen de la interacción y una apropiación colectiva.

Las formas de protección laxas o el *disclosure* del código no implican que el SL sea un bien público. Por el contrario se trata de un bien club, donde los integrantes del club pueden hacer uso económico del código libre en modelos de negocios basados en servicios y sustentados en capacidades cognitivas diferenciales y competencias específicas que les garanticen la pertenencia al club.

H3. Los cambios organizacionales surgen cuando se deja de considerar al software como un producto terminado y se comprende su carácter evolutivo (Raymond 1999a, Stallman 2002, Benkler, 2002; García 2001). La industria avanza hacia modelos de desarrollo más flexibles, opuestos a la tradicional ingeniería de software que se ha mostrado como una adaptación forzada de las ingenierías duras al área de desarrollo de software. Los modelos más flexibles ('metodologías ágiles'), la gestión de proyectos en red y el creciente protagonismo de los usuarios en el proceso de desarrollo (Fowler, 2003; Cockburn, 1999), provocan una fuerte descentralización del desarrollo y diseño que incrementa la diversidad y el número de innovaciones. Estos cambios organizacionales están estrechamente vinculados con los cambios institucionales contemplados en la H2.

H4. Datos: (i) la característica de bien club del software libre, (ii) la relevancia de los procesos de aprendizaje a través de interacciones y (iii) las barreras a la entrada definidas por capacidades diferenciales y por costo de migración de una tecnología a otra, la adopción de software libre en ambientes técnicos (desarrolladores) depende fuertemente de la receptividad y conectividad de los agentes. Este proceso de difusión puede ser analizado a través del concepto de percolación (Frish y Hammersley 1963, David y Foray 1994, Antonelli 1997). La difusión de SL facilita el desarrollo de competencias tecnológicas. Incrementar su difusión permitiría impulsar el sector de software a nivel local y reducir la dependencia tecnología en software. No obstante las competencias y vinculaciones de los desarrolladores (Borello et al 2005) actúan como restricciones en la difusión del mismo.

1.4 Justificación

La problemática del SL como elemento distorsivo del paradigma dominante en la producción y distribución del software está siendo abordada desde múltiples perspectivas. La bibliografía sobre SL suelen extenderse explicando la importancia de abordar esta área con argumentos que van desde las dificultades de las herramientas que provee la teoría para dar explicación al fenómeno, hasta la descripción de las posibles ventajas asociadas a una decisión correcta y oportuna sobre el lugar que debería ocupar el SL dentro de la industria informática. De tal forma, las preguntas centrales de gran parte de los trabajos giran entorno a si el SL va a perdurar o si se trata de una moda pasajera, si la competencia entre ambos paradigmas va a significar la muerte de uno o la convivencia en un delicado equilibrio, y particularmente qué hacer. Qué hacer desde la política pública, qué hacer desde los negocios, qué hacer desde las instituciones académicas y de investigación.

Esta investigación presenta elementos originales en cuatro áreas diferentes: en el enfoque aplicado sobre la problemática, en el análisis desde la especificidad del sector SSI en la Argentina, en los datos y en la aplicación modelo teórico de adopción de tecnologías.

En primer lugar, el enfoque aplicado sobre la problemática resulta novedoso ya que plantea cuestiones no discutidas por la bibliografía de SL desde la economía. Como ya hemos comentado, los enfoques tradicionales sobre el SL apuntan a explicar las motivaciones y a confrontar los dos modelos de desarrollo (abierto y cerrado) exponiendo ventajas y desventajas en cada caso. Una perspectiva que integre el cambio tecnológico, con el organizacional e institucional y que enfatice el desarrollo de capacidades aún no ha sido profundamente explorada por bibliografía. Esta perspectiva resulta central porque: (i) resulta coherente con algunos datos empíricos recolectados que afirman que los motivos de participación en del desarrollo y uso de SL está fuertemente ligado con el compartir y brindar conocimiento (FLOSS 2002; BCG, 2001); (ii) permite explicar el fenómeno desde una perspectiva sistémica que recoja las motivaciones tanto económicas como no económicas en el marco de comunidades epistémicas y aprendizaje interactivo dando una lectura más compleja del fenómeno que la maximización de la utilidad individual; (iii) da elementos para entender a la innovación como un fenómeno colectivo y cooperativo donde las redes y las interacciones personales e institucionales son importantes y caracterizan las nuevas formas organizacionales que se complementan con formas institucionales de protección flexibles de protección de la propiedad; (iv)

posibilita pensar la importancia de la circulación del conocimiento para su creación y la generación de capacidades que posibiliten una menor dependencia tecnológica en software y limiten los efectos *lock-in*.

En segundo lugar, actualmente no hay trabajos que aborden esta problemática desde la realidad del sector SSI en la Argentina. Los trabajos más recientes sobre el sector no desarrollan específicamente la problemática del SL ni tampoco toman posición al respecto haciendo recomendaciones de política sobre el tratamiento que debería recibir el SL. Por otra parte, las organizaciones promotoras del movimiento en la Argentina han escrito documentos que describen la situación y recomiendan su uso y políticas de Estado al respecto. Sin embargo su análisis es parcial, ya que se trata de agentes involucrados directamente. Los varios proyectos de leyes (en los diferentes niveles del Estado, municipal, provincial y nacional) y documentos anexos sobre la promoción del SL, y particularmente, su uso en el Estado, constituyen un aporte central a la mirada del SL desde la realidad local, sin embargo también se trata de una mirada parcial.

En tercer lugar, la parte empírica del trabajo se basa en los datos originales recolectados a través de una encuesta electrónica a trabajadores informáticos argentinos con información sobre las vinculaciones y capacidades de los trabajadores (usos de herramientas y actividades que realizan) y uso de tecnologías libres, entre otras cuestiones (ver anexo metodológico).

Por último, el trabajo también es original porque plantea la aplicación de un modelo que permite explicar el grado de adopción en tecnologías competitivas sobre la base de información sobre las capacidades y relaciones de los agentes. Los modelos más comunes de difusión de tecnologías competitivas no incorporan información sobre los agentes sino que se limitan a la descripción de senderos de adopción de la tecnología sobre la base de sponsoreo o grado de adopción, poniendo el énfasis en las economías de red más que en las características de los agentes y su entorno. El modelo que propone este trabajo (ya aplicado en algunos artículos sobre difusión de tecnologías competitivas en presencia de economías de red) resulta novedoso en la aplicación al caso de la difusión del SL. Por otra parte, las recomendaciones de políticas que surgen de este análisis son novedosas comparadas de las recomendaciones habituales sobre el tratamiento del SL.

1.5 Metodología

La economía convencional ha trabajado largamente sobre una base epistemológica centrada en el individualismo metodológico. Esta noción, adoptada tanto por la economía como otras ramas de las ciencias sociales, puede ser rastreada en el campo de las ciencias exactas y físicas. Las bases del individualismo metodológico pueden hallarse en la concepción newtoniana del estudio de la materia que se concentra en su unidad mínima. De tal forma, observando el comportamiento de la materia en su escala más pequeña, el átomo, se pueden hallar las claves comprender el funcionamiento de los astros. La aplicación de esta noción a la economía implica que la sociedad pueda ser desmembrada como la materia hasta hallar una unidad básica: el individuo representativo, dotado de una conducta capaz de realizar elecciones racionales. A partir de la agregación de estas conductas es posible comprender el funcionamiento del todo.

Esta opción metodológica le permitió a la economía la aplicación de modelos matemáticos y el acercamiento, más que ninguna otra ciencia social, al instrumental de las ciencias más duras, en especial a la física newtoniana. A su vez, con esto logró ganar reputación entre la comunidad científica por la supuesta objetividad de su método, al tiempo que sus conclusiones ganaban en universalidad y atemporalidad, mientras el resto de las ciencias sociales tardaron en aplicar idea de elección racional⁶. Sin embargo, adoptar esta metodología le costó a la economía un alejamiento cada vez mayor de su objeto de estudio, intrínsecamente, institucional, histórico y social. El individualismo metodológico despojó a la economía de su carácter social, al situar a las elecciones individuales en un ambiente vacío de instituciones, grupos de intereses y conflicto. La

⁶ Según Dahl y Lindblom “en el pasado, muchos científicos sociales, no economistas, han sido desafiados por el rigor abstracto y lógico de la teoría económica”; “... nos gustaría pensar que el proceso por medio del cual las teorías económicas se ha apropiado del consenso de su propia comunidad, pudiera extenderse y hacerse más accesible para los no economistas...” De hecho, en los últimos años *the public choice* (la elección pública) de Buchanan ha realizado la contribución más importante al respecto.

construcción teórica implicó para los economistas la adopción de supuestos simplificadores que alejados de la realidad conducían a una disociación cada vez mayor de la teoría y la empiria.

No obstante, y a pesar del predominio de una visión central, dentro de las ciencias económicas conviven diversas corrientes y escuelas contrapuestas, incluso desde lo metodológico. Algunas de estas corrientes complejizaron la herramienta metodológica convencional incorporando restricciones institucionales, costos de transacción, riesgos e incertidumbre, racionalidad limitada y problemas de información, como por ejemplo el neoinstitucionalismo, pero mantuvieron los elementos básicos del individualismo metodológico (Ayala Espino 1999). Por otra parte, el enfoque sistémico se ha constituido en la opción opuesta al individualismo⁷. La perspectiva sistémica trata de comprender el funcionamiento de la sociedad desde una perspectiva holística e integradora, en donde predomine el análisis de las relaciones entre las partes y el todo más que el estudio específico del comportamiento de las partes. El enfoque sistémico no concibe la posibilidad de explicar un elemento si no es en su relación con el todo. Desde esa perspectiva aborda el estudio de la realidad como un todo, considerando sus componentes y los distintos estados de interacción entre ellos. Para ello establece roles, relaciones, normas e instituciones que articulen a las partes y gobiernen sus acciones, es decir, den sentido al funcionamiento del conjunto.

El biólogo austriaco Ludwig von Bertalanffy a mediados del siglo veinte expuso en su *Teoría General de los Sistemas* los elementos fundamentales de este enfoque metodológico. Los modelos sistémicos partieron por explicar los sistemas biológicos sobre la base de la premisa de que la complejidad de los mismos y sus relaciones era inabordable desde la relación lineal causal. Percibían que la complejidad de las estructuras organizadas jerárquicamente que se hallaba en el universo debía ser abordada mediante leyes sistémicas. Este enfoque fue adoptado por diferentes corrientes heterodoxas del pensamiento económico, entre ellas el evolucionismo, que comparte con

⁷ A pesar de que según algunos autores proponen que ambos enfoques pueden ser conciliables o complementarios (Covarrubias 2002).

el enfoque sistémico cierta herencia de los sistemas biológicos y del evolucionismo darwiniano y lamarckiano (Von Berthalanffy, 1987).

El evolucionismo económico comparte principalmente con el darwiniano el método de selección y la noción de irreversibilidad (o *path dependence*). Por el contrario, la dinámica que describe la economía convencional está inspirada fuertemente en los sistemas mecánicos que otorgan un rol central a la explicación del orden y del equilibrio (Nelson y Winter 1982, Dosi et al 1988). Los sistemas dinámicos que aplica la economía convencional son eminentemente ahistóricos y cerrados. A partir de las condiciones iniciales se puede predecir perfectamente la trayectoria que tomará la variable y pequeños cambios en las condiciones iniciales no implican cambios importantes en los senderos hacia el equilibrio. En el caso de los modelos históricos, no es posible hacer predicciones. El modelo es abierto y cambios infinitesimales en las variables exógenas pueden conducir a cambios significativos de las trayectorias y de la situación final. En este tipo de modelos el camino condiciona a la situación final. a diferencia de los modelos de la economía convencional donde lo relevante es el equilibrio final más allá de la trayectoria (Dosi y Metcalfe 2003).

Por otra parte, el evolucionismo económico se aleja de las ideas de Darwin en que no supone que mutaciones aleatorias o el criterio de supervivencia del más apto, sino que rescata la noción de aprendizaje, desechada en la evolución biológica darwiniana donde la herencia es genética y no adquirida. En este último sentido, el evolucionismo económico es lamarckiano más que darwiniano.

La presente investigación comparte elementos de la economía evolucionista como por ejemplo la necesidad de un acercamiento sistémico a la realidad y busca una integración con un enfoque del tipo histórico que permita comprender el objeto de estudio en el marco de los cambios tecnológicos, organizacionales e institucionales que caracterizan la evolución del capitalismo (Langlois, 2003).

La discusión de las tres primeras hipótesis se basa en el análisis de una serie de documentos (trabajos sobre la historia de la informática, y referidos a la problemática del SL, autorreflexiones sobre el fenómeno SL, manifiestos y otras publicaciones de las organizaciones que nuclea al club de desarrolladores, etc) recopilados a través de la web

y de información obtenida a partir de entrevistas no estructuradas a personas que participan de los movimientos de promoción y difusión de SL. Con esta información y los elementos teóricos que propone la bibliografía sobre conocimiento, redes de innovación y creación de competencias se trabajará en un plano teórico esgrimiendo argumentos a favor de las hipótesis a testear. La discusión de estas tres primeras hipótesis está articulada en los capítulos dos y tres.

Los capítulos cuatro y cinco de esta investigación se abocan al testeo de la cuarta hipótesis. El concepto de percolación permite explicar las restricciones del proceso de difusión de tecnologías libres. En el análisis de la información empírica se utiliza una técnica de análisis factorial para describir las trayectorias tecnológicas de los programadores y diversos test paramétricos y no paramétricos para testear el grado de asociación entre las variables consideradas. En esta investigación se analiza el proceso de difusión entre usuarios expertos; es decir, entre quienes utilizan el SL sobre la base de sus propias capacidades, aquellos que se apropian del código fuente disponible en Internet para su uso personal o para desarrollar negocios sobre esa base. Quedan fuera del alcance de este trabajo, el estudio de la difusión del SL entre usuarios finales, por ejemplo los que *compran* servicios sobre SL a empresas del tipo *Red Hat*. Esto permite simplificar el análisis, y deja preguntas para futuras investigaciones; en un principio, puede preverse que solamente podrá existir una mayor difusión de SL en ambientes no técnicos en la medida en que prosperen empresas informáticas con modelos de negocios basados en la provisión de servicios sobre SL⁸.

1.6 Secciones de la investigación

El capítulo dos está dedicado a definir más precisamente la problemática de esta investigación. Para ello se comienza con un análisis de la historia del SL, en el marco del

⁸ Al mismo tiempo, quedan fuera del análisis las discusiones complejas como el costo total del uso de una tecnología, ya mencionado en el capítulo 2, sponsoreo de tecnologías, poder de mercado, red de proveedores, entre otras cosas.

sendero evolutivo de la actividad informática. Se hace especial énfasis en describir las diferentes fases por las que atravesó el fenómeno y la relevancia que fue adoptando en el área de los negocios y de la gestión pública. En segundo lugar, en este capítulo se da mayor precisión sobre las definiciones de SL y de *open source software* y se describen los argumentos de los principales referentes de cada movimiento, marcando diferencias y similitudes. Después de comentar brevemente el plano de las distintas clases de software que existe entre los extremos libre y propietario, en cuarto lugar, se analiza desde la perspectiva privada, pública y académica, las ventajas de la difusión de SL y los principales argumentos a favor de que sea promovido públicamente.

En el capítulo tres se discute la aparición del SL en el contexto del nuevo paradigma tecno-productivo. En primer lugar, en este capítulo se busca identificar en el fenómeno del SL transformaciones interrelacionadas en las áreas institucional, organizacional y tecnológica. La nueva forma de organización de desarrollo de software que plantea el SL requiere de transformaciones institucionales en el marco de fuertes innovaciones tecnológicas que permiten estas nuevas formas de organización. En segundo lugar se discuten las hipótesis tradicionales sobre motivaciones y se plantea la necesidad de vincular el problema de desarrollo de competencias con la difusión de SL. De esta forma se intenta fundamentar la hipótesis de que el SL puede ser considerado un bien club.

El capítulo cuatro está dedicado al estudio de las restricciones de la difusión del SL en países en desarrollo. En primer lugar, se presenta brevemente una justificación del análisis del proceso de difusión de tecnologías abiertas en países en desarrollo; y, en segundo término, se comparan las distintas formas de estudiar el proceso de adopción y difusión de tecnologías competitivas en presencia de economías de red. Se pone especial énfasis en la aplicación del concepto de percolación que evalúa las posibilidades de adopción del SL sobre la base de cuatro variables: la capacidad de receptividad (capacidad de los agentes de absorber la tecnología), la conectividad (que considera los vínculos con otros agentes y cómo esto influye sobre su decisión de adopción), la densidad (número de agentes) y las fuerzas exógenas a favor de la difusión (por ejemplo, políticas públicas). Este concepto resulta atractivo porque plantea la existencia de economías de escala en la adopción y un proceso aleatorio asociado a la heterogeneidad de agentes (tanto en términos de receptividad como de conectividad).

En el capítulo cinco se realiza el ejercicio empírico que demuestra la asociación entre (i) grado de difusión de SL, (ii) desarrollo de capacidades en un marco de un sendero tecnológico específico (*proxi* de capacidad de receptividad) y (iii) desarrollo de redes y utilización de herramientas de comunicación electrónica (*proxi* de la conectividad). Este ejercicio confirma la relevancia de umbrales mínimos (i) de capacidades y competencias (desarrollo de capacidades cognitivas, habilidades específicas, pertenencia a una comunidad epistémica) y (ii) de vinculaciones (aprendizajes a partir de interacciones, racionalidad comunicativa) para la difusión de SL, lo que avala la hipótesis de que el SL responde más a la forma de bien club que de bien público. El capítulo concluye con recomendaciones de políticas que permitan una profundización en la difusión de SL y capitalización de las ventajas potenciales del mismo con el objetivo de desarrollar competencias y reducir la dependencia tecnológica.

2 QUÉ ES EL SOFTWARE LIBRE: EVOLUCIÓN HISTÓRICA Y DEFINICIONES

2.1 Sendero evolutivo del SL y la informática

En las primeras décadas posteriores al nacimiento de la informática, el software estuvo fuertemente ligado al hardware sobre el que se ejecutaba y era considerado un valor agregado a éste más que un producto en sí mismo. Las primeras grandes inversiones en desarrollo de software estuvieron a cargo de los productores de hardware donde éste iba a correr, como por ejemplo los Laboratorios Bell de AT&T, el Centro de Investigaciones de Xerox en Palo Alto, o IBM. La estrategia comercial de estas compañías pioneras radicaba en el establecimiento de plataformas propietarias, verticalmente integradas desde el hardware hasta el software, que incrementaban los costos de migración, elevaban las barreras a la entrada y les permitían operar en los mercados monopólicos de los productos asociados a la plataforma o paquete tecnológico.

Sin embargo, circunstancias más o menos fortuitas, sumadas a la inestabilidad inherente del modelo de plataforma propietaria vertical⁹ condujeron a que el control pasara del hardware al software. El éxito en practicar ingeniería reversa sobre la PC IBM y el fallo que otorgó legitimidad a la producción de clones PC, llevó a la desarticulación del oligopolio del hardware. La ruptura del oligopolio del hardware por una lado, reforzó la caída de los precios de las computadoras que ya se venía dando sobre la base de la

⁹ El modelo de múltiples plataformas verticalmente integradas difícilmente hubiera perdurado hasta nuestro días; los problemas de compatibilidad quizás hubieran conducido a estándares de facto o a una segmentación del mercado. Apple es un ejemplo de empresa que prefirió continuar con una estrategia de plataforma integrada. Sin embargo, tuvo que articularla con una estrategia de nichos para poder seguir siendo competitiva y desarrollar múltiples esfuerzos de compatibilización para resolver los problemas de estándares de facto que dominan la industria.

vertiginosidad del cambio tecnológico¹⁰. Por otro lado, produjo un importante impulso a la industria del software, que contó con un mercado más amplio que el que proponían los desarrollos específicos para un tipo determinado de computadoras.

Hacia mediados de los setenta, en Unix fructificaron los enormes esfuerzos por desarrollar un sistema operativo que corriera en múltiples plataformas. Este sistema operativo y el lenguaje C (para aplicaciones Unix), desarrollados por AT&T, fueron distribuidos e instalados en diferentes instituciones académicas gratuitamente o a costo nominal. Las distintas instituciones usuarias realizaron cambios e innovaciones sobre el código que fueron, a su vez, compartidos por otros (Lerner y Tirol, 2000). Una década más tarde, aparecieron en el mercado DOS y, más adelante, Windows como sistemas operativos independientes del hardware, aunque, para entonces, ya se habían desarrollado estándares para el hardware sobre la base de la PC IBM.

Con el surgimiento de estos OS multiplataforma y la estandarización de las computadoras personales, los productores de hardware perdieron el control del paquete tecnológico y con él el poder de mercado. El sistema operativo aisló al usuario de las particularidades del hardware, y los nuevos productos de software (aplicaciones) pasaron a *acoplarse* directamente sobre el sistema operativo, sin tener necesidad de comunicación directa con el hardware. Las empresas de software lograron, así, ocupar el lugar que antes detentaban los productores de hardware (West 2003).

En este contexto, las empresas tecnológicas empezaron a entender el software como un bien comercializable, y, en un tiempo record, Microsoft, el paradigma de empresa productora de software, pasó de ser un grupo de entusiastas universitarios a una de las empresas más exitosas de los últimos tiempos. Fue la imposibilidad de realizar ingeniería reversa sobre el software¹¹, reforzada por las formas institucionales que adoptaron la

¹⁰ El número de transistores por procesador se ha venido duplicando cada 18 meses en los últimos 20 años (ley de Moore) lo que provocó incrementos espectaculares en la velocidad de procesamiento de la información, y, consecuentemente, fuertes caídas en los precios en los procesadores.

¹¹ No es posible realizar ingeniería reversa sobre un programa de computación con el objetivo reproducirlo, es decir, con un objeto de imitación o copia. No obstante, se realiza –muy frecuentemente– ingeniería reversa sobre formatos y

comercialización y la protección de copia, la que permitió la formación de monopolios en el sector. El procesador de una computadora procesa instrucciones en forma binaria, es decir, como una secuencia de ceros y unos; no obstante, un programador escribe esas mismas instrucciones como código fuente en un lenguaje de programación que luego es ‘compilado’ (traducido a binario o código máquina) por un programa específico denominado compilador. Una vez que el código fuente es compilado no es posible interpretarlo y modificarlo, ni volver al código fuente original. Esto implica que no sea posible realizar ingeniería reversa sobre un programa de computación. Por otra parte, la comercialización de software tomó la modalidad de venta de licencias de uso de los programas y, como forma de protección, la del derecho de copia (*copyright*) sobre el código fuente¹². La comercialización del software excluyó la distribución del código, ya que no era necesario para que el programa funcionase, e impidió que este pudiera ser manipulado o modificado por los usuarios¹³. Este cerramiento (*enclosure*) de la tecnología fue extremadamente efectivo para demorar la entrada de imitadores que desarrollaran versiones mejoradas y productos similares, lo que favoreció el surgimiento de posiciones monopólicas.

La compañía AT&T no había podido lanzarse al mercado de software con el licenciamiento de Unix porque un viejo fallo *anti-trust* de 1956 en su contra se lo impedía. Así, durante quince años Unix estuvo confinado al uso interno y a la investigación académica en universidades y centros de investigación. Esta situación provocó un creciente intercambio de código fuente entre investigadores que dio un gran impulso al avance científico y tecnológico en informática. De ahí que frecuentemente se

protocolos, buscando la compatibilización de sistemas, y menos veces, sobre programas para hacer modificaciones menores, como quitar dispositivos de seguridad o control (*crackear* el software).

¹² La aplicación del *copyright* al caso del software toma algunas características específicas. Para obtener la protección legal sobre las producciones intelectuales es necesario depositar una copia del texto, partitura, imagen, etc a ser protegido; en el caso de los programas de computación están eximidos de este requisito de forma tal que no existe un depósito legal con el código fuente de, por ejemplo, Windows.

¹³ Para enfatizar esta protección la Digital Millenium Copyright Act de los Estados Unidos prohibió entre otras cosas, practicar ingeniería reversa sobre formatos (Chaparro, 2003). En la Argentina, el artículo 57 de la Ley 11.723 sobre el régimen legal de propiedad intelectual establece esto mismo.

asocie a los primeros desarrollos de software con el paradigma abierto o libre, a pesar de que todavía no existía la dicotomía entre sistema abierto y propietario. El proceso de compartir código se aceleró considerablemente con la difusión de Usenet, una red de computadoras que desde 1979 vinculaba a la comunidad de programadores de Unix (Lerner y Tirol, 2000) y que, junto con Apranet, serviría de base para Internet.

Sin embargo, más adelante, AT&T consiguió comercializar Unix masivamente a través de una empresa separada que fue vendida a Novell y luego a SCO (West 2003). El movimiento del software libre comenzó justamente como una reacción a la comercialización de Unix, a la política de AT&T de restringir el acceso de la comunidad académica a su código fuente y a la incipiente comercialización de software en general.

Habitualmente se considera el manifiesto escrito en 1983 por el entonces programador del prestigioso laboratorio de inteligencia artificial del MIT, Richard Stallman, como el punto de partida de todo el movimiento. Un año después de escribir el manifiesto Stallman inició el proyecto GNU¹⁴, que consistía en construir un sistema informático completo compatible con Unix pero que fuera libre, es decir que junto con los programas se distribuyera su código fuente y que éste pudiera ser modificado y redistribuido nuevamente. Para otorgar un marco de legalidad al software libre bajo el proyecto GNU, y garantizar que los desarrollos que se desprendieran de él siguieran siendo libres, Stallman redactó la *GNU General Public License* (GNU-GPL o simplemente GPL)¹⁵ y la aplicó a los programas del proyecto. Fundó también la *Free Software Foundation* (FSF), institución sin fines de lucro encargada de llevar adelante el proyecto GNU y de proveer un marco legal para el desarrollo de software libre.

Durante sus primeros años, el movimiento del SL no trascendió de los ambientes científicos y académicos; esto cambiaría con la llegada de Linux e Internet en la primera mitad de los 90s. En 1991 Linus Torvalds, un estudiante finlandés, comenzó a escribir un

¹⁴ GNU es un acrónimo recursivo para "GNU No es Unix" y se pronuncia fonéticamente (ñu). El logo del proyecto GNU es justamente la cabeza de un ñu.

¹⁵ Esta licencia, diseñada para SL, permite modificaciones en el código y obliga a que los productos que se desprendan de éstas permanezcan bajo la misma GPL (ver detalles más adelante).

kernel (núcleo del sistema operativo) compatible con los kernel Unix propietarios. Este kernel se llamó Linux y llenó el vacío más grande del proyecto GNU, dando origen al sistema conocido comúnmente como Linux o también GNU/Linux. Linux es sin duda la cara más conocida del software libre por el público general, y atrajo sobre sí la atención de los medios y de las grandes empresas que no había logrado la FSF. Luego, con la aparición de las "distribuciones"¹⁶ y de avanzados entornos gráficos de escritorio¹⁷, Linux terminó de salir del ámbito del *hobby* y la academia para competir fuertemente en el mercado de servidores y tímidamente en el de las computadoras personales. Por otra parte, la popularización del Internet permitió expandir a escala mundial el modo organizacional basado en los desarrollo colaborativos que ya habían sido experimentados en el marco de la FSF.

En 1997, Eric Raymond, en su influyente colección de ensayos ‘La catedral y el bazar’, analizó el modelo de desarrollo del SL y sus posibilidades a futuro dentro de la industria tecnológica desde una postura más pragmática que ideológica. Sus escritos criticaron la postura ‘muy fervorosa y anti-comercial de la cultura *hacker*¹⁸’ que rescataba Stallman y la FSF (Raymond, 1999b) y cambiaron el foco, al hablar de *open source software* más que de SL. El movimiento *open source* enfatizó en el efecto de la distribución del código fuente, dejando de lado la discusión de si los programas derivados debían permanecer libres o podían ser utilizados comercialmente¹⁹. La lectura de Raymond puede ser vista como un reflejo del avance que tuvo el SL en el ambiente de los negocios.

¹⁶ Versiones de GNU/Linux con multitud de aplicaciones empaquetadas de forma de facilitar su instalación y uso; las más conocidas Red Hat, SUSE, Mandrake y Debian.

¹⁷ Tales como Gnome y KDE, que incorporaban los sistemas de ventanas.

¹⁸ Cabe aclarar que a pesar de que hoy el apelativo hacker contenga un significado peyorativo y con él suela designarse a las personas que rompen sistemas de seguridad, el autor lo utilizar en el viejo sentido –ampliamente difundido dentro de los círculos más técnicos- de persona de gran capacidad intelectual y de alto reconocimiento entre sus pares. De aquí en más el término hacker será utilizado con este significado.

¹⁹ A pesar de las discrepancias, los trabajos de Raymond y de Stallman demuestran el carácter autoreflexivo del movimiento y dan cuenta de los esfuerzos por construir un sistema de normas basado en premisas éticas (referidas al *enclosure/disclosure* del conocimiento), en las que subyace la idea de que el avance del conocimiento requiere comunicación, interacción y combinación de conocimientos (ver capítulo 3).

En los últimos años el SL ha recibido el apoyo decidido de ‘pesos pesados’ en la IT²⁰, como IBM, Sun y HP y el surgimiento de jóvenes empresas dedicadas enteramente al software libre, como Red Hat o VA Linux. La participación de estos actores fue clave para que el fenómeno dejara de ser un experimento de un grupo idealista y entrara de lleno al mundo de los negocios. Así, quedó demostrado que los modelos de negocios basados en SL eran factibles y que incluso empresas dedicadas a la venta de licencias podrían beneficiarse de realizar contribuciones al desarrollo del SL.

Muchas de aquellas grandes compañías apoyaron el desarrollo de SL como una estrategia para limitar el área de influencia de MS. Estas empresas buscaron no negociar con MS que, desde su posición monopólica, les imponía contratos de exclusividad y condiciones abusivas. MS se había consolidado en el paradigma de empresa de software con una plataforma verticalmente integrada que ofrecía soluciones en todos los segmentos de negocios. Construyó poder de mercado sobre la base del éxito de su sistema operativo, a partir del cual apalancó el resto de sus productos y afianzó su posición sobre la base de estrategias de cerramiento en estándares propietarios que multiplicaron su efecto por las enormes economías de red y efectos *lock-in*²¹.

El resto de las grandes compañías tuvieron que concentrarse sólo en alguna área de negocios (Oracle en base de datos, por ejemplo) o en la producción de hardware y provisión de servicios (Sun, Hp e IBM). Con el surgimiento del SL vieron la oportunidad

²⁰ En este grupo de empresas se consideran las grandes empresas de IT con una trayectoria no menor de 15 años que compiten directa o indirectamente con MS y aspiran al desarrollo de una plataforma vertical propia. Otras empresas de software más jóvenes, sobre todo las nacidas luego del desarrollo de Internet -cuando el SL ya había ganado popularidad- (Google, Ebay, etc), mostraron una actitud más natural hacia SL y no se sintieron obligadas a tomar posición o redefinir su estrategia, quizás porque al no pasar por la transición simplemente lo tomaron como un dato de la realidad. Otro grupo de empresas, con un modelo de negocios con productos para aplicaciones específicas (Adobe, Norton, etc) no mostraron un interés de participar activamente en el SL más que, en el mejor de los casos, portando sus propios productos a plataformas abiertas (Linux).

²¹ David (1985) en su famoso artículo de sobre la economía del QWERTY demuestra que estándares ineficientes pueden sostenerse en el tiempo en presencia de economías de red. Arthur (1989), explica el rol de los accidentes históricos en la determinación de efectos *lock-in* y Katz y Shapiro (1968) analiza este fenómeno en el caso de tecnologías ‘esponsoradas’ (ver capítulo 4).

de no depender exclusivamente de la plataforma MS y de articular sobre software libre desarrollos propios que les permitieran revivir el anhelo de la plataforma integrada.

De las primeras 30 compañías de software del 2005²² más de la mitad se puede decir que tienen una participación activa ante el SL, ya sea porque participan en desarrollos libres o porque financian o soportan algún proyecto libre. Por otra parte, siete de estas 30 grandes compañías adoptaron una contribución ocasional, por ejemplo, realizando estudios sobre el desarrollo del movimiento y el impacto sobre su negocio o ya sea compatibilizando sus productos con las plataformas abiertas. El modelo de participación activa implica incluir dentro de la estrategia de negocios de mediano y largo plazo al SL, mientras que el de participación ocasional se refiere a una adaptación de la estrategia de la firma en la que reconoce el alcance del SL aunque no lo incluye dentro de su modelo de negocio. Se puede decir, entonces, que estas firmas consideran el impacto del SL sobre su negocio en un horizonte de corto plazo. Tan sólo 6 firmas no realizan actividades vinculadas con el SL y en algunos casos manifiestan una firme oposición al fenómeno. Puede asumirse que en estos casos se vinculan con expectativas de fracaso del modo organizacional del bazar.

Tabla 2.1 Modalidad de la participación en SL en 30 más grandes compañías productoras de Software y servicios informáticos

²² Según el ranking de las primeras 500 compañía de software de Sftwaremag.com.

Rank.	Empresa	Modalidad de la participación	Ejemplo de participación	Ventas (Mill. US\$)	Empleo
1	IBM	Activa	Desarrollo de productos e implementación y compatibilización con desarrollos propios. Intragación con el hardware. Apoyo financiero a diferentes proyectos libres. Apertura de código propietario	61.307	329.000
2	Microsoft	Sin participación	Oposición al movimiento	33.969	57.086
3	EDS	Ocasional	Sus productos corren sobre plataforma abiertas	20.669	117.000
4	Computer Sciences Corporation	Ocasional	Sus productos corren sobre plataforma abiertas	15.188	76.000
5	Accenture	Sin participación		15.114	100.000
6	Hewlett-Packard Company	Activa	Desarrollos. Apoyo financiero a diferentes proyectos libres. Integración con el hardware.	13.778	151.000
7	Oracle Corporation	Activa	Desarrollos. Apoyo financiero a diferentes proyectos libres. Compatibilización. Sus productos corren sobre SL	10.156	41.658
8	Hitachi, Ltd.	Activa	Desarrollos libres. Fuerte incidencia en desarrollo del movimiento en Japón y Asia	9.491	347.424
9	SAP	Activa	Desarrollos. Sus productos corren sobre plataforma abiertas	9.314	32.000
10	Capgemini	Activa	Desarrollos. Sus productos corren sobre plataforma abiertas	8.581	59.324
11	NTT Data Corporation	Activa	Participación en desarrollos libres. Apertura de código propietario.	8.015	17.389
12	Lockheed Martin	Activa	Participación en desarrollos libres.	7.651	130.000
13	Atos Origin	Activa	Participación en desarrollos libres.	7.232	46.584
14	Science Applications International	Sin participación		6.720	42.400
15	SYNNEX Corp.	Sin participación		5.314	3.000
16	Ingram Micro Inc.	Activa	Participación en desarrollos libres. Sus productos corren sobre plataforma abiertas	5.092	13.600
17	QUALCOMM Incorporated	Activa	Desarrollo de productos e implementación y compatibilización con desarrollos propios	4.880	7.600
18	Unisys Corporation	Activa	Desarrollo de productos e implementación y compatibilización con desarrollos propios.	4.725	36.400
19	EMC Corporation	Ocasional	Sus productos corren sobre plataforma abiertas. Ofrece servicios sobre plataformas abiertas	4.359	22.700
20	Affiliated Computer Systems	Activa		4.106	47.000
21	Sun Microsystems, Inc.	Activa	Desarrollos. Apoyo financiero a diferentes proyectos libres. Integración desde el hardware.	3.830	35.000
22	Fiserv, Inc.	Sin participación		3.730	22.000
23	CSK Corporation	Ocasional	Sus productos corren sobre plataformas	3.581	4.768

			abiertas		
24	Sungard Data Systems, Inc	Sin participación		3.556	13.000
25	Avaya, Inc.	Ocasional	Sus productos corren sobre plataforma abiertas. Ofrece servicios sobre SL.	3.499	14.900
26	Computer Associates International	Activa	Desarrollo de productos libres. Apertura de código propietario.	3.276	15.300
27	Getronics	Ocasional	Sus productos corren sobre plataforma abiertas	3.246	21.026
28	LogicaCMG	Activa	Desarrollo de productos e implementación y compatibilización con desarrollos propios	3.216	19.695
29	Google, Inc.	Activa	Participa en desarrollos libres. Para algunos autores su éxito se sustenta en el SL	3.189	3.024
30	Convergys Corporation	Ocasional	Sus productos corren sobre plataforma abiertas	2.488	66.300

Fuente: elaboración propia sobre base de Softwaremag.com. y otras fuentes. Datos correspondientes a 2004.

IBM dice haber invertido mil millones de dólares en Linux, además de haberse visto involucrada en más de una docena de proyectos libres relacionados con nuevas tecnologías como Java (de Sun), XML o Web Services. IBM se está especializando en la venta de productos y servicios compatibles con estas tecnologías, de esta forma su participación en desarrollos libres le permite incrementar la venta de sus propios productos (FLOSS, 2002).

Hewlett Packard, por su parte, centra su estrategia de *open source* en el desarrollo de Linux, que le permitió avanzar sobre el desarrollo de sistemas operativos. En el mercado de servidores, esta empresa vende hardware con versiones de Linux especialmente adaptadas (FLOSS, 2002).

Sun cuenta con una estrategia fuertemente comprometida con el *open source*, participa en una gran variedad de proyectos entre ellos, Open Office (una versión abierta de Star Office, el más fuerte competidor de la suite de oficina de MS), Gnome, Mozilla, Apache, entre otros. Las motivaciones de Sun son diversas; por un lado, a diferencia de HP, Sun cuenta con una versión propia Unix (Solaris) para su hardware, por lo que el esfuerzo no está puesto en el desarrollo y adaptación del sistema operativo en sí mismo sino en las interfaces de usuarios, como el entorno gráfico y las aplicaciones de oficina que sean compatibles con su OS. Por otra parte, Sun es propietario de Java, el lenguaje de computación de mayor crecimiento reciente, asociado a las aplicaciones web para empresas (FLOSS, 2002).

No es casual que MS sea una de las pocas organizaciones que no apoya el desarrollo de SL y se enfrente públicamente a él²³, ya que es una de las organizaciones que más amenazada se ve por su expansión. Ni tampoco que el movimiento del SL considere a la

²³ Se enfrenta públicamente al movimiento, a través litigios, por cuestiones de DPI, con poder de lobby, cuando es el Estado el que decide dar apoyo al SL y con propaganda ante el público en general, que descalifica la calidad del mismo. En una serie de documentos llamados los ‘documentos de Halloween’, aparecidos al final de Octubre de 1998, los ejecutivos de Microsoft afirmaron la intención de utilizar varios métodos para obstruir el desarrollo de software libre: específicamente, diseñando protocolos y formatos de archivos secretos, y patentando algoritmos y características de software.

compañía su máximo adversario²⁴; después de todo, el objetivo último del SL es desarrollar una plataforma propia, que es el proyecto GNU.

Las empresas más pequeñas que surgieron con modelos de negocios basados enteramente en SL demostraron que brindar servicios sobre software libre puede ser rentable a pesar de no tener ingresos por licencias²⁵. De acuerdo con el informe FLOSS (2002) los modelos de negocios basados en SL pueden ser agrupados en dos grandes categorías: (i) los distribuidores y revendedores de productos libres ya desarrollados o (ii) los proveedores de servicios sobre código abierto. En el primer grupo estaría Red Hat, Mandrake, etc; estas empresas están asociadas fundamentalmente a la venta de ‘distribuciones’ de Linux. Su actividad está focalizada en la optimización estas versiones, packaging y distribución en un mercado caracterizado por usuarios no expertos (PyMEs o usuarios hogareños²⁶). Que el código fuente de Linux esté libremente disponible en Internet, no implica que el precio de estas distribuciones sea cero. Un usuario final difícilmente esté en condiciones de bajar ese código, adaptarlo a sus necesidades, compilarlo e instalarlo en su casa, ya que son requeridas habilidades específicas. Naturalmente, el margen para fijar precio es limitado porque las barreras a la entrada son menores desde que el código está disponible. El informe FLOSS (2002) indica como factores críticos de éxito de estas compañías la distribución y el marketing. Cuando los programas libres que se venden no son de uso genérico, como OS, sino de uso específico,

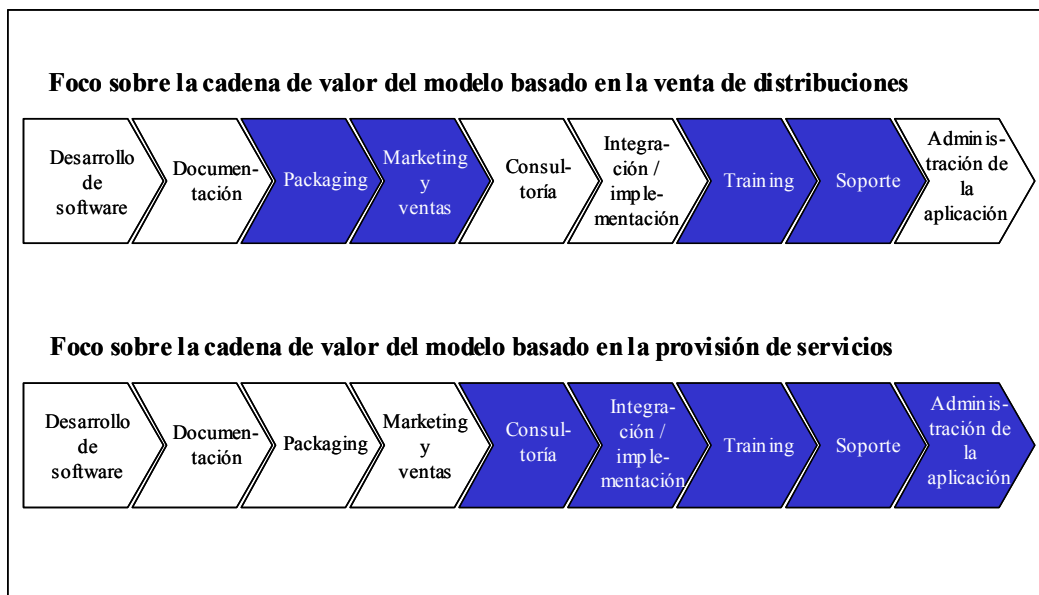
²⁴ En la FSF hay publicados una serie de documentos que hacen referencia directa a MS muestran la antipatía que despierta esta compañía en el movimiento SL: uno de ellos enumera posibles medidas a aplicar contra MS después del juicio antimonopolio referidas a estos mismo documentos de Halloween; otro, con estilo de solicitud, pide que no se envíen adjuntos con formato .doc por correo electrónico, y otro, titulado “¿Será Microsoft el Gran Satanás?” Explica que MS no es la única empresa que mantiene secretos los protocolos y formatos y que su oposición a MS se extiende al resto de las empresas que utilicen estas prácticas.

²⁵ Por ejemplo en diciembre de 1999 la compañía VA Linux entró en bolsa en el Nasdaq, el mercado público de las nuevas tecnologías con una subida record de 698%, pasando de un valor de 30 US\$ a 239,25 US\$ al final de la sesión (Moineau y Papatheodorou 2004).

²⁶ En la Argentina el mercado queda circunscrito al de las PyMEs que se ven presionadas por ‘Software Legal’, un organismo privado de control que presionan para que las organizaciones se deshagan de sus copias ilegales de Windows o paguen licencias sobre ellas.

como bases de datos (mySQL) o servidores WEB (Apache), es muy frecuente que el modelo de negocios combine la distribución con servicios asociados, ya que el objetivo no es penetrar en un mercado masivo de usuarios no expertos, sino en nichos más o menos amplios. El segundo grupo de modelos de negocios sería el que se concentra en el segmento final de la cadena de valor del software, incluyendo servicios como consultoría, integración e implementación de sistemas, training, soporte, mantenimiento, y administración de la aplicación (actualizaciones, etc).

Figura 2.1 La cadena de valor el software en distintos modelos de negocios



Fuente: Elaboración propia sobre base de FLOSS (2002)

El creciente interés y participación del sector público caracteriza la nueva etapa que atraviesa actualmente el SL. Las recomendaciones se refieren a dos áreas de acción: por una parte, la recomendación del uso de SL en el sector público argumentando: el libre acceso del ciudadano a la información pública, la perennidad de los datos públicos y seguridad del Estado y de los ciudadanos (ver apartado 2.4.2). Por otra parte, la promoción del SL como política de Estado, en tanto genera externalidades vinculadas con la determinación de estándares, el desarrollo de capacidades, la construcción de un espacio cuasi público y las posibilidades de reducir la dependencia tecnológica en software (ver apartado 2.4.4).

Recientemente desde el sector público en diferentes niveles de gobierno se está promoviendo el uso de SL. El ejemplo más llamativo es, quizás el del gobierno Chino²⁷, donde fueron creadas diversas empresas estatales con modelos basados en SL que buscan extender su uso tanto en áreas públicas como privadas. También se han debatido proyectos por impulsar el SL en países como la India, Brasil, Venezuela y Perú entre otros²⁸. También en la Argentina se han debatido tanto en el ámbito nacional (a partir de las discusiones del Foro de responsables de sistemas en las oficinas públicas nacionales), como en el caso de algunas provincias y municipios²⁹. Por otra parte, de acuerdo con IDA (2001), Bélgica, Francia Alemania, Italia, España, Suecia y la Comisión Europea cuentan con instrumentos para la promoción del SL.

²⁷ Un comité denominado Northeast Asia Open Source Software Promotion Forum, con representantes de China, Corea del Sur y Japón busca promover programas de uso del software libre en la administración y entre las empresas privadas en los países de noreste asiático. Un joint venture de tres empresas (la empresa estatal china, Red Flag, la japonesa Miracle Linux Corporation y la de Corea del Sur Haansoft), desarrolló Asianux 2.0, una versión de linux compatible con sus respectivos idiomas.

²⁸ El debate entre congresista peruano Villanueva Núñez y MS Perú por la ley de uso de SL en oficinas es reconocido como uno de los documentos donde mejor se exponen los argumentos tanto a favor como en contra del uso de SL en el estado.

²⁹ Tres proyectos de ley a nivel Nacional: 5613-D-00 diputado Dragan; 904-D-02 diputados Dragan, Becerra y Bertone, y 1280-D-04 diputado Cantin. Y once proyectos provinciales y municipales: Provincia de Buenos Aires (Proyecto de Declaración F-210/02-03 senador Conde, Proyecto de ley E-135/02-03 Senador Conde). Ciudad de La Plata (Proyecto de ordenanza 37202 Frangul, Ariccia y Viñes). Ciudad de Buenos Aires (Proyectos de leyes 1416-D-02 Caram, 1499-03 Maiorkevich, y 2801-04 Schifrin & otros). Municipio de Porteña de la prov. de Córdoba (Ordenanza 1275/2004 Peretti, Rossotto y Sadone). Provincia de Jujuy (Proyecto de ley 207-D-2002 Fernandes). Provincia de Misiones (Proyecto de ley D-22034/03 Sosa y Kornuta). Provincia de Santa Fe (Proyecto de Ley 11134-BRA / 771 CD D'Ambrosio y Zogbi). Ciudad de Santa Fe (Proyecto de ordenanza 25495-O-04 Simoniello, Henn, Giménez & Piazza).

Tabla 2.2 Cronología del SL

Año	Evento
1950s y 1960s	El código fuente es distribuido sin restricciones en grupos de usuarios de IBM y DEC
1969	Ken Thompson y Dennis Ritchie escriben las primeras versiones de Unix y el lenguaje C. Su código fuente es distribuido libremente durante los 70. Nacimiento de Arpanet.
1979	Después del anuncio de AT&T de comercializar Unix, la Universidad de Berkeley comienza el desarrollo de su propia versión de Unix, BSD (<i>Berkeley Software Distribution</i>). Nacimiento de Usenet.
1983	Stallman publica el manifiesto GNU sobre el software libre y establece la <i>Free Software Foundation</i> .
1991	Linus Torvalds publica la versión 0.02 de una nueva variante de UNIX llamada Linux en Internet.
1993	Se lanza la versión 1.0 de FreeBSD, un OS basado en Unix BSD.
1994	Marc Ewing forma Red Hat Linux y rápidamente se convierte en el distribuidor líder de Linux.
1995	El Apache Group escribe un nuevo Web server, Apache, basado en HTTPd 1.3 del National Center for Supercomputing Applications' (NCSA's). Este servidor es líder en el mercado actualmente.
1997	Eric Raymond escribe su colección de ensayos 'La catedral y el bazar'
1998	Netscape libera el Communicator 5.0 (Mozilla) abriendo su código fuente.
1999	La empresa VA Linux entra al Nasdaq con un alza record de 698% en un día.
2000	Más empresas de software, como Novell, Real, SAP lanzan versiones de sus productos que corren sobre Linux, avanzando así hacia modelos multiplataformas.
2001-2002	Se lanzan una serie de estudios sobre los beneficios de utilización de SL en el sector público (IDA, FLOSS). En Argentina el primer proyecto data del 2000.
2003-2004	Linux es compatible con varios idiomas asiáticos y se constituye el <i>Northeast Asia Open Source Software Promotion Forum</i> , para la promoción de SL en el noreste asiático.

Fuente: Elaboración propia sobre la base de fuentes diversas.

2.2 Definiciones

En el corazón de la definición de SL está la facultad que se le otorga al usuario de usar sin restricciones el software, y de inspeccionar y modificar su código fuente. Estos derechos se especifican en la licencia, que se establece como un contrato entre el tenedor de los derechos de propiedad intelectual y el usuario del software, y pueden ser más o menos permisivos. Por lo tanto, cada licencia de software libre o de fuente abierta implica una definición sobre qué se considera como SL o de fuente abierta, y esto último está inevitablemente asociado a posiciones políticas y filosóficas de los movimientos que promueven su desarrollo y difusión.

Pueden identificarse dos corrientes relativamente enfrentadas dentro de esta comunidad. Una es la del "software libre", que prioriza sobre todo el aspecto ideológico del código abierto, cuyo ideal es cubrir todas las necesidades informáticas de individuos y empresas sin necesidad de software cerrado o propietario. El referente más claro de esta postura es el propio Richard Stallman. La otra es la del "*open source*", centrada más en los beneficios tecnológicos y económicos del código abierto, generalmente identificada con Eric Raymond. La segunda postura es obviamente la que más agrada en el mundo de los negocios y la apoyan las empresas que utilizan SL pero tienen un modelo de negocios basado en el software propietario e incluso ventas de licencias³⁰.

La Free Software Foundation prefiere hablar de SL antes que de software de fuente abierta. Adopta una definición fuertemente asociada al concepto de libertad -aclarando la ambigüedad a la que se presta la palabra inglesa *free* que puede significar libertad o gratuidad³¹. Para esta corriente un programa es libre si los usuarios gozan de cada una de estas cuatro libertades básicas:

³⁰ Raymond 1999b distingue la postura *open source* de la del SL caracterizando primera pragmática y la segunda de fervorosamente anticomercial.

³¹ Dada la confusión que puede acusar en inglés el término *free*, la FSF aclara que éste debe ser entendido como *free speech* (uno de los baluartes más preciados de la cultura americana) y no como *free beer*.

- i Libertad de usar un programa con cualquier propósito (libertad 0).
- ii Libertad de estudiar cómo funciona un programa y adaptarlo (libertad 1), lo que implica el libre acceso al código fuente.
- iii Libertad de distribuir copias (libertad 2).
- iv Libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente también es un requisito previo para esto (www.fsf.org, 2005).

La importancia de la libertad para esta corriente está esta no solo en el acceso al código fuente sino en la libertad para modificarlo y distribuirlo y que las sucesivas versiones del programa mantengan estas libertades. El desarrollo de software libre implica siempre futuros desarrollos libres, cualidad, que hace que se reconozca a la licencia GPL como una licencia ‘viral’ (porque contagia a los desarrollos que surjan de código GPL). Este diseño de licencia está alineado con propósito fundacional de la FSF de construir una plataforma completamente libre. En este sentido, el SL compite directamente con el propietario; aspira a proporcionar software libre para hacer todos los trabajos que los usuarios de computadoras quieran hacer y por lo tanto hacer que el software propietario se vuelva obsoleto (www.fsf.org, 2005; www.gnu.org, 2005). La competencia entre SL y propietario, desde la perspectiva de la FSF se desenvuelve en tres diferentes niveles. La más evidente es la competencia por la cuota de mercado, por ejemplo, cuál es el *market share* de Apache en el mercado de los servidores web. La segunda es una competencia organizacional: ¿es el modo organizacional del SL es más eficiente que el propietario para desarrollar programas de calidad (ver capítulo 3)? Y la tercera es una competencia ideológica, que puede entenderse como una reacción ante el endurecimiento de los DPI en la economía digitalizada.

Por su parte, el movimiento que lidera Eric Raymond define como software *open source* no simplemente a aquel de libre acceso al código fuente sino que considera los siguientes criterios específicos:

- i Libre redistribución. Cualquier programa *open source* puede ser redistribuido sin necesidad de pagar royalties.
- ii Código fuente abierto. Efectivamente la distribución del programa debe incluir el código del mismo.
- iii Libre modificación. El código puede ser modificado y las modificaciones pueden distribuirse bajo las mismas condiciones de la licencia original.
- iv Integridad del autor. La licencia puede no permitir las distribuciones con modificaciones sino con *patches* u obligar a que los cambios sean firmados o presentados como versiones diferentes al trabajo original.
- v No discriminación contra grupos o personas.
- vi No discriminación de campos de aplicación. La licencia no debe restringir su uso a un campo específico, por ejemplo, no puede restringir su uso en el ámbito de los negocios, o académico.
- vii Distribución de la licencia. Los derechos concedidos deben ser aplicados a todas las personas a quienes se redistribuya el programa, sin necesidad de obtener una licencia adicional.
- viii La licencia no puede ser específica de un producto, sino del programa en cualquiera de sus versiones.
- ix La licencia no puede contaminar a otro software, es decir, las derivaciones de programas de fuente abierta no están obligados a ser de fuente abierta.
- x La licencia tiene que ser tecnológicamente neutral. (www.opensource.org, 2005)

La especificidad del *open source* está en el criterio número ix, ya que no obliga como el SL a que las modificaciones mantengan el mismo tipo de licencia y por este motivo es que las empresas están más cercanas a esta posición que a la primera, que se presenta

como mucho más radical. La postura del *open source* se caracteriza por un mayor pragmatismo.

Eric Raymond (1999b) afirma que si bien el lado más pragmático de la cultura *hacker*³² ha estado presente desde los inicios, no obstante “solo desde la explosión de Linux a principio de los años 1993-1994 halló su verdadera base de poder”. De acuerdo con este autor, “aunque Linus Torvalds nunca se opuso a Stallman, puso el ejemplo mirando benignamente el crecimiento comercial de Linux, apoyando públicamente el uso de software comercial de alta calidad para tareas específicas, y burlándose gentilmente de los más puros y fanáticos elementos de la cultura de código abierto”. Un efecto del rápido crecimiento de Linux fue el surgimiento de una nueva generación de *hackers* para los cuales Linus era la verdadera fuente de inspiración mientras que la FSF era primariamente de interés histórico.

La visión del *open source* no toma una posición de confrontación frente al software propietario: asume que son modelos centralmente complementarios. La competencia será en el plano organizacional. El movimiento *open source* no busca que el futuro todo el software sea libre ni cuestiona que algunas empresas tengan modelos de negocios basados en la venta de licencias. De acuerdo con esta corriente una licencia de código abierto debe proteger “el derecho incondicional de cualquier parte de modificar y distribuir versiones modificadas de software de código abierto”³³. Sobre este derecho se basa el principio organizacional básico del software libre o de fuente abierta que es la contribución descentralizada en desarrollos colaborativos.

Las dos licencias más utilizadas en el mundo resultan ser exponentes de cada una de estas corrientes. La licencia más difundida es la GPL, asociada al movimiento SL. La segunda en importancia, asociada al movimiento “open source” es la BSD (Berkeley Software

³² Dentro del ambiente del SL la palabra *hacker* no tiene la connotación negativa asociada a actividades delictivas (*cracker*) que se le da frecuentemente. Un *hacker* en este contexto es un programador habilidoso; de acuerdo con Raymond (1999c), la definición de *hacker* está asociada a la afición por lo técnico y al placer de resolver problemas sobrepasando los límites.

³³ *Open Source Definition* de la *Debian Free Software Guidelines* (1997).

Distribution) desarrollada por la universidad de Berkeley para licenciar su propia versión de Unix. La licencia BSD permite a terceros no sólo la modificación y redistribución, sino también la modificación y redistribución en forma de software propietario (algo no permitido por la GPL). Existe, además un conjunto numeroso de licencias con menor peso de importancia que fueron desarrolladas en el marco de proyectos específicos, que pueden ser alineados a alguno de estos dos modelos, de igual forma, hay diferentes versiones tanto de la GPL como de la BSD³⁴.

En el marco de esta investigación no parece relevante analizar las características específicas de cada caso, pero sí enfatizar la existencia de una enorme diversidad de formas de protección ‘libres’, así como la trascendencia que han tenido estas formas de protección a otros ámbitos, tanto científicos como culturales. Este es el caso del movimiento de *open science* que se opone al control de las publicaciones científicas (*Science, Nature, Scientific American*, etc) sobre sus contenidos, desarrollados por investigadores que se ven obligados a resignar derechos sobre sus trabajos para poder ser aceptados en estos prestigiosos *journals*. Otro caso es el desarrollo de las licencias *Creative Commons* y *Science Commons*, diseñadas para ser aplicadas sobre diversos tipo de producción intelectual (www.creativecommons.org, 2005). La idea de *Creative Commons* es producir licencias de copia para que artistas, autores, educadores e investigadores puedan usar para anunciar ciertas libertades en el uso de sus creaciones. Si la regla por default del copyright es “todos los derechos reservados” en las licencias *Creative Commons* la fórmula viene a ser “algunos derechos reservados”³⁵. Estas

³⁴ Según las estadísticas de SourceForge, uno de los mayores repositorios de proyectos libres en Internet, en septiembre de 2005, el 85% de los proyectos acogidos (54.000) usaban licencia GPL o una variante de ésta (LGPL, *Lesser General Public License*), mientras que un 15% (9.900) usaban licencias tipo BSD (BSD, Apache, *Artistic License*, MIT y Mozilla). No obstante, en estas estadísticas pueden estar sesgadas a favor de la GPL ya que es mayor el porcentaje de proyectos GPL que permanecen inactivos. Del mismo modo, proyectos de la talla de Linux (GPL), Apache (Apache license), PHP (LGPL), Perl (dual, GPL y *artistic license*) o Mozilla (Mozilla license) quedan excluidos de estas estadísticas porque no están radicados en Sourceforge sino que tiene sus propios sitios.

³⁵ Estas licencias se construyen sobre la base de combinatorias de cuatro derechos posibles que los creadores pueden conservar o liberar. (1) *Attribution*, lo que significa que el creador requiere que se le atribuya la autoría como condición para utilizar la obra. (2) Uso no comercial, el autor permite solamente usos no comerciales de su obra. (3) No derivaciones, el creador pide que su obra sea utilizada como tal y no como las bases de alguna otra cosa. (4) *Share*

licencias fueron impulsadas por los reconocidos legalistas Lawrence Lessig de la *Stanford Law School* y James Boyle, de la *Duke Law School*, bajo la premisa de los derechos de propiedad intelectual en la sociedad digital han perdido su principal cometido que era la difusión y el intercambio de ideas, y de que el conocimiento avanza sobre saberes previos.

Una discusión más detallada sobre las licencias y los derechos de propiedad intelectual se desarrollará más adelante (ver sección 3). Por lo pronto, cabe decir que a pesar de las diferencias planteadas, ambas corrientes insisten en diferenciar gratuidad de libertad lo que permite la aplicación de modelos de negocios basado en SL y constituye el primer elemento a considerar para asumir que la sustentabilidad económica del modelo. También plantean un cuestionamiento sobre el uso de los derechos de propiedad intelectual sobre el desarrollo de software -específicamente sobre la aplicación del copyright para la protección del código fuente de los programas de computación- y sobre el modelo de negocio basado en licencias propietarias que prohíban la modificación de los programas. Según estas corrientes, estas circunstancias impedirían desarrollos de software más veloces y de mejor calidad.

En síntesis, ya sea este SL u *open source*, el sentido del ser del software abierto -en oposición al cerrado- radica en una idea muy simple: cuando los programadores pueden leer, redistribuir y modificar el código fuente de un programa de computación, el programa evoluciona. Lo pueden mejorar, adaptar y reparar sus errores a una velocidad sorprendentemente superior a la que se podría llegar a través del método convencional de desarrollo de software (Raymond 1999a). De acuerdo con el movimiento *open source* este proceso evolutivo veloz produciría mejor software que el modelo cerrado tradicional, en el que sólo unos pocos programadores pueden ver la fuente y mientras que el resto está limitado a “usar ciegamente una opaca bola de bits” (www.opensource.org 2004). Esta premisa que enfatiza exclusivamente la eficacia, la superioridad técnica y productiva que genera el modelo organizacional del bazar. El movimiento del SL le agrega un discusión

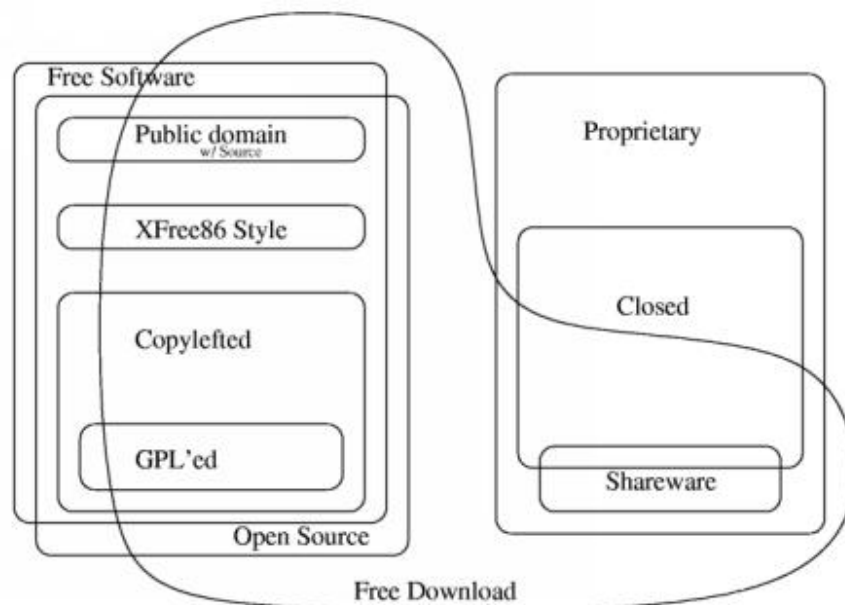
Alike cualquier derivación que se haga sobre la obra en base al uso de esta licencia debe estar también bajo una licencia *Share Alike*.

más profunda sobre las bases éticas y filosóficas de la cooperación y de la libertad (una discusión más profunda sobre la filosofía del SL como base para el cambio institucional se desarrolla en la capítulo 3).

2.3 Otros tipos de software

Además entre estos dos extremos de Software libre y Software propietario existen diferentes tipos de software, algunos de ellos hasta aquí no mencionadas, que se asocian a diferentes formas de distribución o de protección. Muchas veces surgen confusiones entre las diferentes formas que puede adoptar un programa de computación de acuerdo a la disponibilidad del código fuente, la gratuidad o el tipo de licencias que lo protege. Por ejemplo, no todo programa que esté disponible gratuitamente en la red es software libre y no todo software cerrado es propietario. En la siguiente tabla y figura se sintetizan y explican las formas más frecuentes con el solo objetivo de aclarar posibles confusiones y enriquecer el panorama

Figura 2.2 Diferentes categorías de software



Fuente: Chao-Kuei. GNU-Projet (www.gnu.org, 2005)

Tabla 2.3 Diferentes categorías de software

Software libre	El software libre es software que viene con autorización para que cualquiera pueda usarlo, copiarlo y distribuirlo, ya sea literal o con modificaciones, en forma gratuita o no, y que las distribuciones gocen de las mismas libertades que el programa original.
Software de Fuente Abierta	El término software de fuente abierta es aquel que es distribuido con el código fuente para que cualquiera pueda usarlo, copiarlo y distribuirlo, ya sea literal o con modificaciones, en forma gratuita o no. En este caso no es necesario que las distribuciones gocen de las mismas libertades que el programa original.
Software de dominio público	El software de dominio público es software que no está protegido con copyright. Dominio público es un término legal preciso que implica <i>sin copyright</i> .
Software protegido con copyleft	El software protegido con copyleft es software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando éstos redistribuyen o modifican el software. Esto significa que cada copia del software, aun si ha sido modificado, debe ser software libre.
Software abarcado por GPL	La GPL (General Public License/Licencia Pública General) de GNU es un conjunto específico de términos de distribución para proteger con copyleft a un programa.
El sistema GNU	El sistema GNU es un sistema operativo libre completo estilo Unix. Un sistema operativo libre estilo Unix consiste en muchos programas. Software GNU es software que es liberado dentro del sistema GNU. El software GNU está protegido con copyleft.
Software semilibre	El software semilibre es software que no es libre, pero viene con autorización para particulares de usar, copiar, distribuir y modificar (incluyendo la distribución de versiones modificadas). Ejemplo de esto es el software licenciado bajo licencias del tipo BSDs
Software privativo	El software privativo es software que no es libre ni semilibre. Su uso, redistribución o modificación está prohibida, o requiere que usted solicite autorización o está tan restringida que no pueda hacerla libre de un modo efectivo.
Freeware	El término <i>freeware</i> no tiene una definición clara aceptada, pero es usada comúnmente para paquetes que permiten la redistribución pero no la modificación (y su código fuente no está disponible). Estos paquetes no son software libre.
Shareware	El shareware es software que viene con autorización de redistribuir copias, e invita a quien continúe haciendo uso de una copia a pagar un cargo por licencia. El shareware no es software libre. Para la mayoría del shareware el código fuente no está disponible.

Fuente: Elaboración propia en base a GNU-Projet
(www.gnu.org, 2005)

2.4 Software libre vs. software propietario

La expansión del fenómeno del SL condujo a que en diversos ámbitos, tanto públicos como privados se llevara a cabo una fuerte discusión sobre las ventajas o desventajas de usar software libre *vis a vis* propietario.

No resulta evidente suponer un juego de un único ganador entre ambos modelos. Ya no es tan fácil asumir, como se ha hecho antes del surgimiento del Linux, que el modelo abierto sólo puede abrirse lugar en círculos científicos o técnicos. Aunque también resulta difícil creer que en algún momento que el SL reemplace totalmente al propietario. No obstante, es muy posible que siga ganando terreno en áreas donde se valore la inversión a largo plazo en una plataforma (consideración muy pertinente al sector público), la interoperabilidad, y la provisión descentralizada de soporte.

En el presente apartado se muestran algunos de los argumentos más frecuentes en la literatura con el objetivo de contextualizar el proceso de difusión del SL. Se analizan las ventajas del uso de SL en ambientes tales como instituciones científicas y académicas, empresas privadas y en la administración pública. Finalmente se exponen algunos argumentos sobre el rol del Estado en la promoción del uso y desarrollo de SL y algunas ventajas específicas para países en desarrollo.

2.4.1 El uso privado

Frecuentemente se confunde la libertad con la gratuidad a partir de la ambivalencia de la palabra inglesa 'free'. Lo primero que cabe aclarar cuando se explican los beneficios económicos del SL es que éste no es gratis por definición. Es posible -e incluso deseable desde un punto de vista de la sustentabilidad del modelo- la utilidad comercial del código libre: las distribuciones de software libre se comercian aunque, a precios sensiblemente más reducidos que las licencias de software propietario. Esto es así porque habitualmente

los modelos de negocios basados en SL se fundamentan en la prestación de un servicio más que en la venta de licencias.

A primera vista parecería que el uso de SL implica necesariamente un ahorro con respecto al uso de software propietario, aunque no implique gratuidad dado que no es necesario pagar licencias. Sin embargo, cabe aclarar que el licenciamiento del software constituye sólo una porción del costo total que implica el uso de una tecnología. Este costo total, conocido como *Total Cost of Ownership* (TCO) incluye, además de los costos de licencias, los costos asociados al hardware y al software y los costos de mantenimiento, integración, soporte y training, que se refieren a la contratación de personal calificado en condiciones de brindar estos servicios específicos.

El TCO no es fácil de calcular, y los estudios suelen ser intencionalmente parciales hacia uno u otro lado. De acuerdo con Ghosh (2004) los costos de licencias más los asociados al hardware y software representan entre un 15% y un 40% del TCO, y los elementos restantes (mantenimiento, integración, soporte y training) entre un 60% y 85%. La mayor participación de estos elementos está asociada al costo de la mano de obra calificada.

Si el costo de la licencia es relativamente bajo se puede aceptar que el TCO del SL y del software propietario son lo suficientemente similares como para no permitir generalizaciones. Esto es así en tanto se considere el contexto de las economías industrializadas, sin embargo, la situación es muy distinta en los países emergentes, especialmente aquellos que han logrado un buen nivel educativo y cierto desarrollo tecnológico (como Argentina, Brasil o India, por ejemplo). En estos países, al estar las licencias de software propietario cotizadas en dólares, y la mano de obra calificada en moneda local, se vuelve más ventajoso invertir en SL.

El costo de la licencia de Windows XP más MS Office asciende a 595 US\$³⁶ esto representa 1,4 meses de salario bruto promedio de un trabajador argentino³⁷, y 2,7 meses de PBI per cápita³⁸, lo que pone de manifiesto que el costo de la licencia sí importa.

³⁶ Versión estándar disponible en EEUU. Precio de Amazon.

De acuerdo con FLOSS (2002) aún en el caso de los países más ricos como Reino Unido, Alemania y Suecia el elevado costo de las licencias está dentro de los tres motivos más importantes para usar SL en organizaciones privadas. En estos países los primeros dos motivos son la performance y estabilidad de los productos libres y la seguridad que proveen.

Esto implica que el análisis de los costos de uso productos de software propietario es una mirada parcial ya que no considera la existencia de otros factores de altísima incidencia. Las limitaciones de interoperabilidad, la imposición de determinadas plataformas y más grave aún, la imposibilidad de adaptación a necesidades específicas (Arroyo, 2002) son algunos de los efectos de operar con plataformas cerradas.

2.4.2 El uso en la administración pública

La recomendación de utilizar SL en el sector público se está haciendo sentir cada vez más en varios países. Los más avanzados al respecto son sin duda los países de la Unión Europea. El grupo IDA (*Interchange of Data between Administrations*), perteneciente a la Comisión Europea, bajo la dirección de Patrice-Emmanuel Schmitz (de la consultora Unisys Belgium) preparó un informe preliminar titulado **Study into the use of Open Source Software in the Public Sector** (Estudio del uso de software Open Source en el sector público). En él se analizan las principales ventajas que presenta el uso del software de código abierto, y numerosos ejemplos de utilización en instituciones públicas de Francia, Alemania, Bélgica y España, entre otros países³⁹.

Libre acceso del ciudadano a la información pública

³⁷ 1.239\$ de acuerdo al Sistema Integrado de Jubilaciones y Pensiones y considerando un tipo de cambio de 2,9 pesos por dólar. Fuente Dirección Nacional de Programación Macroeconómica.

³⁸ Siendo el PIB per cápita de 7.700\$ para el 2004. Fuente INDEC.

³⁹ Las ventajas del SL postuladas en este estudio coinciden en gran medida con los fundamentos de los proyectos de ley que se están proponiendo hoy en nuestro país y con la exposición del congresista peruano Edgardo Villanueva Núñez en respuesta a la carta de Microsoft Perú.

El Estado es depositario de una gran cantidad de información provista por sus ciudadanos, quienes son dueños de dicha información. En un estado democrático es necesario garantizar el acceso de la ciudadanía a estos datos, por lo que es imprescindible que el acceso no esté sujeto a ninguna restricción imponible por un proveedor único. Los formatos de datos abiertos (es decir, formas de almacenamiento que siguen normas y procedimientos públicamente disponibles) garantizan que el Estado o cualquier entidad interesada pueda generar software capaz de leer, interpretar y procesar esta información.

Perennidad de los datos públicos

Juntamente con la forma en que se almacena la información, es preciso que toda la estructura informática que utiliza el Estado sea inmune a las condiciones monopólicas de una empresa privada o a las vicisitudes que puedan afectarla. El mantenimiento de las muy variadas plataformas que utiliza el sector público no puede depender de lo que en el mercado se considere "obsoleto". Utilizando SL, el proveedor del software no es el único en condiciones de brindar mantenimiento.

Seguridad del Estado y de los ciudadanos

A medida que avanza la digitalización de los procesos burocráticos, los alcances y responsabilidades del software que utiliza el Estado se vuelven cada vez más críticos. Esta criticidad impone la necesidad de que sea posible auditar el funcionamiento del software; esta auditoría sólo es posible si se dispone del código fuente del software utilizado. Existen numerosos ejemplos de software propietario que, se descubrió, contenía accesos no documentados ("backdoors") que le permitían al proveedor (o a todo aquel que conociera el secreto) acceder ilegítimamente a los datos del cliente. Los proyectos de la ciudad de La Plata y de la Ciudad de Buenos Aires citan, por ejemplo, el resonado caso de la clave NSA (*National Security Agency*, organismo de inteligencia estadounidense) descubierta casualmente en el sistema operativo Windows NT, cuyo propósito aun se desconoce.

2.4.3 El uso académico y científico

Además de los argumentos ya mencionados existen razones específicas que justifican la utilización del SL en la academia.

De acuerdo con Martínez (2003) la utilización de SL en la enseñanza de informática resulta esencial. Si se considera que el objetivo de la educación universitaria no es formar técnicos expertos en los detalles particular de herramientas específicas, sino profesionales con una formación integral basada en conceptos que trasciendan la moda, el software libre es una manera natural de permitir al docente y sus estudiantes el análisis de las tecnologías y conceptos utilizados en la construcción de la pieza de software. La utilización de software propietario implica la incapacidad de conocer el código o cualquier aspecto que no esté relacionado con la ejecución del programa, hace que se limite el papel del profesional que está siendo formado al de un usuario calificado del sistema que desconoce los mecanismos de funcionamiento (*know why*), limitando de esta forma la adquisición de saberes al plano del *know what*⁴⁰. Por otra parte, la utilización del SL promueve el desarrollo de capacidades a través de la participación en comunidades de usuarios en donde el intercambio fomenta procesos de aprendizaje del tipo *learning by interacting* (ver capítulo 3).

En lo que se refiere a la investigación científica, la utilización de software propietario en experimentos compromete la verificabilidad de las conclusiones (Pfaffenberger, 2000). Tradicionalmente, la producción científica toma la forma de *papers* publicados en revistas o *journals* especializados. Estas revistas toman la decisión de publicación sobre la base de un severo escrutinio del *paper* realizado por reconocidos miembros de la comunidad científica. La creciente utilización de las tecnologías informáticas en la actividad científica permitió reducir costos de experimentación ya sea informatizando el análisis de resultados o con la utilización de modelos de simulación. No obstante, muchos de estos programas que habilitan estas prácticas son propietarios y por lo tanto no son públicos los algoritmos que utilizan en sus procedimientos. Como software propietario no

⁴⁰ Federico Heinz, se refiere a esta cualidad como “formar monos de apretar *reset*”.

se puede acceder al código fuente no es posible inspeccionar el corazón de los algoritmos utilizados y por lo tanto no se puede determinar la validez de los mecanismos lógicos supuestos. En este sentido la verificabilidad de los resultados obtenidos se ve afectada.

2.4.4 Algunos argumentos adicionales a favor de la promoción del SL

Externalidades y estándares

La intervención pública en presencia de externalidades ha sido justificada hasta por los más acérrimos defensores del libre comercio. Desde esta perspectiva se justifica que el Estado promueva el desarrollo de SL ya que implica una creación del espacio público y la valorización del capital social y la participación en desarrollos libre alimenta el proceso de generación de capacidades.

Los estándares abiertos representan una ampliación del espacio público. La elección de una tecnología muchas veces está asociada a la elección implícita de un estándar⁴¹, en presencia de economías de red la elección de un individuo de la tecnología está afectada por el número de individuos que compongan la comunidad de usuarios, ya que el valor de la tecnología se incrementa con el número de usuarios que tenga. De tal forma, la elección individual de un estándar dado genera efectos *feedback* positivos e incrementa la probabilidad que el próximo agente elija el mismo estándar. Si las economías de red son lo suficientemente fuertes, de las acciones descentralizadas de los agentes en el mercado surgirá un único estándar, denominado frecuentemente estándar *de facto*⁴². Como ha sido demostrado por diversos autores (Arthur, 1989; David, 1985) no existe nada en este mecanismo que *a priori* garantice la emergencia del estándar socialmente óptimo. El estándar que surge de este proceso puede ser abierto o propietario, según si las

⁴¹ Uno de los ejemplos más comunes sobre estándares competitivos se refiere al mercado de las video caseteras, donde competían VSH por un lado y Betamax de Sony por el otro.

⁴² El opuesto al estándar *de facto* es el *de jure*, asociado a especificaciones técnicas que surgen de un consenso entre los jugadores dominantes del mercado y se formaliza a través de los organismos oficiales de estandarización (Hilbert, 2004).

especificaciones son de dominio público y está permitida su explotación comercial, o si los usos están reservados al creador que encuentra protección legal (DPI) o a través de trabas tecnológicas (imposibilidad de realizar ingeniería reversa). Evidentemente esta protección posibilita el surgimiento de posiciones monopólicas. Desde la perspectiva del usuario, por ejemplo una empresa privada, la dependencia que implica la inversión en una tecnología cerrada y monopólica es un riesgo que aumenta con el tiempo. Por ejemplo, antes de Internet, varios proyectos de construir redes globales de computadoras fracasaron por confiar en protocolos propietarios que pronto encontraron problemas de interoperabilidad. Fueron necesarios estándares abiertos (TCP/IP, HTTP, etc) para lograr la amplia distribución que tiene Internet. Por eso, la enorme inversión económica en digitalización hace cada vez más necesario software de base de código abierto⁴³. En este sentido existe una percepción generalizada de que los estándares abiertos son de interés públicos ya que (i) facilitan la interoperabilidad alcanzando mucho más rápidamente los beneficios derivados de las economías de red, (ii) reducen los efectos *lock-in vendor*⁴⁴ porque la existencia de múltiples oferentes facilita cambiar de proveedor, (iii) incrementan la competencia y finalmente, (iv) el desarrollo de mercados basados en este estándar no depende del éxito o fracaso de una sola firma (Hilbert, 2004).

Ventajas económicas y de promoción al desarrollo local: la perspectiva de los países en desarrollo

La utilización del SL en el Estado puede ser justificada por el impulso que pueda dar sobre la industria de software y de esta forma contribuir al cambio del perfil de especialización de estos países. Desde una perspectiva dinámica, las demandas del Estado pueden promover al desarrollo de la industria local de servicios asociados a la implementación y mantenimiento SL. En la actualidad las necesidades informáticas del Estado son satisfechas por empresas extranjeras, por desarrollos propios y en algunos

⁴³ En esta línea, es posible observar cómo el SL tiene muchísimo más peso en las áreas de infraestructura informática (sistemas operativos, software de comunicación, servidores) que en las áreas más nuevas y cercanas al usuario (aplicaciones de escritorio, juegos, etc).

⁴⁴ Atrapado por un único proveedor.

casos por provisiones de empresas locales. A pesar de que actualmente se esté implementando plataformas abiertas en el Estado aún predominan las propietarias. Por otra parte, el Estado constituye uno de los demandantes más grandes de informática del mercado y sus requerimientos contribuyen a la fijación de estándares y a la formación de aprendizajes. Desde esta última perspectiva, la adopción de SL por parte del Estado actuaría como efecto demostración, en un proceso de mayor difusión del SL (ver capítulos 4 y 5). Un beneficio adicional es que los dividendos quedarían en la economía local, en lugar de ser girados a los países centrales donde están las casas matrices de las grandes productoras.

Las empresas de software tiene básicamente dos formas de entrar al mercado: o bien con productos estándar para uso masivo o bien con desarrollos a medida. Tradicionalmente, las grandes empresas de software han crecido sobre la base de un modelo de negocios basado en la venta de licencias de productos masivos. Las empresas locales de países en desarrollo estuvieron acotadas a resolver las necesidades del mercado interno donde las grandes empresas no podían entrar ya sea por problemas idiomáticos o idiosincrásicos⁴⁵. En este sentido, frecuentemente se asocia al modelo de venta de licencias con el éxito. Podemos notar que en la industria de software (y mucho más en el caso particular de los países latinoamericanos) la gran parte del trabajo se presta en forma de servicios y productos hechos a medida; la incidencia de los productos de venta masiva (el "software en cajitas") no es el principal motor del sector. El modelo de negocios propuesto por el software de código abierto se basa en la prestación de servicios (en oposición a la venta de licencias). Esto podría dar la pauta, en primer lugar, de una tendencia hacia una industria del software más basada en los servicios⁴⁶; en segundo

⁴⁵ Más recientemente, la generalización del outsourcing profundizó esta tendencia. En los últimos años se instalaron en una diversidad de países en desarrollo las denominadas *software factories*. A través de estas factorías segmentan el procesos de desarrollo de software y terciarizan en países con bajo costo laboral las parte menos complejas y más monótonas del proceso, como el *coding* y el *testing*.

⁴⁶ En el capítulo 3 se discuten las características del cambio organizacional en el área del desarrollo del SL. Como será en ese apartado, las nuevas metodologías de desarrollo están más cercanas a modelos de prestación de servicios que a los de venta de licencias.

lugar que el modelo de servicios no es forzado o ajeno a la realidad de los países en desarrollo; y, en tercer lugar, que la adopción del SL en los países en desarrollo permitiría acercarse a los países desarrollados y complejizar el perfil de especialización de las economías en desarrollo.

2.5 Síntesis y conclusiones

En este capítulo se han recorrido los principales puntos de la historia del movimiento del SL en el contexto de la evolución de la actividad informática. Sobre esa base fue posible precisar definiciones y distinguir las distintas corrientes que conviven al interior del movimiento y los argumentos centrales de quienes reclaman una mayor participación pública en el desarrollo y promoción del SL.

La evolución histórica del fenómeno demuestra que el movimiento del SL no solo no es marginal en la actualidad sino que no lo ha sido previamente. La historia del SL es parte de la historia de la informática: el desarrollo de software por sus características técnicas se prestó naturalmente a los desarrollos colaborativos con la diferencia histórica de que la protocomunidad de desarrolladores previa al crecimiento de la WEB quedó desdibujada por el efecto escala de miles de desarrolladores, *debuggers*⁴⁷, testers y usuarios participando al mismo tiempo en desarrollos colaborativos *on line* (en línea), sobre el modelo de organización del SL. El crecimiento del movimiento fue tan vertiginoso como el crecimiento de Internet en si mismo, y en esa explosión de actividad surgieron corrientes que obligaron a precisar las definiciones, objetivos y el alcance político e institucional del movimiento.

En este contexto, han surgido las principales diferencias entre quienes enfatizan el aspecto técnico-organizacional del SL y quienes rescatan una visión política-institucional que permitiría redefinir el alcance de los derechos de propiedad intelectual. La acción

⁴⁷ Por *bug* se entiende una falla en el programa informático. Muchas veces fallas o errores en los sistemas informáticos era atribuidos a la presencia de insectos en el hardware. *Bug* significa insecto en inglés.

pública se vuelve ineludible en la medida en que se transforman los métodos tradicionales de protección de propiedad intelectual y se abre el juego a un mayor número de participantes. La acción privada y unipersonal deviene grupal o comunitaria y la propiedad, colectiva. El predominio de bienes club y de las formas de protección laxas no son arbitrarios sino coherentes con el ritmo de innovación requerido en estas nuevas actividades.

3 ALGUNAS NOTAS SOBRE LA ECONOMÍA DEL SOFTWARE LIBRE

3.1 A título de introducción o la economía digital ¿el fin de la era de la escasez?

El surgimiento y la expansión de las tecnologías informacionales produjo importantes cambios en las formas de organización, producción y en los patrones de consumo. Sobre la premisa de que los bits se diferenciaban radicalmente de los átomos, distintas disciplinas sociales se vieron obligadas a revisar conceptos y dar explicaciones de fenómenos que antes parecían marginales. La organización de la producción de conocimiento, que ya venía siendo revalorada con anterioridad a la aparición de los modelos de crecimiento endógeno, obtuvo un lugar central en la literatura sobre desarrollo y crecimiento económico, que consideraron crecientemente el rol de las externalidades, los procesos de aprendizaje y el cambio tecnológico como fuentes fundamentales y endógenas del desarrollo económico.

La relevancia que adquirió la información y el conocimiento en las actividades productivas condujo a que se analizaran con más detalle las particularidades económicas de estos bienes tales como su intangibilidad, no rivalidad en el consumo y excludibilidad limitada. De la misma forma se destinaron importantes esfuerzos a comprender más cabalmente los procesos de creación, difusión y apropiación de estos bienes que cobraban protagonismo en el nuevo escenario tecnológico. Los fenómenos de economías de red, basados en la creciente interoperabilidad, consecuencia de los procesos de descentralización productiva y la emergencia de nuevas formas organizacionales, pusieron de manifiesto la interdependencia de las decisiones individuales y que los efectos acumulativos de estas decisiones no coordinadas ex-ante podrían conducir a situaciones subóptimas y a efectos *lock-in*.

La producción y consumo de programas de computación como un nuevo sector emergente de la economía digital planteó un conjunto de nuevos problemas asociados a lo que parecía el fin de la era de la escasez, que compartía con otros productos

digitalizables, como la música, las películas y los libros y con otros sectores propios del nuevo paradigma tecnoproductivo como la biotecnología. La posibilidad de reproducción a costo ínfimo implicó una verdadera revolución en la transmisión de la información, la difusión y el consumo de bienes informacionales que aún no se ha estabilizado, al permitir una mayor separabilidad, intercambiabilidad, disponibilidad y transporte de bits de información por interacciones *on-line* (en línea). La difusión de Internet, como una red abierta de comunicación digital, profundizó a una velocidad pasmosa los cambios que ya estaban presentes en la era de los disquetes de 5-1/4': la posibilidad de reproducir, intercambiar y modificar contenidos digitales a costo ínfimo.

Uno de los desafíos centrales de la economía en tanto disciplina social está en dar cuenta de la forma de valorización en el mercado de activos cuyo costo de reproducción tiende a cero. En este contexto, la discusión que se ocupa de analizar las interacciones entre difusión y apropiación de conocimiento ha tomado un lugar clave. Una primera forma de aproximación a esta problemática es a través del estudio de las formas institucionales de protección de la propiedad intelectual, es decir de los derechos de propiedad intelectual. Partiendo de la base de que la información puede ser considerada un bien público en tanto sus características de no rivalidad y difícil excludibilidad, son necesarias de formas institucionales que resuelvan los problemas intrínsecos de su naturaleza. Los derechos de propiedad intelectual aparecen entonces como la forma de incentivar la actividad creativa: si, mediante formas institucionales, no se garantizaran ganancias extraordinarias durante un período de tiempo, los beneficios que surgirían del libre accionar de los mercados no justificaría la inversión en el desarrollo de innovaciones y creaciones. En esta discusión no solamente participaron y participan académicos e intelectuales sino importantes grupos económicos, asistidos por sus imponentes departamentos legales, como las compañías discográficas, la industria del cine, la farmacéutica, entre otras, reclamando mayor protección sobre estos bienes intangibles, base central de su negocio. En oposición, un grupo intelectuales y tecnólogos junto con algunas organizaciones no gubernamentales han puesto crecientemente resistencia al avance de los derechos sobre

áreas que tradicionalmente pertenecían al dominio público o eran consideradas de uso justo⁴⁸ o áreas que plantean serios conflictos morales⁴⁹.

El surgimiento del SL, y su injerencia en la industria del software puso de relieve que, en el nuevo paradigma, pueden resultar incluso más efectivas formas de apropiación distintas de los medios legales de protección de copia. La historia relatada en la sección previa muestra cómo el SL fue ganando espacio en el mundo de los negocios ya sea porque para algunas grandes empresas era funcional a su estrategia competitiva o porque surgieron empresas que basaron enteramente su negocio en la provisión de servicios asociados al SL. Sin embargo, la mayor parte de los trabajos abocados a “la economía del SL” suponen que, *a priori*, no es razonable que los agentes elijan liberar su producción intelectual. En ese supuesto subyace otro sobre la naturaleza del conocimiento: sintéticamente se lo puede resumir en que información y conocimiento son sinónimos.

Si el problema de la naturaleza del conocimiento se resuelve suponiendo que el conocimiento es información y entonces un bien público que requiere de medios legales de protección para garantizar no haya sub-inversión en su generación, queda por dar respuesta a por qué los agentes de la comunidad de desarrolladores de SL eligen generar este conocimiento y usarlo económicamente. Sin embargo, si se asume que el conocimiento es radicalmente distinto a la información en la medida en que no todo conocimiento es codificable, que nunca es una unidad cerrada en sí misma sino que está en constante evolución y transformación, que existen diferentes formas y fuentes de aprendizaje, que su valor puede crecer o desaparecer con el uso y la difusión y que las formas de apropiación no siempre dependen del control, entonces la decisión de liberar los productos de la actividad creativa puede ser perfectamente considerada como una decisión económica racional, dentro de los límites impuestos por la toma de decisiones en condiciones de incertidumbre radical.

⁴⁸ Por ejemplo, prestar un libro o cantar una canción en un campamento. Actualmente en EEUU está reglamentada la cantidad de préstamos inter-bibliotecarios que pueden hacerse y la ASCAP (American society of composers authors and publishers) amenazó con demandar a los campamentos de *girls scout* por cantar sus canciones en los fogatas nocturnas sin pagar *royalties*.

⁴⁹ Como la patentabilidad de genes o de algoritmos matemáticos.

En la sección 3.2 de este capítulo se enmarca la cuestión del SL en el análisis del cambio tecnológico, organizacional e institucional. Seguidamente, en la sección 3.3 se debaten los argumentos centrales acerca de la cuestión de las motivaciones que es en lo que pareciera focalizarse la economía del SL y se propone una visión que supera la falsa dicotomía entre altruismo y búsqueda de lucro personal.

3.2 Cambio organizacional e institucional en el nuevo contexto tecnológico

El SL puede ser leído como un fenómeno donde se manifiestan transformaciones interrelacionadas en las esferas institucional, organizacional y tecnológica; ya que se trata de un exponente del nuevo paradigma tecno-productivo que revoluciona la organización de la producción del conocimiento a partir de la utilización de nuevas tecnologías, desarrolla nuevas formas institucionales acordes con estos cambios y discute los fundamentos de la modalidad convencional de apropiación del conocimiento a través de los derechos de propiedad intelectual tradicionales.

La discusión sobre los derechos de propiedad intelectual y el surgimiento de nuevas formas de licencias que propician medios innovadores de apropiación del conocimiento conforman las transformaciones institucionales. El desarrollo de software en comunidades epistémicas y la apropiación de los beneficios económicos de estos desarrollos sobre la base de servicios hace necesario que estén bien definido el conjunto de reglas y normas de comportamiento, que, sustentado en una serie de principios éticos, justifique la apropiación comunitaria del SL, desacredite el sistema tradicional de apropiación individual y penalice las conductas *free-rider*.

Por otra parte, el SL significa también una forma innovadora de organización de la producción de conocimiento, con una mayor descentralización gestionada por un conjunto de reglas comunitarias; es decir, implica un cambio organizacional. En este caso específico, la producción de conocimiento se refiere al desarrollo de software: sin embargo, en algunos casos (por ejemplo, en los acuerdos de cooperación para desarrollos tecnológicos entre empresas) puede verse una expansión de las formas comunitarias o

cooperativas de desarrollo de conocimiento, afianzadas a través de reglas institucionales explícitas. La descentralización del desarrollo es inseparable de la libertad del desarrollo (*freedom*). Es decir, que el software está abierto y es susceptible de cambios que pueden hacerse de una forma descentralizada. Esto garantiza una aceleración en el ritmo de innovaciones que no sería posible en el esquema de organización del software propietario.

Por último, puede notarse que ambas transformaciones se dan en el marco del afianzamiento de un nuevo paradigma tecnológico de información y comunicación que permitió reducir los costos y aumentar la velocidad de las comunicaciones. Dentro de este paradigma la generalización de Internet y el desarrollo de la Web resulta un hito clave, ya que significó a través de la convergencia hacia estándares comunes, la expansión de las redes públicas de comunicación digital. Es decir, una expansión del espacio público.

Estos puntos serán discutidos a lo largo de este apartado.

3.2.1 Cambio tecnológico y organizacional

El desarrollo de software ha planteado tradicionalmente una serie de complicaciones que la industria tuvo que enfrentar en la medida que la actividad transitaba desde ambientes académicos hacia el mundo de los negocios. El desarrollo de software mostró prontamente las características de las actividades intensivas en conocimiento y sus dificultades asociadas con la incertidumbre, la gestión y el control.

La primera respuesta fue la adaptación y aplicación de métodos de la ingeniería convencional al desarrollo de programas de computación. La metáfora de asociar el desarrollo de software a la construcción se instaló en el sector y las tareas rápidamente se organizaron y tomaron los nombres propios de la ingeniería civil: ingenieros, arquitectos, diseñadores y desarrolladores (o simplemente constructores).

En términos generales, las metodologías ingenieriles se han caracterizado principalmente por separar el diseño y la construcción, con el objetivo fundamental de volver predecible

y controlable la mayor parte de la actividad. Además de este, otros objetivos también justifican el énfasis de las ingenierías de planificar antes de construir. En primer lugar, puede mencionarse el objetivo de división del trabajo entre personas con diferentes habilidades: los diseñadores que cuentan con elevadas capacidades intelectuales y los constructores que poseen principalmente habilidades manuales. De esta forma, los constructores pueden a partir del plano llevar adelante la obra sin más indicaciones, sin tener que intervenir en la toma de decisiones. Por otra parte, como el plano determina qué es lo que hay que hacer y cómo hay que hacerlo también determina las interdependencias que existen entre las distintas tareas. Esto último es crucial ya que permite establecer plazos, asignar recursos y presupuestar las diferentes etapas. En definitiva, el diseño previo permite ejercer el control sobre la ejecución sujeta a un contrato: el plano. El principio fundamental de la planificación es anular posibles desviaciones con relación al plano original, por lo que su naturaleza es de resistencia al cambio. En una obra de construcción civil, digamos la construcción de un puente, esto se comprende rápidamente ya que la construcción es siempre mucho más costosa y requiere más tiempo que la planificación. Por lo tanto, es clave poder separar la etapa de diseño, creativa e incierta, para volver a la construcción predecible y controlable (Fowler, 2003).

La aplicación de estas metodologías sobre el desarrollo de software condujo, efectivamente, a una mayor posibilidad de predicción y control sobre el desarrollo, pero a su vez le restó flexibilidad y adaptabilidad ante los cambios. Por seguir con la metáfora, en la construcción de un puente queda claro que los requerimientos no cambian (soporte que permita el tránsito de un lado al otro de un río), en oposición en el desarrollo de un software es muy probable que los requerimientos cambien (desde el vamos es muy difícil que un cliente esté en condiciones por sí mismo de describir exhaustivamente todos sus requerimientos) y lo natural sería que el proceso sea adaptable. Como se ha mencionado, las metodologías ingenieriles resisten el cambio ya que no están preparadas para ser adaptables. Otro efecto de la aplicación de estas metodologías al software es que han puesto a los procesos por arriba de las personas. Asumiendo que los saberes de los desarrolladores están acotados a seguir el plan y que el de los clientes a describir de una vez y para siempre sus necesidades, estas metodologías buscan circunscribir la parte creativa de la actividad en la etapa del diseño. De esta forma se desperdician las

capacidades que tiene para ofrecer los clientes de ajustar con mayor precisión el producto a sus necesidades y los desarrolladores de conocer las ventajas y limitaciones de las tecnologías que se apliquen así como las condiciones en que se desenvuelve el proceso de desarrollo. Asimismo, al pensar a los desarrolladores como recursos lineales, conduce a suponer que incorporando el número suficiente de programadores se pueden acortar los plazos de desarrollo. Esto no puede ser así en actividades intensivas en conocimiento, donde las interacciones entre las partes son mucho más ricas y fructíferas que las ganancias de productividad que se puedan obtener de la división de tareas en compartimentos estancos (Cockburn, 1999). Una última consecuencia de la aplicación de las metodologías ingenieriles es que permitieron concebir al software como un producto cerrado: en el modelo ingenieril del desarrollo de software, desde el comienzo, hay una visión clara de lo que va a ser el producto al concluir el proyecto. Esta visión impide concebir al software como en evolución constante, que es, como ya hemos visto, una de las características centrales de SL.

En oposición a las metodologías ingenieriles, recientemente se ha ido consolidando un paradigma alternativo conocido como el de las metodologías ágiles. La forma organizacional del SL tiene muchos puntos de coincidencia con este nuevo paradigma⁵⁰. Ambos grupos comparten la perspectiva del software como un proceso evolutivo, han comprendido las limitaciones de las formas tradicionales de organización y plantean alternativas que tienen como punto fuerte una mayor interacción con los clientes y con los desarrolladores. Las metodologías ágiles aprendieron mucho de la forma organizacional del SL, sobre todo la adoptada después del surgimiento de Internet.

Las máximas que componen el manifiesto del desarrollo ágil⁵¹ muestran justamente esta interacción y aprendizaje:

⁵⁰ De todas formas cabe aclarar que las metodologías ágiles son aplicadas tanto a desarrollos libre como propietarios. Se trata de una forma novedosa de organizar el desarrollo de software. La metodología de desarrollo del SL puede ser considerada por Fowler (2003) como una metodología ágil.

⁵¹ www.agilmanifiesto.org (2005)

- Los individuos y sus interacciones por sobre los procesos y las herramientas
- Software que funcione por sobre documentación exhaustiva⁵²
- Colaboración con el cliente por sobre la negociación de contratos
- Respuesta al cambio por sobre seguir estrictamente un plan.

Esto conduce a que los métodos ágiles sean adaptables en lugar de predictivos. Mientras que los métodos ingenieriles tienden a intentar planear una gran parte del proceso en detalle para un plazo largo de tiempo, los métodos ágiles intentan ser procesos que se adaptan y crecen en el cambio. En relación con la dicotomía gente o procesos, los métodos ágiles están orientados a la gente. La meta de los métodos ingenieriles es definir un proceso que funcionará bien con cualquiera que lo use (Fowler, 2003), partiendo de la premisa de que todo el proceso puede ser codificado en forma exhaustiva⁵³. Los métodos ágiles, por su parte, afirman no sólo que eso no es posible sino tampoco deseable, ya que implicaría suponer que los desarrolladores son todos idénticos entre sí y que sus capacidades sean sumables más que articulables. Las metodologías ágiles al igual que el SL enfatizan el trabajar a favor de la naturaleza humana y no en su contra (Fowler, 2003), ya que de este modo se podrán alcanzar el máximo desarrollo de las capacidades cognitivas del grupo de trabajo. Los métodos ágiles proponen la creación de un ambiente propicio para la creación, con predominio de formas organizacionales horizontales, mayor comunicación e interacción entre las partes y un control por resultados más que procesos. La forma organizacional del SL, además de los mencionados, va a hacer uso de otros incentivos como la satisfacción del ego, la autorrealización, el deseo de desarrollar nuevas capacidades, la pertenencia una comunidad y la reacción contra las formas tradicionales de organización y comercialización de software⁵⁴.

⁵² Descripción y codificación completa del proceso.

⁵³ En los próximos apartados se discuten, justamente, los límites en la codificación de tareas y procesos desde una visión de las ciencias cognitivas.

⁵⁴ En los próximos apartados se ampliarán estas cuestiones.

Si bien la metodología del SL es una metodología ágil, el fenómeno del SL es más complejo, porque mientras las metodologías ágiles plantean una forma innovadora de organización de la producción de software, el SL, además de contribuir a la cuestión organizacional, también plantea una transformación en la forma de distribución y comercialización de software y en las formas institucionales de protección a la creación intelectual⁵⁵.

Como se ha mencionado en el capítulo previo, Raymond (1999) fue el primero en caracterizar la forma de organización predominante en los desarrollos libres (que denominó como *bazar*) y compararla con la de los propietarios (*catedral*). En su artículo *La catedral y el bazar* este autor propuso los elementos centrales que definen al modo de organización del SL. A continuación analizaremos algunos de estos elementos⁵⁶ para luego explicar los efectos de la modularidad combinada con la descentralización como elementos claves del modo organizacional del bazar.

Un conjunto importante de las premisas enumeradas por Raymond, al igual que el manifiesto sobre metodologías ágiles, hacen hincapié sobre los problemas de traducción entre usuarios⁵⁷ y desarrolladores⁵⁸. De acuerdo con este autor, el modo organizacional

⁵⁵ De todas formas, entre ambos movimientos también existen diferencias en la cuestión organizacional: las metodologías ágiles no contemplan un elemento central del desarrollo del SL que es la total descentralización operativa y coordinada a través de la red sobre la base de una comunidad preexistente. De hecho una de las consignas centrales de las metodologías ágiles es el trabajar en grupos reducido y revalorar el contacto cara a cara, situación claramente alejada del paradigma del SL.

⁵⁶ No todos los ítems señalados por Raymond son tratados aquí, ya que muchos de ellos están demasiado asociados a cuestiones técnicas que escapan al alcance de esta investigación. De igual forma, la exposición de los elementos fue reordenada para seguir la línea argumentativa del apartado.

⁵⁷ En el caso del SL parece más apropiado hablar de usuarios que de clientes, en la medida que no es necesario que hayan ventas, particularmente al interior de la comunidad; sin embargo, en el marco de la discusión sobre las traducciones pueden entenderse como sinónimos.

⁵⁸ De acuerdo con numerosos autores (Lundavall, 1992, Nonaka y Takeuchi 1995, Rulani 1994, Yoguel y Boscherini, 2001) el conocimiento se valoriza en una continua traducción entre los lenguajes científico-tecnológicos y de los saberes empíricos-contextuales.

del SL busca dar alguna respuesta a esta cuestión, inherente a todo proceso de innovación o de desarrollo tecnológico.

Raymond propone que: (i) *Todo buen software comienza a partir de las necesidades personales del programador. O dicho en otros términos, todo buen trabajo empieza cuando uno tiene que rascarse su propia comezón.* Esta primera premisa del modo del bazar pone de relieve la importancia de la comunicación entre usuario y desarrollador al punto tal que afirma que el mejor software es aquel donde el usuario y el desarrollador son la misma persona. En este caso extremo los problemas de traducción son por definición nulos. (ii) *Tratar a los usuarios como colaboradores es la forma más apropiada de mejorar el código, y la más efectiva de depurarlo de errores.* Los usuarios son colaboradores porque conocen la mejor forma de aplicar el software en la resolución de un problema concreto. Alguien que se toma el trabajo de descargar el código fuente de un programa libre instalarlo y ponerlo en funcionamiento en una tarea específica sabe cuáles son las debilidades del código y muy probablemente tenga algo para modificar o mejorar⁵⁹. (iii) *Libere rápido y a menudo, y escuche a sus usuarios,* una vez más insiste en la importancia de la voz del usuario y agrega liberar rápido y a menudo, que puede ser leído como la voz del desarrollador. Liberando los desarrollos parciales, los potenciales usuarios tienen la posibilidad de conocer la oferta de código disponible y participar en el proyecto corrigiendo el código. Esta premisa se opone fuertemente al modo de desarrollo de la catedral, donde solamente versiones ya testeadas son liberadas y está muy cerca de las metodologías ágiles. Fue Linus Torvalds quien dio un giro fundamental sobre el estilo de lanzamiento de versiones usado hasta el momento (incluso por la FSF). El riesgo de liberar con frecuencia es toparse con un error insondable que termine por cansar y aburrir a los usuarios-colaboradores; sin embargo, Linus tenía en la cabeza una cuarta premisa que anula esta posibilidad: (iv) *Dada una base suficiente de desarrolladores asistentes y beta-testers, casi cualquier problema puede ser caracterizado rápidamente, y su solución ser obvia al menos para alguien; o simplemente con muchas miradas, todos los errores*

⁵⁹ Linus Torvalds, el creador de Linux, dijo “básicamente soy una persona muy perezosa que le gusta obtener crédito por lo que realmente hacen los demás”.

son superficiales. Otras premisas que insisten sobre la importancia de la comunicación entre los usuarios y los desarrolladores son (v) *Lo más grande, después de tener buenas ideas, es reconocer las buenas ideas de sus usuarios. Esto último es a veces lo mejor.* O (vi) *Si el coordinador de desarrollo tiene un medio al menos tan bueno como lo es Internet, y sabe dirigir sin coerción, muchas cabezas serán, inevitablemente, mejor que una.* De acuerdo con estas premisas se puede afirmar que el modo organizacional del SL encuentra su espacio natural en la red y en una comunidad de desarrolladores que la precede. El éxito de un programa libre está en saber utilizar esos recursos la red de comunicaciones pero fundamentalmente la red de desarrolladores. En este sentido, el valor de un programa libre no está asociado al trabajo que contenga sino al que pueda comandar en el futuro, como afirma Raymond: *“cuando uno comienza la construcción del edificio comunal, lo que debe ser capaz de hacer es presentar una promesa plausible. El programa no necesita ser particularmente bueno. Puede ser burdo, tener muchos errores, estar incompleto y pobremente documentado. Pero en lo que no se puede fallar es en convencer a los co-desarrolladores potenciales de que el programa puede evolucionar hacia algo elegante en el futuro”.*

Hay dos elementos que subyacen a las máximas de Raymond y que son el eje central del modo de organización del bazar: la modularidad y la descentralización. Ambos elementos presentes en la forma organizacional del bazar, no significan por si solos el éxito de la forma organizacional sino que requieren ser combinados de una manera virtuosa sobre la base de la existencia de una comunidad previa. Estos tres elementos distinguen al modo de organización del SL y garantizan una mayor eficiencia no solo tecnológica sino también en la gestión del conocimiento y en el ritmo de innovaciones.

Un sistema modular implica la existencia de sub unidades independientes interconectadas a través de interfases claras que componen un sistema mayor. Es más fácil crear y mantener un sistema modular ya que puede identificarse claramente cuáles son las tareas y vinculaciones de cada módulo y por lo tanto detectar más fácilmente las fallas (Simon, 1962). Por otra parte, es posible que los bloques sean reutilizados en diferentes productos con la misma o similar funcionalidad. En un sentido dinámico, la modularización permite más fácilmente la sustitución de componentes, por lo que la mejora del producto puede

no requerir un rediseño en su conjunto sino simplemente la sustitución de algunos de sus módulos, lo que reduce la complejidad de alterar el producto (García, 2001).

Los sistemas modulares se han utilizado en software desde siempre no obstante, combinados con la descentralización planteada por el SL adquieren una relevancia particular.

Frecuentemente en la bibliografía sobre formas organizacionales surgen como condiciones polares la descentralización coordinada a través del mercado y la jerarquía en el interior de una organización (Coase, 1937; Williamson, 1975; Chandler, 1977). Recientemente el tránsito hacia formas organizacionales coordinadas en redes o tramas productivas redefinió la discusión incorporando formas intermedias donde predominan intercambios administrados que sobrepasan las relaciones de compra-venta y donde no siempre estén presentes relaciones jerárquicas. De hecho, numerosos autores han discutido en las últimas décadas el movimiento hacia una forma organizacional más descentralizada en oposición a la organización chandleriana que predominó en el paradigma previo (Storpper 1989, Rulani, 2000; Langlois, 2003; Casalet, Cimoli y Yoguel 2005, entre otros).

Para que la descentralización productiva sea posible se requiere que unidades independientes operen sobre módulos también independientes que luego sean articulados de manera coherente, y que la articulación esté asegurada sobre la base un de compromiso previo. Ese compromiso será distinto ya sea el caso de mercado, jerarquía o red. En la coordinación a través del mercado el compromiso queda sujeto a la relación mercantil de compra-venta y en el caso de jerarquía, a las relaciones de subordinación al interior de la organización. Pero cuando se trata de una red o trama el compromiso queda sujeto a distintas formas contractuales que pueden regir en el marco de acuerdos de cooperación formales o informales. En el caso del SL el compromiso está determinado por un conjunto de *normas comunitarias*.

La descentralización productiva aplicada por el SL adquiere características especiales ya que no sólo no son sujeto de contratos o acuerdos formales sino que ni siquiera existen pautas de retribución alguna. El desarrollo de SL involucra miles de programadores en desarrollos tanto de gran como pequeña escala, donde el principio central de organización

es que el software permanece libre de la mayor parte de las limitaciones de copia y uso que tiene el software propietario. Nadie es dueño del software en el sentido tradicional de poder decidir cómo se desarrolla o utiliza. El resultado es la emergencia de una forma de colaboración productiva e innovativa, cuyos participantes actores no están organizados en firmas y donde sus decisiones no están guiadas por las señales de precios (Benkler, 2001).

Como se ha analizado en el capítulo previo, la aparición de Internet actuó como el elemento del contexto tecnológico determinante en la evolución del SL. A pesar de que los principios básicos del SL estuvieran ya presentes en los inicios de la computación, la gran explosión en su crecimiento estuvo asociada al desarrollo de Internet, que permitió acceder a una escala global de colaboradores y a una mayor velocidad de intercambio. Sin embargo, la sustentabilidad del modelo iba a requerir una transformación en el modelo de negocio –ya discutida en el capítulo 2- y en las formas institucionales de protección de la propiedad intelectual. Esta última sólo pudo ser posible bajo la preexistencia normas, reglas y pautas de comportamiento comunitario sustentadas, a su vez, en una jerarquía de valores fuertemente establecida. En resumidas cuentas, el cambio organizacional fue posible en la medida en que la “comunidad” preexistiera y sus reglas estuviesen fuertemente arraigadas entre sus miembros.

Por lo tanto, el compromiso que articula la descentralización productiva en el SL es el buen funcionamiento de la red colaborativa que está asegurado por la extensión mundial de la red y por la serie de principios de cohesión comunitarios. Estos principios están presentes en el discurso del SL construido a lo largo de su historia e incluso antes de que el fenómeno alcanzara identidad propia.

En conclusión, la comunidad preexistente aglutinada por una forma discursiva afianzada en casi de 20 años de historia, la aplicación de desarrollos modulares y la descentralización operativa a escala global garantizan la solidez del modo organizacional del SL y la superioridad tecnológica de sus resultados en términos de estabilidad y elegancia del software.

El próximo apartado está dedicado al estudio de los discursos organizacionales del SL.

3.2.2 Las bases antropológicas de la comunidad *hacker* base del cambio institucional

Podemos definir brevemente a la comunidad de SL como el conjunto de desarrolladores que aportan su tiempo y trabajo para la creación del SL, más las interacciones entre estos miembros de la comunidad. Estas interacciones están reguladas por normas específicas de comportamiento (tanto implícitas como explícitas) que permiten a la comunidad ser el sustento insustituible del cambio organizacional que representa el SL. Dichas normas, a su vez, se desprenden de un conjunto de discursos que forman la imagen que la comunidad tiene de sí misma, y que puede apreciarse en los “escritos fundadores” de la misma.

El primero de estos escritos es, sin duda, el *Manifiesto GNU* publicado por Richard Stallman en 1983. Como se ha mencionado en el capítulo 2, este documento es la primera formulación programática de una plataforma de software enteramente basada en SL. En él, Stallman carga de sentido político la decisión de crear software por fuera del entonces naciente sistema de licencias propietario. El acercamiento entre este documento y la ética kantiana pone de manifiesto qué idea tiene Stallman de sí mismo y sus colaboradores: altruistas que donan su trabajo para construir un bien común a toda la sociedad. Las repetidas menciones de sus renuncias pasados y futuros⁶⁰ acrecientan la idea del sacrificio de los intereses individuales en pos de cumplir con un mandato ético; idea que se completa con la noción del software propietario como “nocivo” y “destrutivo” de ese bien común. Aunque ya en el Manifiesto se habla de posibles modelos de negocios que incluyan al SL, son vistos como accesorios a la idea del SL en sí, y apenas útiles para cubrir las necesidades básicas de los programadores. Es fácil descubrir similitudes entre éste y otros discursos, anteriores y posteriores, surgidos del seno de la sociedad de consumo, como el movimiento hippie y el ecologismo. Esta visión, que reviste al SL de un sentido de importancia social y justificación ética, predominó en la comunidad durante

⁶⁰ Su renuncia al MIT y la aceptación de posibles bajas remunerativas.

los años 80 y principios de los 90s, y hasta nuestros días permanece arraigada en numerosos grupos de desarrolladores de SL.

Pero esta imagen de paladines nunca fue única dentro de la comunidad de SL. Desde el principio hubo quienes desestimaron las posturas abiertamente políticas por considerarlas ajenas a la actividad específicamente técnica de escribir software (los que Raymond llama “los pragmáticos”); el ejemplo más claro, como se dijo, fueron los seguidores de la licencia BSD (creada en 1979). Además de la tradición académica (a la que pertenecía Stallman) había una fuerte tradición “hobbysta” en los comienzos de la computación personal. Esta tradición aportó al software propietario sus empresarios más renombrados (los ejemplos más famosos: Bill Gates y Steve Jobs), pero también aportó el concepto de una comunidad de iguales, unidos por un mismo interés, que manejan el conocimiento de forma comunitaria. Estas simples economías donáticas del conocimiento que constituían versiones simplificadas y, sobre todo, no jerárquicas, de la institución académica, también forman parte de la historia de la comunidad de SL.

La tradición hobbysta pasaría a un primer plano con la llegada de Internet. En otro “escrito fundacional” de la comunidad, el mensaje enviado a Usenet por Linus Torvalds el 25 de agosto de 1991, que constituyó la primera aparición pública de Linux, Torvalds escribe: *“Estoy haciendo un sistema operativo libre para clones de AT 386 (es solamente un hobby, nada grande ni profesional como gnu).”* En esa sencilla oración se manifiestan dos elementos claves de la nueva ola de SL: a) para el hobbyista, el desarrollo de software es un fin en sí mismo; b) el proceso de desarrollo es informal: no existe un plan detallado de antemano, ni, por supuesto, fechas a cumplir, sino que el progreso del proyecto sigue los designios de su líder. El proceso informal, sumado a la imprescindible velocidad y alcance del nuevo medio, fueron las condiciones necesarias para que surgiera la comunidad del SL en la fisonomía que se le conoce hoy. Esta nueva modalidad de desarrollo, como ya se ha dicho, está descrita en *La catedral y el bazar*. Raymond es también autor de otro influyente ensayo sobre lo que él denomina “cultura hacker”: *Cultivando la noosfera*, y que también podemos considerar un documento formativo de la

comunidad de SL⁶¹. Este artículo propone un método antropológico para describir el funcionamiento de la comunidad, señalando tres “tabúes” de la cultura *hacker*:

- No bifurcar proyectos
- No distribuir modificaciones no autorizadas
- No remover el nombre de un colaborador de los créditos públicos que acompañan al software.

Raymond deduce que estas acciones atentarían contra la “teoría de la propiedad” (implícita hasta el momento del análisis de Raymond) de la comunidad; y la compara con las tradiciones del reparto de tierra en las colonias angloamericanas. La idea de propiedad vendría a sostener el funcionamiento de la comunidad de SL como una economía donática, en la cual los participantes ceden voluntariamente su trabajo a cambio del prestigio que ganan ante sus pares: el prestigio es observable en los “títulos de propiedad” que cada miembro exhiba. El propósito del ensayo es postular la motivación de ganar prestigio como el motor fundamental de la comunidad. Está claro que esta imagen de los colaboradores del SL contrasta fuertemente con la imagen altruista que promovía la FSF; el mismo Raymond, en la introducción, plantea su argumento como un desenmascaramiento ideológico, como un puente que cierre la brecha entre acción y discurso en la comunidad.

El argumento de Raymond es interesante y, sin duda, único al momento de su publicación (electrónica en abril de 1998, en libro en 1999), dado que en ese momento el fenómeno todavía no había suscitado demasiado interés de especialistas ajenos a la informática⁶². La cuestión de las motivaciones será analizada con detalle en el apartado 3.3 de este

⁶¹ Gran parte del crédito del que gozan los ensayos de Raymond se debe a que él mismo se considera miembro de esta cultura hacker, al igual que Stallman y Torvalds.

⁶² Como veremos más adelante (en el apartado 3.3.1), las hipótesis sobre las motivaciones que van a sostener las lecturas económicas sobre el fenómeno del SL, surgen de lo propuesto por Raymond.

capítulo, pero es necesario señalar aquí algunos problemas del análisis de *Cultivando la noosfera*. El primero es que la metáfora del bazar es, como observa Zeitlyn (2003), demasiado parecida en funcionamiento a la metáfora clásica de mercado, y por lo tanto no es buena para describir relaciones donáticas entre los participantes. Zeitlyn (2003) propone introducir el concepto de “afinidad sanguínea” (*kinship amity*) para mejor analizar las complejas relaciones entre los miembros de la comunidad en torno al concepto de “proyecto”, que funcionarían de forma parecida a clanes familiares.

La otra objeción que es necesario hacer es observar que Raymond utiliza demasiado libremente el concepto de economía donática (“*gift culture*”). En *Cultivando la noosfera*, afirma que “*Las culturas donáticas son adaptaciones no a la escasez sino a la abundancia. Surgen en poblaciones que no tienen problemas significativos de escasez material de bienes de supervivencia*” Luego afirma que dentro de la comunidad no hay necesidades básicas (equipamiento informático, conexión a la red, etc) insatisfechas, por lo que la “cultura hacker” es una cultura de la abundancia donde sólo el dar puede generar relaciones jerárquicas.

Sin embargo, no es cierto que las economías⁶³ donáticas sean economías de la abundancia. Para poner un ejemplo antropológico, Marvin Harris (1995) analiza en la evolución de economías donáticas en tres estadios:

- a. Un estadio estrictamente igualitario (el de los aborígenes !kung) donde la relación donática sirve para socializar el producto de la caza y, por lo tanto, evitar los riesgos de las fluctuaciones individuales del producto,
- b. Un estadio donde “cabecillas” (en el ejemplo, los indios semanis de Malasia) se destacan de sus pares al esforzarse más y ceder

⁶³ Raymond introduce una mezcla de categorías al poner en el mismo plano tres “formas organizacionales”: la jerarquía de comando, la economía de intercambio y la cultura donática. Esta confusión le permite obviar el problema de analizar específicamente el funcionamiento económico de lo que él denomina “cultura donática”. Para analizar esta contradicción, aquí se utilizará el término “economía donática” en lugar de “cultura donática”.

voluntariamente beneficios para obtener “prestigio” (aunque no poder real) sobre la comunidad, y

- c. Un estadio donde esos cabecillas han evolucionado en “grandes hombres”, donde sí hay una clara jerarquización política en base a la “generosidad” de estos grandes hombres. El ejemplo (que, casualmente, también menciona Raymond) es el potslach (festín) de los Kwakiutl.

Estos ejemplos nos permiten ver que también en las sociedades donde predomina la relación donática hay un recurso escaso que es necesario administrar. En el caso del SL, este recurso escaso es obvio: es el trabajo de los miembros de la comunidad. Considerando esto último, se pueden reinterpretar los tabúes que cita Raymond (especialmente el de no bifurcar proyectos) como medidas que evitan desperdiciar el trabajo de los miembros de la comunidad en esfuerzos duplicados; es decir, como comportamientos económicos que son esenciales a la supervivencia del SL. Desde este punto de vista, la naturaleza de la motivación personal para participar en el SL no es el determinante de las relaciones organizacionales que existen entre la comunidad y sus miembros.

Algunas preguntas relacionadas con un problema de jerarquías institucionales pueden plantearse a esta altura de la investigación: ¿es posible que las reglas, valores, hábitos de una sociedad capitalista, gobernada por la propiedad privada, puedan ser "jerárquicamente" dominadas por reglas de una comunidad, sea de trueque, de *hackers* o de *hippies*? ¿Es posible alcanzar autonomía institucional a otro nivel que (al menos) del Estado-Nación⁶⁴? Resumiendo, la pregunta clave es ¿cuál es la capacidad del movimiento del SL para imponer nuevas formas institucionales que trasciendan la esfera comunitaria? Para poder dar respuesta es necesario, en primer lugar, entender al surgimiento del SL no como una consecuencia de un conjunto de voluntades virtuosas sino como una respuesta natural al nuevo contexto tecnológico y organizacional; en segundo lugar, indagar en la

⁶⁴ Siendo que frecuentemente se plantea que las instituciones nacionales están subordinadas a una lógica global.

interacción entre normas e instituciones de esta comunidad y las de la sociedad capitalista en donde está inserta, y, en tercer lugar, poder determinar si además de las pautas discursivas que dan coherencia a la comunidad existe una acción concreta y una “materialidad” de esa acción, que actúe como vínculo real entre ambos mundos.

Distintos autores (Barlow 1994; Raymond, 1999, Stallman 1998, Lessig 2001) vienen sosteniendo desde la década del 80 que la forma de protección de las rentas económicas que genera el conocimiento través de los derechos de propiedad intelectual en su definición tradicional pierde vigencia en distintas disciplinas y áreas del conocimiento. Para Barlow (1994) esto se debe a que el cambio tecnológico, producto de la emergencia de las tecnologías informacionales, ha redefinido el rol de la información y el conocimiento en la sociedad y en la producción y la forma en que se presenta como bien económico. De acuerdo con este autor, no solo logró mayor protagonismo, sino que su forma se volvió más etérea y meliflua, al circular a velocidades insospechadas por redes de fibra óptica. Barlow afirma que, *“el canon acumulado del copyright y la ley de patentes, se creó para transportar formas y métodos de expresión completamente distintos de la vaporosa carga que ahora se le pide que lleve”* y que *“la legislación de propiedad intelectual no se puede remendar, adaptar o expandir para que contenga los gases de la expresión digitalizada (...). Tendremos que desarrollar un conjunto completamente nuevo de métodos acorde con este conjunto enteramente nuevo de circunstancias”*.

Estos autores ponen especial énfasis en que el cambio tecnológico se ha adelantado al cambio institucional y reclaman nuevas formas de protección. La hipótesis de esta investigación es sutilmente distinta, porque reconoce que el defasaje no sólo se da entre cambio tecnológico e institucional sino también con relación al cambio organizacional. De acuerdo con lo analizado en el capítulo 2 y el apartado 3.2.1 del presente capítulo, los mecanismos de protección diseñados hace más de tres siglos no sólo resultan ineficientes (como postulan Barlow y Lessig, entre otros) sino incluso contraproducentes para el desarrollo de la actividad innovativa.

El fenómeno del SL resulta interesante porque plantea que la ausencia de formas de legales de protección implica que no sea posible la apropiación económica de las rentas

que genera el conocimiento. Si se considera al SL como un bien club, donde los integrantes del club pueden hacer uso económico del código libre en modelos de negocios basados en servicios y sustentados en capacidades cognitivas diferenciales y competencias específicas, puede notarse que no solo la apropiación económica es posible sino que además la forma que adopta es compatible con la forma organizacional basada en la interacción en red y permite maximizar los aprendizajes que surgen de la interacción y una apropiación colectiva.

3.3 La cuestión de las motivaciones

3.3.1 La visión de la economía convencional

La mayor parte de los trabajos sobre la economía del SL buscan dar una explicación sobre la cuestión de las motivaciones; es decir, dar una respuesta a la pregunta de por qué agentes racionales se involucran en proyectos de desarrollo que implican un compromiso sostenido e intenso en una actividad no contractual y no remunerativa. Por qué estas personas y organizaciones contribuyen gratuitamente al desarrollando código fuente libre y renuncian a la posibilidad de apropiarse la renta monopólica que la legislación de propiedad intelectual prevé.

El artículo de Lerner y Tirol (2002)⁶⁵ fue uno de los primeros trabajos que intentó dar una explicación económica del fenómeno del SL. En este trabajo se sintetiza lo que puede

⁶⁵ Por razones de extensión, en este apartado se expondrá uno de una serie de artículo que tienen una aproximación similar. Entre otros pueden mencionarse el de Baldwin y Clark (2002) que busca las “*transacciones*” en la economía del SL -o sea qué obtienen a cambio los programadores que dedican tiempo al desarrollo de un bien público-; o el trabajo de Hann, et al (2002), que afirma que los “*retornos (monetarios) por la participación en proyectos libres tardan, pero llegan*”. Dentro de la línea del “*career concern*”, de acuerdo con estos autores, las contribuciones al SL se traducen en mayor prestigio y éste en mejores oportunidades laborales y mayores salarios. Un enfoque similar es el utilizado por Varian (2001), que demuestra que un sistema que se sustente en la provisión puramente voluntaria nunca puede ser confiable, por los inconvenientes de la conducta “*free rider*”.

catalogarse como el enfoque neoclásico sobre la economía del SL. Su explicación sobre las motivaciones surge del análisis convencional de la maximización de la utilidad individual del *hacker*. Utilizan elementos de la literatura sobre *career concerns*⁶⁶, para explicar algunas características de los proyectos de desarrollo de SL, y la noción de *user-driven innovation*, de la teoría de la innovación (Rosenberg, 1976), para explicar cómo el rol de los usuarios puede acelerar el proceso de innovación.

El trabajo de Lerner y Tirol parte de la premisa de que los programadores maximizan racionalmente sus funciones de utilidad individuales en el momento de decidir qué contribución hacer al SL: en cuántos y cuáles proyectos colaborar y cuántas horas semanales dedicarle. La decisión de participar en desarrollos libres está afectada por diversos factores. De acuerdo con estos autores *“un programador decide participar en un proyecto (libre o propietario) solo si obtiene un beneficio neto positivo, definido en un sentido amplio. Estos beneficios están compuestos por los pagos inmediatos (beneficios corrientes menos costos corrientes) y por los pagos mediatos de la participación”*.

En referencia a los beneficios y costos inmediatos se hallan las compensaciones monetarias por su trabajo de programador en desarrollos comerciales, los costos de oportunidad asociados de destinar tiempo al desarrollo de programas libres y los beneficios que se obtienen de reparar un *bug* o customizar un programa a sus propios requerimientos, es decir los beneficios derivados del autoconsumo. Los autores utilizan el concepto de *user-driven innovation* para explicar la importancia para los programadores de introducir mejoras sobre sus herramientas de trabajo. Los programadores son usuarios y desarrolladores al mismo tiempo, lo que les permite ajustar el producto a sus necesidades específicas. En consecuencia, el SL pocas veces está bien documentado o tiene interfaces amigables para usuarios no expertos. De acuerdo con Nadeau (1999), un desarrollador de SL, *“cada versión que Microsoft lanza de sus productos, siempre escucha al más ignorante de sus clientes. Esta es la clave para simplificar cada versión*

⁶⁶ La bibliografía sobre *career concerns* surge como una rama de la teoría de los incentivos dentro de la organización industrial. Los trabajos sobre *career concerns* sugieren que muchos de los incentivos en organizaciones no se presentan a través de incentivos explícitos en contratos formales sino implícitos y vinculados con la preocupación por la carrera profesional (Dewatripont et al. 1999).

del software para seguir conquistando al público no usuario. Los desarrolladores de Linux y OS/2, por su parte, tienden a escuchar a sus clientes más inteligentes...El bien que hace Microsoft al acercar la computación a no usuarios termina por perjudicar a los usuarios expertos". Efectivamente, en sus comienzos el SL ha tenido una mayor difusión en ambientes técnicos y especializados ya que estos tienden a valorar más las posibilidades de *setear* el programa a sus propios requerimientos o modificar el código de acuerdo a sus necesidades que tener una linda interfase gráfica. En este sentido, la función de utilidad de los desarrolladores de la comunidad, que consideran Lerner y Tirol en su artículo, entraría también la satisfacción derivada de contar con una herramienta apropiada para su actividad no disponible en el mercado.

Dentro de los beneficios mediatos, Lerner y Tirol señalan dos cuestiones que se entrelazan y parecen difíciles de diferenciar: la preocupación por la carrera profesional y la gratificación del ego. Contribuir en desarrollos libres otorga prestigio en la comunidad de desarrolladores y en el mercado de trabajo en general, por lo que haber participado en proyectos de desarrollo libres exitosos está asociado a mejores oportunidades laborales. Otras veces el prestigio y el reconocimiento de los pares son valiosos en sí mismos. Lerner y Tirol subrayan una cita de Raymond donde afirma que *"la 'función utilidad' que los hackers de Linux están maximizando no es económica en el sentido clásico, sino algo intangible como la satisfacción de su ego y su reputación entre otros hackers. (Uno podría hablar de su "motivación altruista", pero ignoraríamos el hecho de que el altruismo en sí mismo es una forma de satisfacción del ego para el altruista). Los grupos voluntarios que funcionan de esta manera no son escasos realmente; uno en el que he participado es el de los aficionados a la ciencia ficción, que a diferencia del mundo de los hackers, reconoce explícitamente el 'egoboo' (el crecimiento de la reputación de uno entre los demás) como la motivación básica que está detrás de la actividad de los voluntarios"*.

La hipótesis de Lerner y Tirol sobre las motivaciones no se aparta en lo más mínimo de la conducta del agente representativo racional de los libros de microeconomía básica: la maximización de la utilidad; la única diferencia es que la función de utilidad en este caso está definida en un "sentido amplio". De acuerdo con estos autores, si la conducta de los desarrolladores parece no-racional es porque en la construcción de la función de utilidad

no fueron considerados factores que, a pesar de que no tengan un pago monetario inmediato, resultan cruciales para tomar de decisión de participar (o no) en desarrollos libres.

Bajo las hipótesis de racionalidad instrumental e información perfecta, los individuos se comportan dentro el supuesto habitual de conducta maximizadora y la función de utilidad supuesta incorpora los elementos de gratificación del ego, pagos futuros (*career concerns*) y autoconsumo (*user-driven innovation*). De acuerdo con la hipótesis de información y racionalidad perfecta, los miembros de la comunidad *hacker* que participan en desarrollos libres conocen todas las oportunidades laborales actuales y las que puedan surgir en el futuro en el caso de participar en cada uno de los diferentes proyectos de desarrollos libres⁶⁷, pueden ponderar el pago monetario de cada caso con la gratificación personal por reconocimiento de los pares y de disponer de una herramienta de trabajo apropiada, ordenar la utilidad que les reportaría cada decisión y comparar con la desutilidad que les significaría dedicar parte de su ocio al desarrollo de un bien público y sobre ese balance toman una decisión acertada y racional sobre qué contribución realizar.

Puede reconocerse que este enfoque presenta algunos elementos que permiten comprender el fenómeno desde una perspectiva económica. Quizás las hipótesis de información y racionalidad perfecta sean demasiado radicales, sin embargo permitieron modelar casos, donde incorporando las variables adecuadas, la conducta de los agentes de participar en desarrollos libres iba más allá del simple altruismo que pregonaba el discurso inicial de la FSF. Sin embargo, la lectura de estos autores no se apartó lo suficiente de la hipótesis de Raymond de satisfacción del ego, sin advertir, como ya se ha señalado, que la *Catedral y el Bazar* contribuyó a la evolución del discurso más que a desenmascarlo. Por lo tanto, este primer enfoque dejó sin responder un conjunto de interrogantes que demuestran que la complejidad del fenómeno requiere de explicaciones más generales que den cuenta de sus múltiples facetas. Por ejemplo, cómo se explica el hecho de que cada vez sean más los programas libres para usuarios finales preparados

⁶⁷ Es decir, que pueden elegir entre los alrededor de 54.000 proyectos que alberga SourceForge (por mencionar sólo un repositorio de código fuente libre) y detectar cual de ellos es el más prometedor para su futura carrera profesional.

con interfases gráficas y una instalación fácil que no requiera conocimientos de informática, o que grandes empresas de software y hardware participen en el desarrollo de fuente abierta y/o “*abran*” código de sus programas propietarios. Incluso la existencia de empresas dedicadas al desarrollo de software libre compitiendo en el mercado. Los trabajos dentro de esta línea teórica pueden responder por las motivaciones de los programadores pero no por las empresas y organizaciones que participan en estos proyectos.

3.3.2 Algunas regularidades empíricas sobre la cuestión de las motivaciones

Los trabajos empíricos sobre el desarrollo de software libre a nivel internacional y local⁶⁸ permiten contrastar algunas de las ideas referidas a la cuestión de las motivaciones. Pueden encontrarse algunos elementos comunes entre estos trabajos que describen las motivaciones de los agentes. Se considera que estas regularidades empíricas constituyen un buen punto de partida para reelaborar las hipótesis tradicionales de las motivaciones referidas al ego y la reputación.

Básicamente, el desarrollo de competencias al igual que el sentimiento de pertenencia a una comunidad parece jugar un rol central en la decisión de participar de desarrollos libres. Sin embargo, las ganancias monetarias (ya sean ganancias directas por desarrollar SL o vía reputación y desarrollo de la carrera profesional) están presentes en la mayor parte de los desarrolladores.

⁶⁸ Los trabajos considerados fueron las encuestas FLOSS, BCG y UNGS-SADIO (para mayor detalle sobre estas encuestas ver anexo metodológico). Cabe aclarar que la encuesta UNGS-SADIO, a diferencia de las otras dos, tuvo el objetivo de caracterizar las competencias tecnológicas de los trabajadores informáticos, por lo tanto fueron incluidos en la muestra tanto desarrolladores de SL, como usuarios y otros trabajadores informáticos que no utilizan SL. No obstante, un apartado de cuestionario estuvo dedicado a recabar información sobre el proceso de difusión de SL entre estos trabajadores. Ver anexos por detalles de cada una de estas encuestas.

Algunas de las conclusiones del estudio FLOSS acerca de las motivaciones son que el 78,9% de los desarrolladores de programas libres o abiertos afirma que dentro de sus motivaciones principales para comenzar a desarrollar código libre ocupó un lugar central la de aprender y desarrollar nuevas habilidades y el 40,8% comenzó con esta tarea para compartir su conocimiento. Solo el 4,4% empezó a desarrollar SL con el objetivo de hacer dinero⁶⁹. El 70,5% afirma que entre sus motivaciones para continuar a desarrollando código libre estuvo aprender y desarrollar nuevas habilidades, mientras que el 67,2% lo hace para compartir sus conocimiento. Mientras que el porcentaje de encuestados que continuó por motivos monetarios es el 12,3%. A pesar de esto, más del 50% de los encuestados gana dinero de alguna forma a partir del desarrollo de SL.

El trabajo realizado por el BCG basado en una encuesta a cerca de 500 desarrolladores identifica cuatro grupos de desarrolladores que se diferencian entre sí por las motivaciones principales de cada caso. De acuerdo con este trabajo los grupos serían:

- Los creyentes (*believers*) cuya principal motivación es que según ellos el software debe ser libre. Este grupo está compuesto por el 33% del panel y tienen como segunda motivación principal desarrollar capacidades.
- Los que buscan mejorar sus capacidades (*skill enhancers*), este grupo está compuesto por 21% de la muestra (a pesar de que la razón mejorar las capacidades fue señalada como una motivación principal por el 43% del muestra). La segunda motivación principal de este grupo es el estímulo intelectual de participar en estos desarrollos.
- Los que buscan divertirse (*fun seeking*), 25% de la muestra, entre ellos las motivaciones centrales son la estimulación intelectual, porque resulta divertido, para trabajar en equipo y para mejorar sus capacidades. Cabe destacar que dentro

⁶⁹ Cabe mencionar que los encuestados podría marcar hasta cuatro motivaciones principales, por ello los porcentajes no suman 100%.

de la encuesta la estimulación intelectual como motivación principal recibió el 43% de las respuestas.

- Los que hacen como una tarea profesional (*professionals*) son el 21% de la muestra, y sus motivaciones son hacerlo porque es trabajo, por el estatus profesional y por el estatus que les significa trabajar en la esfera del software libre. El 30% del panel señaló como motivación principal que se trataba de un trabajo.

Aquí nuevamente puede señalarse la importancia de la auto capacitación a la hora de decidir participar en el desarrollo de SL. Tanto el grupo de los *skill enhancers* como de los *professionals* aparece la cuestión de la formación como algo central, mientras que el los otros dos grupo si bien no es central, aparece como segunda motivación principal. Esto muestra por una parte que las motivaciones no son únicas y predecibles sino diversas y que responden a cuestiones vinculadas con el interés personal pero también con cuestiones políticas como prima en el grupo de los *believers*. Por otra parte, muestra la vinculación entre el desarrollo de capacidades y el desarrollo de SL.

Por último la encuesta realizada a desarrolladores argentinos reafirman estas observaciones. Entre las personas que desarrollan SL desarrollar habilidades apareció como motivación principal (primeras tres) el 21% de las veces. En el segundo y tercer puesto estuvo el compartir conocimiento y mejorar las oportunidades laborales que fueron mencionadas 12% de las veces cada caso. Por otra parte el 71% de los encuestados que desarrollaban SL, afirmó que ganaba dinero con su actividad en el campo del SL, directamente (al 52% de los que desarrollan SL, les pagan por hacerlo o por proveer servicios asociados) o indirectamente (el 19% de este grupo).

En síntesis, todos estos trabajos demuestran que la cuestión de las motivaciones es compleja. La construcción de capacidades aparece como un punto central aunque la retribución económica no es marginal.

3.3.3 Breve digresión sobre la hipótesis de racionalidad

La cuestión de las motivaciones puede verse emparentada con la noción de racionalidad. En la teoría económica se asume que: los agentes se comportan de acuerdo con sus objetivos, dada la hipótesis de racionalidad utilizada. La economía convencional prevé un ajuste perfecto entre comportamiento y objetivos ya que asume que los agentes tienen racionalidad perfecta; es decir, el agente representativo neoclásico se *mueve* en tal dirección porque su objetivo es *ir* en esa dirección y su apreciación de la realidad no lo engaña. Diferentes corrientes heterodoxas de la teoría económica suavizan esta relación al asumir hipótesis de racionalidad limitada (Simon, 1957). Entonces, la dirección que toma un agente está asociada a su objetivo, dada la información que dispone y su capacidad para sintetizarla. No obstante estos enfoques se asemejan en que el contexto en el que los agentes toman decisiones está libre de instituciones y relaciones sociales que trasciendan a la relación mercantil. Es decir, se relacionan entre sí en una economía pura de mercado.

En el marco de esta investigación se propone la utilización de otra hipótesis de racionalidad: la racionalidad comunicativa (Habermas 1988, Johnson y Lundvall, 1994). Según esta hipótesis la racionalidad de los miembros de la comunidad SL está condicionada por su relación individual con los otros miembros y por su relación con la comunidad en general. Esta hipótesis de racionalidad implica que un análisis de la construcción individual de las motivaciones no alcanza para poder comprender el comportamiento de los individuos, ya que el ambiente en el que se desenvuelven existen instituciones que establecen normas y reglas de comportamiento que se manifiestan en la forma discursiva discutida en el apartado 3.2.2.

En la discusión que se presenta en el siguiente apartado sobre creación de competencias y bienes club se utiliza esta hipótesis de racionalidad.

3.3.4 SL, conocimiento y procesos de aprendizaje. El SL como un bien club

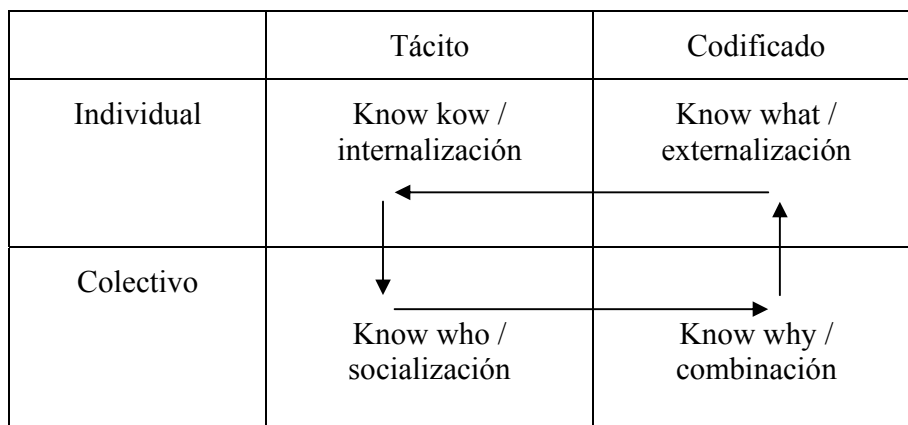
La cuestión de las motivaciones puede ser abordada desde una perspectiva que considere el contexto tecnológico e institucional en el que se desenvuelven los agentes. El avance de la economía del conocimiento y las particularidades del conocimiento y la información en tanto bienes económicos provocan, como ya se ha destacado, cambios en las formas organizacionales e institucionales. En consecuencia, cuestiones como la distinción entre conocimiento tácito y codificado y transformaciones y complementariedades⁷⁰, la dinámica entre difusión y apropiación de conocimiento (Yoguel et al, 2005) y desarrollo de formas organizacionales, son claves a la hora de explicar las motivaciones a participar en desarrollos libres.

En apartados anteriores se mencionó que el uso y desarrollo de SL está asociado a la creación y circulación del conocimiento. A diferencia del esquema propietario, el SL permite generar y amplificar procesos de creación de conocimiento en tanto la tecnología no se presenta como un paquete cerrado sino que existen posibilidades de modificación, adaptación, y, por lo tanto, aprendizaje, en el marco de una comunidad de desarrolladores y usuarios que colaboran en el proceso. De esta forma, la creación de capacidades tiene dos fuentes, por un lado, el ensayo y error en las tareas de desarrollo y, por el otro, en las interacciones que se dan en el interior de comunidad de SL. Es decir, *learning by doing* (aprender haciendo), cuando el programador se enfrenta al código escrito por otras personas para modificarlo y adaptarlo a sus propias necesidades, y *learning by interacting* (aprender interactuando), cuando el programador se pone en contacto con otros programadores de la comunidad para participar de procesos de desarrollo conjuntos. Estas fuentes de aprendizaje están relacionadas con dos tipos de saberes diferentes el *know how* y el *know who*. El primero, saber cómo, es individual y se refiere a las capacidades concretas construidas a partir de la experiencia y trayectoria tecnológica

⁷⁰ Explorada por diferentes autores (Lundvall, 1994; Antonelli, 1997 y 1999, Cowan et al 2000; Malerba y Orsenigo, 2000; Johnson et al, 2000; Nonaka y Takehuchi 1995; entre otros).

personal. El segundo, saber quién, incorpora la dimensión colectiva e implica poder identificar a los actores de la comunidad con saberes específicos. Este último, permite que el conocimiento no quede limitado a las capacidades individuales sino de la comunidad en su conjunto. Las herramientas concretas de comunicación de la comunidad dan forma a estos procesos de aprendizaje. Por ejemplo, la participación de foros de discusión instrumentados por la comunidad de desarrolladores permite compartir experiencias y coordinar acciones conjuntas de desarrollo y de investigación. Para poder articular estos dos tipos de saberes se requieren umbrales mínimos en otros dos, el *know why* y *know what* (saber porqué y saber qué). Estos últimos están relacionados con el conocimiento científico y tecnológico y se desarrollan a partir de la formación en ciencia base. La distinción entre los cuatro tipos de saberes mencionados fue introducida por Lundvall (1994) al combinar las formas tácita y codificada del conocimiento con las dimensiones individual y colectivo de formación de competencias. El *know how* aparece entonces como el conocimiento tácito-individual, es decir aquel desarrollado sobre la base de la experiencia y difícilmente transferible: se trata esencialmente de un conocimiento local y específico al individuo. El *know who* es el conocimiento tácito-colectivo, que puede construirse sobre la base de interacciones y por lo tanto es específico a un grupo (organización o comunidad). Las formas codificadas son el *know why* y *know what* codificado-colectivo y codificado-individual, respectivamente, en estos casos se trata de información de carácter general y particular pero fácilmente transferible, dada su forma codificada.

Figura 3.1 Tipos y metabolismo de conocimiento



Fuente: Elaboración propia sobre la base de Nonaka y Takehuchi (1995)

La mayor parte de la bibliografía sobre conocimiento enfatiza la distinción entre estas dimensiones, tácitas y codificadas, y explican el proceso de desarrollo de competencias como una continua transformación entre ellas⁷¹. Parten de la base de que, el conocimiento tácito es la fuente de diferenciación y por lo tanto de creación de ventajas competitivas, en la medida en que es altamente localizado y de difícil imitación. Mientras que el conocimiento codificado está disponible en manuales, descripción de procedimientos y diferente documentación a la que sólo puede restringirse el acceso mediante el secreto o trabas legales⁷². El proceso por el cual se desarrollan ventajas competitivas tiene que ver con una continua transformación de una forma de conocimiento en otra a partir de los procesos de socialización, externalización, combinación e internalización, lo que presupone que las transformaciones de tácito a codificado son posibles. Otros autores⁷³, por su parte, enfatizaron la existencia de lógicas subyacentes en el conocimiento tácito y codificado que condicionan las posibilidades de transformación de una forma a otra y por lo tanto de difusión y apropiación. Esto permitió ampliar el concepto de *creación de conocimiento* al de *desarrollo de capacidades cognitivas* que dependen más de la combinación entre ambos tipos de conocimiento más que la transformación de una forma a otra. De acuerdo con estos autores, no todo el conocimiento tácito es codificable ya que para las actividades de codificación y decodificación son necesarias capacidades previas que son necesariamente tácitas (Yoguel et al, 2005)

Cowan, David y Foray (2000), desarrollaron la noción de “libro de códigos desplazado”, que alude a un conocimiento altamente codificado que, para algunos observadores externos al grupo no es comprensible y es percibido como tácito que se asocia a la noción de comunidades epistémicas, donde está presente un entendimiento común guiado por la

⁷¹ Por ejemplo Lundvall, 1994; Antonelli, 1997 y 1999, Johnson et al, 2000; Nonaka y Takehuchi 1995.

⁷² Por ejemplo, las patentes pueden ser entendidas como conocimiento codificado del que solo puede hacer uso los propietario de la invención.

⁷³ Cowan et al (2000), Ancori et al (2000) y Nootboom (1999); y por el lado de las ciencias cognitivas, Bloch, (1991), Goody (1977) y Schank y Abelson, (1977).

memoria colectiva, convenciones o una autoridad procedural que combina las distintas porciones de saberes de los individuos de la comunidad. En esta perspectiva, lo que aparecería como codificado es de acceso limitado por los requerimientos de conocimientos tácitos previos y lo que aparece como tácito puede ser codificado a través de reglas no escritas. La circulación del conocimiento (pseudo tácito y tácito) generado en esa comunidad es libre sólo en su interior, lo que permite asumir al conocimiento como un bien club (Yoguel et al, 2005).

La comunidad de SL puede entenderse como una comunidad epistémica ya que el conocimiento que circula al interior de la red y el proceso de creación no queda limitado a una persona sino que pertenece al conjunto de la comunidad. A pesar de que a simple vista pueda suponerse que predominan las formas codificadas de conocimiento⁷⁴, están presentes elementos tácitos imprescindibles para la decodificación y un entendimiento común (reglas y autoridad procedural) que dan cohesión a la comunidad. Las capacidades cognitivas de los agentes, construidas a lo largo de senderos personales en interacción constante con la comunidad, les permite decodificar el enorme raudal de información que circula y está disponible en Internet. En este caso, la noción de circulación de conocimiento al interior de la red no es una metáfora sino que implica un espacio concreto Internet por lo que la información que circula está altamente codificada pero la forma de exclusión no es ni el secreto ni las trabas legales sino el umbral mínimo de capacidades.

En consecuencia, la comunidad no tiene ingreso irrestricto, la participación en proyectos de desarrollo libres implica la construcción de un espacio cuasi-público donde un umbral de mínimo de capacidades cognitivas requeridas para la decodificación funcionan como barreras a la entrada. El fenómeno puede ser descrito desde la noción de comunidades epistémicas y libros de códigos desplazados porque, por una parte, el conocimiento no es un bien público puro (no rival y no excluible), y por otra, la creación y circulación de conocimiento es intensa dentro de la comunidad pero opaca para los que permanecen

⁷⁴ En definitiva se trata de desarrollo de código fuente y la comunicación se establece de forma electrónica por lo que predominan los mensajes de texto más que los orales.

fuera de ella. En síntesis, ingresar a la comunidad implica participar de estos procesos de creación de conocimientos, y sus consecuencias son por un lado poder hacer uso del código ya desarrollado pero también aprender del mismo.

La auto denominación de *comunidad* por parte de estos grupos demuestra la importancia que le otorgan sus miembros al sentido de pertenencia. En tal sentido, resulta insuficiente afirmar que la motivación puede circunscribirse a una cuestión material, aunque también resulta insuficiente una explicación netamente altruista, discutida incluso por el mismo Raymond. Sobre todo a partir de la fuerte difusión del fenómeno y la aparición de grandes actores económicos en escena.

La tesis de este trabajo sobre las motivaciones se encuentra a mitad de camino entre el altruismo y la búsqueda del beneficio económico o el ego personal. El error que cometen las hipótesis sobre motivaciones guiadas por el altruismo y el beneficio económico está en considerar al SL como un bien público. Los reservorios de código libre que pueden encontrarse en la web no pueden pensarse como bienes públicos porque no verifican las dos cualidades que definen a este tipo de bienes: consumo no rival y no excluible. Se trata de un bien cuyo consumo es no rival, ya que no sufre desgaste alguno con el uso y la copia digital implica su reproducción casi sin límites a costos ínfimos. Sin embargo no está tan claro que su consumo sea no excluible. Si bien en términos hipotéticos cualquier persona (con acceso a la red) tiene a su disposición el inmenso reservorio de código abierto y libre, las posibilidades de hacer un uso efectivo de este material están limitadas a las personas que tengan la capacidad para manipular, adaptar, compilar y poner en funcionamiento los programas para computadoras que ese código encierra. Evidentemente estas capacidades constituyen una barrera ineludible al uso efectivo de este supuesto bien público.

En este sentido, cabe reformular la idea de que el SL como bien público por la del SL como bien club, en donde para acceder al *club* hace falta contar con los conocimientos necesarios. La visión de la comunidad de desarrolladores de SL es compatible con esta noción de bien club.

Los aportes que los desarrolladores realizan al corpus de código libre pueden ser leídos como una contribución a la construcción un espacio cuasi-público desde el cual pueden

extraer y devolver constantemente. Quienes pertenecen al club son quienes están capacitados para hacer un uso económico de este material y por lo tanto probablemente (aunque no obligatoriamente) retribuyan con aportes personales (Zeitlyn, 2003). Las retribuciones son fundamentales para la supervivencia del sistema, como se ha comentado en el apartado 3.2.2, quedan garantizadas en el conjunto de pautas discursivas y tabúes que genera el movimiento.

La dinámica de aportes y usos no está liderada por una lógica de reciprocidad o intercambio, como plantea Raymond y el artículo de Lerner y Tirol, en realidad entran en juego cuestiones tales como compartir conocimientos y habilidades, obtener reconocimiento personal, la posibilidad de realizar desarrollos conjuntos, etc. No existe una tasa de cambio entre horas de programación invertidas en SL y la cantidad de código abierto del que se pueda hacer uso; o entre horas de *debuging* de código ajeno y aprendizaje y desarrollo de habilidades o gratificación del ego, ni una tasa de sustitución en una supuesta función de utilidad del *hacker* de la que habla Raymond.

Por una parte, los estudios empíricos demuestran que es difícil establecer un orden de prioridades y que las motivaciones pueden ser tantas como individuos haya en la comunidad ya que es ante todo una cuestión personal. Pero, por otra, la trascendencia de la existencia del ‘otro’ y de la comunidad en la labor anulan posibles especulaciones individuales del tipo costo beneficio.

Los miembros intercambian conocimiento pero no siguiendo una lógica de maximización de utilidad sino de contribuir al crecimiento de la comunidad y su diferenciación con el resto. Persiguen la consolidación de la comunidad porque ahí encontrarán el mayor beneficio colectivo, y la mayor economía de los esfuerzos de la comunidad, aunque el comportamiento no sea compatible con la búsqueda de una satisfacción personal individual (Zeitlyn, 2003). De esta forma, el conocimiento se transforma en un bien económico.

A pesar de que este Raymond insista en que la economía del SL es una economía del ego, el análisis que hace de las motivaciones de los desarrolladores voluntarios de SL⁷⁵, es coherentes con la idea de una valoración mediata de la labor; la importancia de la reputación, el comportamiento de la comunidad y la existencia de reglas y tabúes dentro de ella. Raymond hace una buena descripción del funcionamiento de la actividad, porque es parte de esta comunidad y uno de los primeros en reflexionar sobre la misma, sin embargo el análisis que hace sobre el fenómeno del SL no puede ser leído ingenuamente sino como un giro adicional, como un *aggiornamento* a la etapa post-internet del discurso original de la FSF.

En Cultivando la Noosfera, el autor analiza la cultura hacker comparándola con las relaciones donáticas o de intercambio puro. De acuerdo con Raymond “*queda muy claro que la sociedad hacker de código abierto es de hecho una cultura de regalos*”, dentro de ella y el software es compartido gratuitamente. Define a la Noosfera como el territorio de las ideas o el espacio de los proyectos software, donde los *hackers* cultivan sus propios proyectos y contribuyen en los desarrollos de otros. Las reglas que regulan este espacio, cómo se realizan las contribuciones, cómo se establecen los roles y se seleccionan los proyectos, son, en definitiva, reglas implícitas del club.

En conclusión, las motivaciones son siempre personales y por lo tanto múltiples, entonces, otorgarle a la cuestión de la motivación un rol tan central en la explicación del fenómeno puede ser un error que se deriva de la hipótesis de racionalidad perfecta. La pregunta clave sobre la economía del SL está en cómo esta comunidad ha desarrollado una forma discursiva que combine exitosamente las tecnologías informacionales con formas organizacionales e institucionales que superen las limitaciones en el ritmo de innovación del modelo tradicional de desarrollo de software.

3.4 Síntesis y conclusiones

⁷⁵ Cultivando por la Noosfera, La Catedral y el bazar o Una breve historia de la cultura *hacker*.

Este capítulo estuvo dedicado a entender el fenómeno del SL desde una visión evolutiva de las formas organizacionales, tecnológicas e institucionales y a abordar la cuestión de las motivaciones desde una perspectiva original, que discute las hipótesis convencionales de altruismo o satisfacción del ego.

Este capítulo aporta una lectura innovadora sobre las motivaciones al contemplar el comportamiento de los individuos en un ambiente institucional y organizacional complejo y al relacionar su decisión de participar en desarrollo libre con la creación de competencias. Este contexto está caracterizado por la utilización de herramientas de comunicación digital, la interacción en redes horizontales y la presencia de comunidades epistémicas o clubes definidos por el grado de desarrollo de capacidades cognitivas.

En este sentido, se demuestra que las motivaciones están asociadas a diferentes aspectos pero que el discurso del SL establece reglas de comportamiento que aseguran la subsistencia del mismo la comunidad y del SL. El ingreso a la comunidad queda determinado por el desarrollo de capacidades cognitivas (Yoguel et al 2005) y habilidades específicas, mientras que la colaboración entre pares depende de la existencia de un conjunto de pautas éticas comunes. Los conceptos de comunidad epistémica y relaciones donáticas (Maus, 1960; Zeitlyn, 2003) y capital simbólico (Zeitlyn, 2003), así como la hipótesis de racionalidad comunicativa (Habermas 1988, Johnson y Lundvall, 1994) contribuyen a explicar el comportamiento de la comunidad de desarrolladores de una forma compleja e integradora.

4 EL PROCESO DE DIFUSIÓN DE SL EN LA ARGENTINA. UNA APROXIMACIÓN TEÓRICA

4.1 El SL en los países en desarrollo: la relevancia del debate

A lo largo de los capítulos anteriores se ha tratado la problemática del SL desde una perspectiva global. Como se mostró en el segundo capítulo, el movimiento del SL surgió en los países desarrollados (PD) y sus principales actores (desarrolladores como Stallman, Raymond, Torvals e instituciones como la FSF, Creative Commons, etc) y promotores están localizados en aquellos países. De acuerdo con las estadísticas más recientes, el porcentaje de contribuciones realizadas por programadores de países en desarrollo (PED), en general, y por argentinos, en particular, es muy pequeño⁷⁶. En este sentido, una mira superficial puede dar la impresión de que se trata de un fenómeno ajeno a la realidad local. Sin embargo, no por esto debe pensarse que carece de relevancia un análisis desde los países en desarrollos.

Actualmente, muchos de los PED están desarrollando sus infraestructuras informáticas, generando una industria local del software, o simplemente difundiendo el uso de las nuevas tecnologías en diversas áreas de la sociedad. En este contexto de fuerte expansión de las nuevas tecnologías, la elección de proveedores y estándares es estratégica; ya es clave en el proceso de transición hacia el nuevo paradigma tecno-productivo (Katz 2003).

Muchos PED se han movido hacia el SL, y si bien claramente no han comenzado por el desarrollo de programas libres, sí cuentan con importantes tasas de utilización (en algunos casos promovida desde el sector público), que probablemente derivarán en

⁷⁶ 6% para los países en desarrollo y 0.3% para la Argentina (FLOSS, 2002). Ver más estadísticas en el capítulo siguiente.

procesos de aprendizaje, construcción de ventajas competitivas y, posteriormente, desarrollos locales.

Desde una perspectiva más general o de largo plazo, la discusión sobre el rol del SL en los países en desarrollo parece aún más relevante, en la medida en que puede abrir *ventanas de oportunidad* en los sectores más dinámicos pero que se encuentran actualmente altamente concentrados. La reciente expansión de las tecnologías informacionales, redefine la importancia del alcance que tengan los derechos de propiedad intelectual; en la medida en que esto tiene fuertes repercusiones sobre las estructuras de mercado en los sectores de nuevas tecnologías y sobre las posibilidades de aprendizajes y desarrollo de capacidades en individuos y en organizaciones locales.

La discusión del rol del SL en los PED permite abordar temáticas centrales de la economía del desarrollo, ya que plantea la posibilidad de afrontar las posiciones dominantes de los PD en estos nuevos sectores. La adaptación organizacional e institucional que demanda el nuevo paradigma tecnológico puede darse de forma más natural en las regiones en las que no está tan profundamente afianzado el viejo paradigma y donde los grupos que presionan por conservar el *statu quo* son predominantemente extranjeros.

Los PD se han ubicado como proveedores monopólicos en el sector de software. El SL, en particular, y el cambio organizacional e institucional, en general, desafían las posiciones dominantes basadas en un mayor endurecimiento y control de los derechos de propiedad intelectual. El debate en los PED sobre qué posición tomar frente a la actual revolución tecno-productiva juega un rol central en la estrategia de desarrollo de estos países. La competencia entre plataformas abiertas y cerradas conduce a enriquecer la discusión sobre cómo encarar la construcción de una sociedad de la información y el conocimiento en los países menos adelantados (Hilbert, 2002).

4.2 Límites en la difusión de SL

El análisis del proceso de difusión del SL y sus límites presupone los elementos ya discutidos en los capítulos previos sobre la cuestión de las motivaciones. Apelando a las nociones de bienes club y comunidades epistémicas, se ha explicado por qué la difusión de SL no es automática sino que implica la decisión y capacidad de los agentes de ingresar a la comunidad (ya sea como productores o simples usuarios).

La posibilidad de absorción de SL por parte de un programador local depende del desarrollo de sus capacidades cognitivas y competencias tecnológicas, pero también de las de las interconexiones que tenga con otros posibles adoptantes. La presencia de economías de red complejiza las decisiones de adopción al incorporar dimensiones estratégicas asociadas a la elección de un sendero tecnológico que, *a priori*, no se puede afirmar que subsistirá en futuro y en donde los costos de migración son elevados y crecientes. La difusión de SL entre usuarios expertos se trata de un problema de adopción de tecnologías competitivas en presencia de economías de escala. Básicamente, los desarrolladores pueden elegir entre profundizar sus capacidades en tecnologías de plataforma cerrada (como las tecnologías MS) o desarrollar un sendero de aprendizaje dentro del paradigma abierto, o bien combinar ambas estrategias. No obstante, la presencia de economías de escala en el número de adoptantes juega a favor de la especialización, aunque no necesario un paradigma desplace por completo al otro. Describir del proceso de difusión permite identificar las restricciones que puedan encontrarse y diseñar de políticas públicas que permitan superarlas.

Diferentes autores han discutido y modelizado la difusión de tecnologías competitivas en el marco de economías de red (Griliches, 1957; Mansfield, 1961 y 1968; Arthur, 1989, Katz y Shapiro 1986, David y Forey 1994; Silverberg et al 1988, Antonelli, 1997, entre otros)⁷⁷. Estos trabajos coinciden en que buscan identificar las condiciones que afectan la elección de los agentes y cómo el proceso de adopción puede conducir a (i) una

⁷⁷ La bibliografía sobre difusión es muy vasta y los autores mencionados son no más que una muy pequeña muestra. La selección puede resultar caprichosa, pero permite ilustrar el avance de esta literatura y es coherente con los objetivos y el alcance de la presente investigación.

coexistencia entre las distintas tecnologías, o (ii) a un monopolio tecnológico; y, en este segundo caso, si el proceso por sí mismo permite (o no) alcanzar un óptimo paretiano⁷⁸.

4.3 Algunos modelos de difusión de tecnologías competitivas en presencia de economías de escala en la adopción.

Siverberg, Dosi y Orsenigo (1988) identificaron tres enfoques básicos que dominan el pensamiento económico de difusión: el primero que sigue la línea de los trabajos pioneros de Mansfield (1961, 1968) y Griliches (1957) que intentan representar la forma “S” de las curvas de difusión; es decir, en descubrir los parámetros y regularidades de los procesos de difusión. El enfoque epidémico de Mansfield la difusión se asocia a las expectativas de ganancias derivadas de la innovación, conducidas por una progresiva diseminación de la información sobre sus características tecno-económicas. En tal sentido, la difusión es entendida como un proceso de ajuste hacia un equilibrio que depende del aprendizaje por parte de los potenciales adoptantes.

Un segundo enfoque conocido, justamente, como modelos de difusión de equilibrio, retoma esta premisa para definir un modelo en donde la difusión es vista como una secuencia de equilibrios determinados por cambios en los atributos económicos de la innovación y del ambiente (entre ellos mencionarse el trabajo de Katz y Shapiro (1986)). Ente otras cosas, estos modelos incorporan (i) diferencias entre los potenciales adoptantes, (ii) las interacciones entre las decisiones de oferta de las firmas productoras de innovaciones, (iii) las expectativas tecnológicas de los oferentes y adoptantes, (iv) las estructuras de mercado y (v) la definición de los derechos de propiedad sobre la tecnología. Sin embargo, en ambos casos se elimina la incertidumbre radical y se sostiene que las elecciones, son guiadas por el comportamiento maximizador, bajo una hipótesis de racionalidad e información perfecta.

⁷⁸ En este caso sería que tecnología que resulte dominante sea la tecnológicamente superior.

El tercero es el enfoque evolucionista del proceso de difusión e incluye las hipótesis evolucionistas de senderos de desequilibrio, incertidumbre radical, cambio tecnológico y los procesos de aprendizaje, heterogeneidad entre los agentes, endogeneidad de las estructuras de mercado, entre otras cosas. El trabajo de Sirverberg, Dosi y Orsenigo (1988) se inscribe en esta tradición a la que se les suma los artículos de Arthur, (1989) y David y Foray (1994), comentados más adelante.

Metcalfe (1988), por su parte, plantea la diferencia del análisis de *adopción* de tecnologías, visión más micro que contempla el análisis del contexto de la toma de decisión, del de *difusión*, donde el objetivo es explicar el surgimiento de un único estándar tecnológico o la coexistencia de distintas tecnologías competitivas. En este sentido, el análisis de difusión está más asociado a la difusión de tecnologías competitivas. A partir de esto propone una taxonomía de modelos de difusión sobre la base de dos dimensiones. Por una parte, diferencia los trabajos de acuerdo a si consideran que el proceso de difusión es en equilibrio o en desequilibrio. Es decir, si es que el patrón de difusión refleja una secuencia de equilibrios cambiantes, en donde para cada momento del tiempo los agentes se hallan completamente ajustados a sus expectativas y maximizando su utilidad dados una serie de parámetros o si la difusión describe una trayectoria *hacia* el equilibrio, donde se estabilizan las cuotas de mercado. Asociado a este criterio está otro referido a si el proceso es exógeno o endógeno, en la medida en que cada equilibrio instantáneo responde a cambios en las condiciones externas o las condiciones previas determinan la evolución futura y la posición final. Por otra parte, diferencia los trabajos que, para describir las conductas de los agentes que guían el proceso de difusión, consideran, racionalidad e información perfecta, de los que trabajan con hipótesis de racionalidad limitada.

Tabla 4.1 Tipología de modelos de difusión

	Equilibrio	Desequilibrio
Racionalidad perfecta	Modelos neoclásicos. Katz y Shapiro	
Racionalidad limitada		Modelos evolucionistas: Arthur, David y Forey.

Fuente: elaboración propia sobre la base de Metcafe 1988

Por ejemplo, el modelo de Katz y Shapiro analiza el caso de adopción de tecnologías competitivas en donde las externalidades de red son importantes. Las externalidades de red provocan dos efectos sobre la dinámica de la industria. Por una parte, la tecnología más atractiva será aquella que haya tenido más ventas en el *pasado*, porque determina el alcance de la red de usuarios y los efectos de *learning by doing* y otros tipos de aprendizajes; pero, por otra parte, será también más atractiva la tecnología que prometa tener más adoptantes en el *futuro*, porque es la variable que determinará la magnitud de estos efectos y por lo tanto la utilidad futura, en este campo entran las expectativas como una variable clave del análisis.

Las preguntas que se hacen los autores son si: (i) en presencia de economía de red, ¿el mercado alcanzará un estándar *de facto*? Y (ii) si lo alcanza, ¿este será socialmente óptimo? De acuerdo con estos autores, las respuestas dependerán de la están, o no, bien definidos los derechos de propiedad sobre las tecnologías ofertadas. El modelo opera bajo los supuestos convencionales de la economía convencional sobre la información y conducta (información y racionalidad perfecta y expectativas racionales) y sobre la tecnología (conocimiento e información son sinónimos y toda la tecnología es perfectamente codificable y libre, hasta la aplicación de los derechos de propiedad).

Según esta visión, si los derechos de propiedad no están bien definidos, entonces cualquier competidor podrá entrar en el mercado y en ese caso la tecnología que sea líder en precios *hoy* alcanzará la máxima cuota de mercado. En este caso el surgimiento de un estándar *de facto* puede no ocurrir y si ocurre puede ser que no sea el socialmente óptimo. En el caso de que una firma controle los derechos de propiedad sobre la tecnología, entonces puede verse incentivada a *sponsorear* esa tecnología, ya que los retornos de una mayor cuota de mercado cubrirán los costos de la promoción (menor precio). En

este caso la estandarización va a surgir a favor de la tecnología esponsorada pero esa situación no está asociada a la presencia de un óptimo social. En tercer lugar, si en ambas tecnologías están definidos los derechos de propiedad, y los agentes tiene previsión perfecta, entonces la tecnología dominante será aquella demuestre ser superior en el futuro, ya que en la adopción no primará una comparación de los costos de hoy sino los de mañana. En este caso el óptimo dinámico está garantizado.

Tabla 4.2 Equilibrios posibles en el modelo de Katz y Shapiro

	Sin derechos de propiedad	Derechos definidos sobre una tecnología	Derechos definidos sobre ambas tecnologías
¿El mercado alcanza un estándar <i>de facto</i>?	No necesariamente	Sí	Sí
¿Es socialmente óptimo?	Si ocurre, es óptimo estático	El estándar será sobre la tecnología que tiene definidos los derechos de propiedad sea o no el óptimo dinámico	Sí, es óptimo dinámico

Fuente elaboración propia sobre base de Katz y Shapiro (1986)

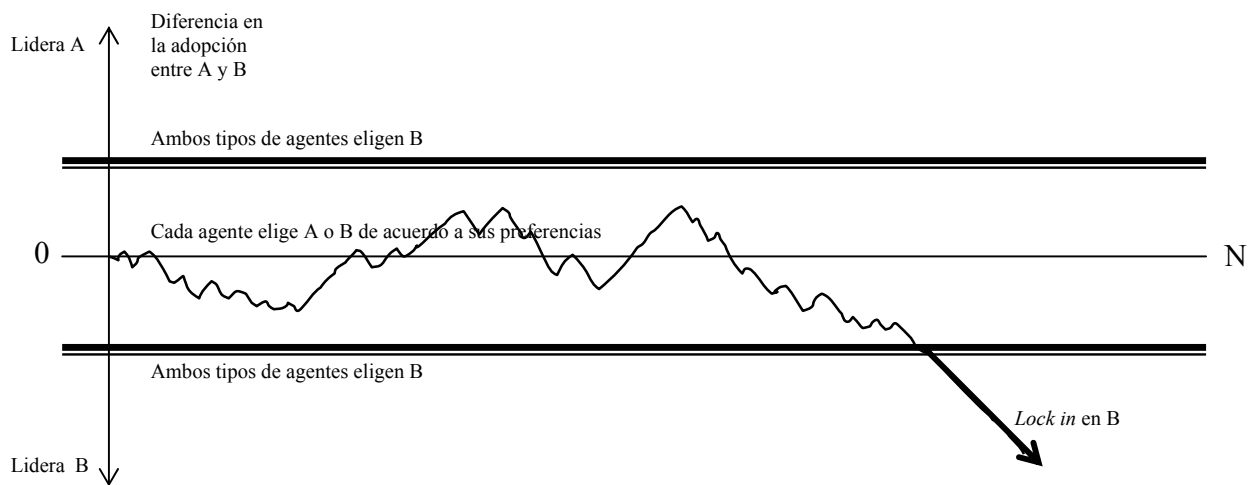
En conclusión, se trata de un modelo de equilibrio con racionalidad perfecta en donde la optimalidad de la solución depende de la internalización de las externalidades al estar correctamente definidos los derechos de propiedad. La aplicación de este modelo al caso de difusión SL en competencia con las soluciones propietarias, puede identificarse con la situación dos, es decir donde los derechos de propiedad están definidos sobre *una* de las tecnología (las soluciones propietarias). Siguiendo el razonamiento del modelo, en ese caso, los oferentes de las tecnologías libres no podrían apropiarse de una mayor cuota de mercado porque entrarían en competencia con otros oferentes, mientras que los oferentes de las tecnologías propietarias sí. Se deduce de este modelo una estandarización a favor de estas últimas. Sin embargo, esto no es lo que se observa: en todo caso hay una tendencia a la coexistencia de estándares. Por otra parte, este modelo no considera algunos elementos clave analizados en el capítulo previo, como que apropiación económica de la tecnología no depende necesariamente de la definición de derechos de propiedad en el sentido neoclásico, sino de la pertenencia al club definida por el grado de

capacidades cognitivas o de la heterogeneidad de agentes en términos de competencias y de vinculaciones.

El modelo de Arthur demuestra que los procesos de difusión con rendimientos crecientes en adopción pueden conducir a *lock-in* tecnológicos no necesariamente eficientes, causados por eventos acumulativos de accidentes históricos aleatorios. De esta forma se opone a las conclusiones del modelo de Katz y Shapiro que limita las posibilidades de *lock-in* en subóptimos a casos donde los derechos de propiedad no está correctamente definidos. Por otra parte, también se diferencia en que se trata de un modelo de desequilibrio dinámico, donde los agentes no cuentan con toda la información relevante y tienen racionalidad limitada.

De acuerdo con este autor, los adoptan la tecnología en diferentes momentos del tiempo sobre la base de comparar los retornos de adopción de ambas tecnologías. Estos retornos están compuestos por dos fuentes, por un lado la preferencia de adopción de cada individuo y en segundo lugar los retornos de escala, dados por el número de individuos con anterioridad la tecnología adoptador la tecnología. Estas funciones de retorno, que serán distintas para los diferentes tipos de agentes, provocan que, en la medida en que las economías de escala no sean importantes, primen las preferencias individuales pero que los efectos acumulativos sobre cómo los agentes vayan ingresando al mercado podrán producir que las economías de escala definan la elección y en ese caso quedará determinado un estándar *de facto*.

Figura 4.1 Acumulatividad y efectos *lock in*



Fuente: Arthur (1989)

En conclusión este modelo enfatiza el efecto de las economías de escala en la adopción y su efecto sobre la creación de *lock in* tecnológicos, el rol de la incertidumbre y la presencia de desequilibrios. Estos elementos resultan interesantes para explicar la difusión de SL, no obstante, la heterogeneidad de agentes queda reducida a dos casos y no afecta sino de una forma aleatoria sobre el patrón de difusión. Tampoco toma en consideración las redes de comunicación entre los agentes que pueden llevar a efectos contagio o de aislamiento abriendo la posibilidad de coexistencia como parecería ser el caso del SL.

Los trabajos de David y Foray (1994) y de Antonelli (1997) analizan procesos de difusión de innovaciones aplicando el concepto de percolación, en el primer caso se aplica a la difusión de estándares de para el intercambio electrónico de datos entre organizaciones y el segundo al análisis de los incentivos y la dinámica de cooperación en clubes tecnológicos. A continuación se analiza este enfoque teórico.

4.4 El concepto de percolación y adopción de tecnologías libres

Para referirse al proceso por el cual diferentes agentes económicos adoptan determinada tecnología en un momento del tiempo y lugar geográfico establecido, los economistas

suelen utilizar la expresión “difusión tecnológica”. Por difusión en física se entiende al proceso mediante el cual un *fluido* atraviesa un *medio* donde el recorrido del *fluido* está asociado a un proceso aleatorio.

El proceso de percolación es sutilmente distinto. También describe la forma en que un *fluido* atraviesa un *medio* pero en este caso, el mecanismo aleatorio está asociado al *medio* y no al *fluido* (Frisch y Hammersley 1963). *Fluido* y *medio* son expresiones abstractas que pueden adoptar significado de acuerdo al contexto. En nuestro caso en particular nos referiremos al *fluido* como al SL y al *medio* como al conjunto de desarrolladores y/o usuarios expertos y las interacciones que puedan existir entre ellos (redes jerárquicas u horizontales)⁷⁹. Todo proceso de percolación se especifica a través de las leyes (probabilísticas) que gobiernan el medio, tanto en términos de existencias de celdas libres (es decir posibilidad de receptividad) como en términos de conectores entre las celdas (conectividad). Otras dos fuerzas, de carácter exógeno, consideradas en el proceso de percolación son la densidad (en términos del número de agentes) y presión externa (asociada con elementos que influyen sobre el proceso que no están asociados a las especificidades de los agentes y del medio).

El proceso de percolación, a diferencia del de difusión, pone el énfasis en las características del medio. En términos de competencias en los agentes y la densidad de sus relaciones. La utilidad de este concepto radica en diferentes puntos. En primer lugar, parece más apropiado vincular diferencias aleatorias en el medio que en el recorrido del fluido a la hora de hablar de tecnología, dado es que lo correcto es suponer heterogeneidad de agentes más que agentes no sin idénticos entre sí. En segundo lugar, cada agente es caracterizado desde una doble perspectiva: desde su capacidad de recepción y desde su vinculación con otros agentes. De esta forma, que un agente adopte una tecnología pasa a ser función de ambas variables: recepción y conectividad, dadas otras características del medio como la densidad del entramado y la presión exógenas. En tercer lugar, los

⁷⁹ En el caso del trabajo de David y Foray el fluido está asociado al estándar EDI (*Electronic Data Interchange*) y el medio a el conjunto de empresas. En el paper de Antoneli, el fluido es el conocimiento localizado y el medio el entramado productivo o clubes tecnológicos.

modelos tradicionales sobre la difusión tecnología con rendimientos crecientes conllevan a la convergencia hacia una técnica, cuestión que frecuentemente no se constata empíricamente. Este enfoque permite predecir la coexistencia de estándares ya que el proceso de adopción toma lugar en un territorio definido por celdas y conectores que pueden dar lugar a grupos o islas con fuertes vinculaciones internas pero relativamente aislados del resto, de forma que la generación de estándares sea local y no global⁸⁰. Por otra parte, en el concepto de percolación está planteada la existencia de economías de red, poniendo de manifiesto que los grados de libertad de la decisión de adopción se reducen a medida que se generaliza el uso de una de las tecnologías competitivas en un entorno dado.

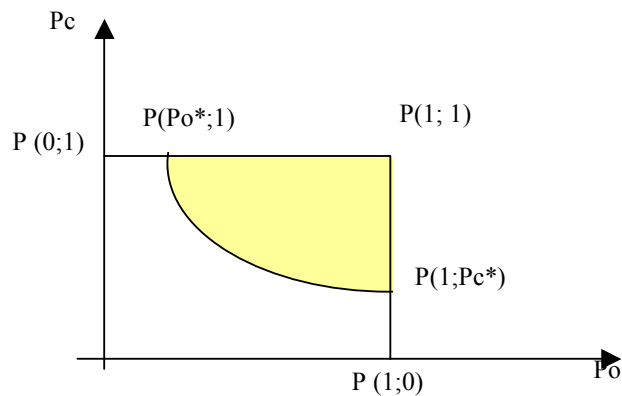
Este enfoque puede ser aplicado en la caso del SL ya se trata de una tecnología cuya barrera de entrada puede ser definida en términos de umbrales mínimos en la capacidades de los agentes y en las vinculaciones con agentes que hayan adoptado SL. La característica de bien club del software libre, la relevancia de los procesos de aprendizaje a través de interacciones y las barreras a la entrada se definidas por capacidades diferenciales, permiten afirmar que la adopción de software libre depende fuertemente de la receptividad y conectividad de los agentes. En este sentido, las decisiones de adopción de SL pueden ser analizadas a través del concepto de percolación.

En este caso la probabilidad de receptividad (P_o) está asociada al grado de capacidades cognitivas que le permitan a los agentes a apropiarse del código disponible, mientras que la probabilidad de conectividad (P_c) puede asociarse al grado de pertenencia a comunidades virtuales que potencien el intercambio de información. Por otra parte, el nivel de presión externa afecta a adopción más allá de los valores P_o y P_c antes definidos, es decir la presión externa (E) es lo suficientemente importante, la probabilidad de percolación es mayor para cualquier valor de P_o y P_c . De forma similar, mayor número de agentes en una trama dada, es decir, mayor de densidad (G), mayor será la probabilidad de percolación.

⁸⁰ Esta propiedad, David y Foray la denominan la naturaleza local de los efectos *feedback*.

Formalmente, la probabilidad de percolación P es la probabilidad conjunta de P_o y P_c dados E y G : $P = P(P_o P_c / E; G)$. El proceso de percolación está caracterizado una serie de propiedades que juegan un rol central en el análisis de adopción de SL. En primer lugar, la existencia de valores críticos de $P_o > P_o^*$ y $P_c > P_c^*$ debajo de los cuales la probabilidad de percolación es cero. Esto puede ser entendido como la necesidad de umbrales mínimos en ambas variables requeridos para que la percolación sea posible.

Figura 4.2 Propiedad de proceso de percolación. Niveles mínimos y área con probabilidad de percolación superior a cero



Fuente: David y Foray (1994).

En segundo lugar, que la probabilidad de percolación es mayor en sistemas con conectores imperfectos y elevada receptividad que en el caso inverso. Esta propiedad es fundamental para el diseño de política porque significa que los esfuerzos puestos para mejorar la receptividad de los agentes incrementa más la probabilidad de percolación que haciendo los mismo esfuerzos en conectividad.

La interpretación económica del proceso de percolación como herramienta para entender las leyes que gobiernan la circulación de la información y conocimiento al interior de una comunidad se vuelve más importante cuando los parámetros de la probabilidad de percolación no son asumidos como exógenos y estáticos sino como endógenos y dinámicos. La conectividad y la receptividad, son de hecho variables que se pueden construir a partir de la aplicación de políticas activas (Antonelli, 1997).

El análisis de la información empírica realizado en el capítulo 5 está basado en las ideas que subyacen a al concepto de percolación. En ese capítulo se evalúan las posibilidades de adopción del SL a partir de la capacidad de receptividad –capacidades cognitivas y tecnológicas de los agentes- y la conectividad –interacciones y vínculos con otros agentes y su efecto sobre la decisión de adopción-.

4.5 Síntesis y conclusiones

En este capítulo, luego de una justificación del análisis de la difusión de SL en países en desarrollo y en la Argentina en particular, propone una serie de enfoques teóricos para caracterizar el proceso de difusión.

Existe una larga tradición en la literatura sobre procesos de difusión que va desde enfoques más convencionales con supuestos de información y racionalidad perfecta a enfoques donde predomina las nociones de desequilibrio e incertidumbre cuyas principales conclusiones son el predominio de soluciones subóptimas imposibles de determinar a priori.

El enfoque de percolación utilizado de David y Foray (1994) y Antonelli (1997), resulta útil para el análisis del proceso de adopción de SL entre usuarios expertos en la Argentina y provee elementos para caracterizar sus limitaciones. Por otra parte, este enfoque resulta apropiado dadas las características de SL en tanto bien económico discutidas en el capítulo 3. De acuerdo con este enfoque se requieren umbrales mínimos en las competencias de los agentes como en sus vinculaciones para haya un mayor grado de adopción de SL, asimismo, las propiedades fundamentales del proceso de percolación provee elementos para el diseño de políticas públicas que lleven a incrementar la utilización del mismo.

5 DIFUSIÓN DE SOFTWARE LIBRE EN LA ARGENTINA. LÍMITES Y RESTRICCIONES

5.1 Introducción: rasgos característicos

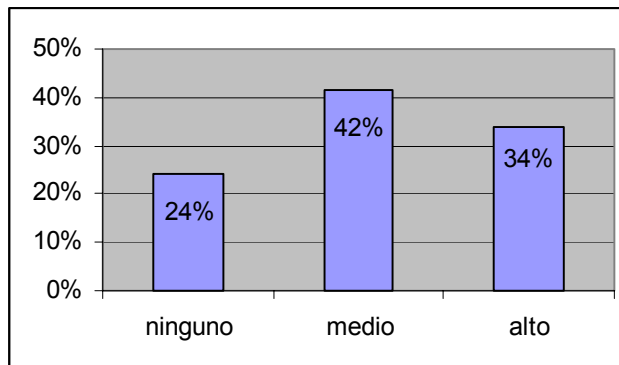
A partir de la información que surge de la encuesta a trabajadores informáticos (ver detalles en el anexo correspondiente) pueden estilizarse algunos rasgos del proceso de difusión de SL en la muestra.

Las características fundamentales son que: (i) el grado de adopción de SL es relativamente elevado entre los trabajadores de la muestra (más de las tres cuartas partes del panel lo utilizan en alguna medida), pero tipo de uso es relativamente pobre en términos del tipo de aplicaciones que predominan; (ii) la decisión de utilización de software libre recae sobre las personas y no se trata de una decisión de la organización donde se desempeñan estos trabajadores, lo que resta sistematicidad al uso de SL; (iii) la difusión está asociada a las capacidades de los trabajadores y, en especial al dominio de las herramientas de desarrollo vinculadas a las metodologías ágiles y al paradigma del SL; (iv) el desarrollo local de SL es muy limitado y las motivaciones están asociadas al desarrollo de capacidades y a cuestiones políticas o ideológicas, (v) el dominio del idioma inglés y la utilización de herramientas de comunicación digital están asociados tanto al uso como al desarrollo SL.

5.2 La difusión de SL entre programadores argentinos: un análisis descriptivo

Del total de encuestados el 76% señaló hacer algún uso de SL. Sin embargo sólo el 34% clasificó su nivel de utilización como elevado. Esta situación da la pauta de que el porcentaje de trabajadores que interactuó alguna vez con SL es importante, pero que por algún motivo elige seguir operando con soluciones propietarias en vez de incrementar el grado de utilización de SL.

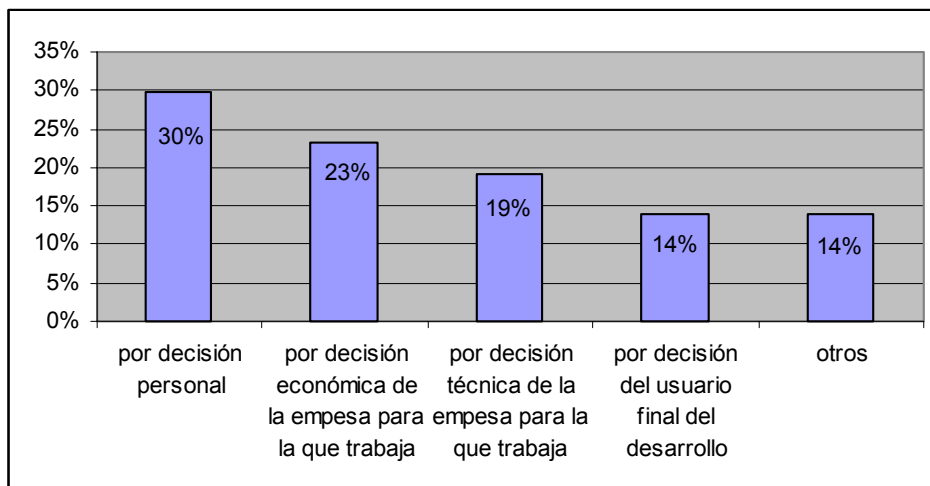
Gráfico 5.1 Nivel de utilización de SL



Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS.

La decisión de utilizar software libre frecuentemente es personal ya que no existen requerimientos específicos al respecto (30% de los que utilizan). En el 42% de los casos la decisión recae sobre la empresa o institución para la que trabajan; que basa esta decisión en motivos económica el 23% de las veces, en motivos técnica en el 19% de los casos.

Gráfico 5.2 Sobre quién recae la decisión de utilizar SL

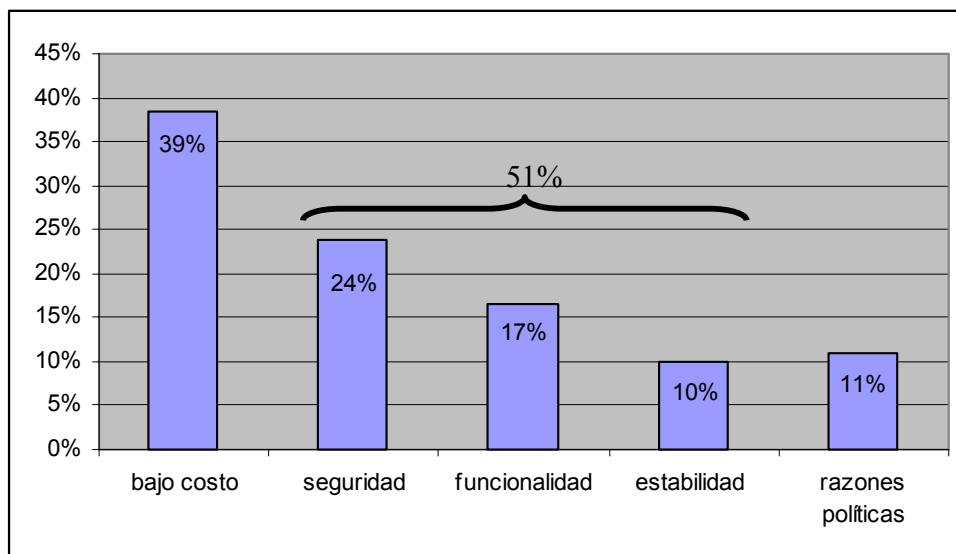


Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Porcentajes sobre el total de trabajadores que utilizan SL.

Cabe destacar el bajo nivel demandas SL de los usuarios finales del desarrollo. Esto da la pauta de que es muy fuerte el uso de SL por motivos personales. Cuando se trata una decisión de la organización ésta se debe fundamentalmente a las ventajas económicas seguida por las ventajas técnicas. Algunas experiencias de adopción de SL por parte de empresas argentinas se explica por esto mismo: trabajadores del área de sistemas que abogan por el uso de SL plantean los beneficios para la empresa posibles de su adopción, en algunos casos desacuerdos con otras áreas (como compras) impiden la migración de lleve a cabo. Esta es una situación frecuente donde la falta de traducción entre los usuarios finales de la tecnología y los del departamento de compras, conduce a que estos últimos realizan la elección de la tecnología con parámetros tales como la existencia de contratos previos o la valoración de una marca reconocida, sin que necesariamente se ajuste a las necesidades de los usuarios de la tecnología en la empresa. En otros casos, las sugerencias de el área de sistemas encuentra eco en áreas estratégicas, este es el caso de empresas que ante la crisis tuvieron que adoptar medidas de racionalización de recursos y, por lo tanto, una opción de implique no solo ahorros de licencias sino también de hardware suele ser escuchada. Algunas empresas argentinas que migraron algunos de sus servidores y/o *work stations* a plataformas abiertas que pueden mencionarse son: Aceitera Gral. Deheza, Loma Negra, Hospital Italiano, entre otras (Information Technology, 2003)

Entre los motivos por los cuales se utilizan soluciones libre figuran fundamentalmente el bajo costo y cuestiones técnicas, entre ellas se destacan la seguridad, la estabilidad y la funcionalidad del SL.

Gráfico 5.3 Motivos de utilización de SL

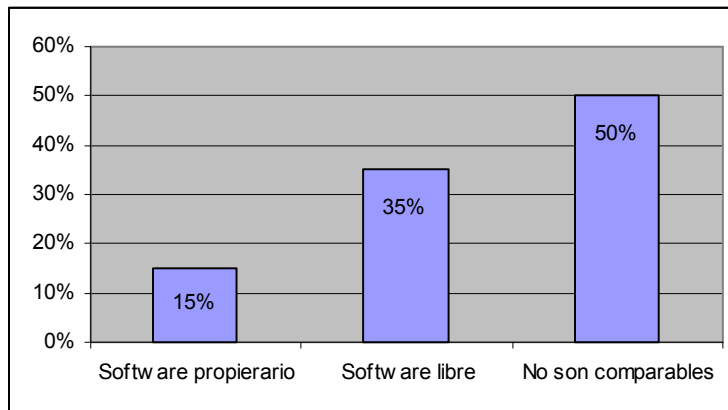


Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Porcentajes sobre el total de trabajadores que utilizan SL.

A pesar de que las cuestiones técnicas predominan sobre el resto de las razones, los costos son un motivo relevante en la elección. El encarecimiento de las licencias del software propietario después de la devaluación condujo a que muchas empresas revisaran su estrategia de abastecimiento de software. Como se señaló en el capítulo 2, a pesar de que en los países desarrollados pueda ser discutible que el costo de utilización del SL alcance al propietario, en economías en desarrollo como la Argentina, donde la hora programador es relativamente barata difícilmente el software propietario pueda competir en precios. De hecho, muchas de las migraciones hacia SL que se realizaron recientemente, terminaron por concretarse durante la crisis de 2002 motivadas por la reducción de costos.

Entro los usuarios de SL, predomina una visión de complementariedad más que de competencia entre ambos paradigmas. Ante la pregunta de si es el SL o el software propietario el que satisface mejor los requerimientos actuales; el 50% de los usuarios opinó cubre necesidades distintas, mientras que el 35% sostuvo que el SL es más apropiado.

Gráfico 5.4 Qué satisface mejor los requerimientos actuales



Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Porcentajes sobre el total de trabajadores que utilizan SL.

Solo el 12% del total del panel y el 16% de los que utilizan SL participan en desarrollos libres. La mayoría comenzó a participar en la comunidad de desarrolladores después de la expansión de Internet, a mediados de la década del 90⁸¹. La dedicación es muy intensiva: en 4 casos supera las 30 horas semanales y para el resto el promedio de participación es de 8 horas semanales.

Con relación a las motivaciones, se destacan aquellas relacionadas con el desarrollo de capacidades. El motivo señalado con mayor frecuencia fue *desarrollar habilidades*, seguido de *compartir conocimiento*. En segunda instancia se ubican las argumentaciones más políticas, es decir, que su participación en desarrollos libre se debe a que consideran que sobre el software *no deberían recaer derechos de propiedad intelectual*. Estas respuestas son coherentes con la hipótesis de esta investigación sobre una estrecha relación entre la creación de competencias y el uso y desarrollo de SL. Por otra parte el 71% de los encuestados que desarrollaban SL, afirmó *ganar dinero* con su actividad en el campo del SL: directamente (al 52% de los que desarrollan SL, les pagan por hacerlo o por proveer servicios asociados) o indirectamente (el 19% de los desarrolladores de SL).

⁸¹ Sólo dos casos son anteriores; uno en 1989 y otro en 1990.

Box 5.1 Rasgos del panel.

Los rasgos característicos del panel pueden sintetizarse en⁸²:

* **La mayor parte son jóvenes, el 35% menor de 30 años y el 80% menor de 40.** Existe una asociación negativa entre la edad y el grado de utilización de SL⁸³. Los trabajadores más jóvenes ha demostrado utilizar más intensivamente el SL. Así mientras que el 72% de las personas que no utilizan SL tiene más de 30 años, el 67% los de alta utilización de son menores de treinta

* **Predominan los hombres (83% del panel).** Sin embargo no se detectó relación entre el género y el grado de utilización de SL.

El 58% vive en la ciudad de Buenos Aires y en el Gran Buenos Aires, el 29% en el resto del país y el 5% restante son argentinos residentes en el exterior. Se detectó una mayor participación de residentes del interior del país entre los de mayor grado de utilización de SL⁸⁴.

* **El 50% tiene formación universitaria completa⁸⁵, y el 26% universitaria incompleta. El 17% del panel declaró haber realizado estudios de postgrado vinculados a la informática.** Existe una asociación positiva el grado de educación formal y el grado de utilización de SL⁸⁶, lo que está relacionado con la idea que el SI es un red de conocimiento y que se requiere un umbral mínimo de competencias básicas.

* **La mitad del panel hace menos de 10 años que trabaja en informática, mientras que sólo el 10% hace más de 25 años que se desempeña en el sector, lo que está fuertemente correlacionado con la edad de los entrevistados.** Al igual que con la edad, existe una asociación negativa entre la antigüedad en el trabajo y el grado de utilización de SL⁸⁷.

⁸² Para mayor detalle ver el anexo estadístico.

⁸³ La probabilidad de aceptar la hipótesis nula de ausencia de asociación es inferior al 5%.

⁸⁴ La probabilidad de aceptar la hipótesis nula de ausencia de asociación es inferior al 1%.

⁸⁵ De los cuales el 11% tiene estudios no vinculados con la informática, 60% son ingenieros y licenciados en computación y el 29% restante son analistas y licenciados en sistemas.

⁸⁶ La probabilidad de aceptar la hipótesis nula de ausencia de asociación es inferior al 10%.

⁸⁷ Con una probabilidad de rechazar la hipótesis nula inferior al 0%

*** El 57% se encuentra empleado en relación de dependencia, mientras que el 23% es contratado por obras o servicios y el 20% restante es propietario. De los empleados en relación de dependencia, prácticamente el 60% trabaja en empresas privadas, el 23% en el estado nacional, provincial o municipal y el 17% restante en instituciones académicas o tecnológicas, o en ONGs. Por su parte, cerca del 45% de los contratados por obras o servicios trabaja en el Estado, mientras que el 42% lo hace en empresas privadas. A pesar de que no se puede afirmar que haya relación entre el tipo de organización para la que trabaja y el grado de utilización de SL⁸⁸, el grupo de usuarios de SL está sobre representado entre los trabajan en instituciones académicas o de investigación. El caso del tipo de relación con la organización y uso de SL no se detectó asociación.**

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS.

5.3 Receptividad y conectividad

Aplicando el concepto de percolación se puede asumir que la adopción de software libre en ambientes técnicos (entre trabajadores informáticos) depende fuertemente de la receptividad y conectividad de los agentes. Para testear esta hipótesis se desarrollaron indicadores de: (i) adopción de SL, (ii) utilización y desarrollo de redes (*proxi* de conectividad) y (iii) competencias tecnológicas (*proxi* de receptividad). El primer indicador surge simplemente de la pregunta de la encuesta sobre el nivel de uso de SL (ver anexo correspondiente). El grado de utilización estuvo originalmente categorizado en cinco niveles: *Ninguno, Alguno, Bajo, Medio y Alto*.⁸⁹

El indicador de conectividad

⁸⁸ Con una probabilidad de rechazar la hipótesis nula inferior al 20%

⁸⁹ En algunos gráficos y cuadros, por motivo de claridad en la exposición, se presenta el grado de utilización en tres categorías *Ninguno, Bajo y Alto*; donde *Bajo* engloba a la segundo y tercer nivel de la original y *Alto* al cuarto y quinto.

En el segundo caso, se construyó un índice que incorporó información sobre el grado de utilización de herramientas de comunicación digital y la participación en grupos o comunidades virtuales.

Se parte del supuesto de que cuanto mayor sea la interacción con otros agentes que hayan adoptado SL mayor será la posibilidad de efecto contagio. Para que este efecto contagio se dé es necesario que los agentes estén interconectados entre sí. El uso intensivo de las herramientas de comunicación digital da la pauta de que los agentes mantienen intercambios de información que pueden afectar su decisión de adopción. En segundo lugar, se supone que la utilización de herramientas de comunicación digital puede resultar insuficiente para evaluar la conectividad porque puede estar asociada a comunicaciones esporádicas. La participación en comunidades virtuales y en grupos de usuarios permite evaluar la relación con otros agentes con contenidos específicos y continuidad en el tiempo. En este sentido, el indicador de conectividad conjuga ambos elementos: la utilización de herramientas de comunicación digital así como la pertenencia e interacción con comunidades virtuales (ver anexo correspondiente).

Indicador de receptividad

En construcción de un indicador que diera cuenta de la receptividad se planteó la necesidad de describir la trayectoria tecnológica de los agentes. Para ello fue necesario identificar patrones comunes en el tipo de herramientas (lenguajes y tecnologías) que dominan para poder describir, a partir de otras variables *ilustrativas*, el sendero de creación de competencias y desarrollo de capacidades tecnológicas.

Se partió del supuesto de que las herramientas de desarrollo tiene un rol central en la construcción capacidades de los trabajadores⁹⁰. Evidentemente, el dominio de herramientas está lejos de explicar por sí mismo las competencias tecnológicas de los

⁹⁰ Jordy Micheli (2003) señala las herramientas de trabajo utilizadas de modo efectivo y creativo, así como sus conocimientos y habilidades y la manipulación de datos, palabras, representaciones orales y visuales son mas importantes que las credenciales acerca de su nivel y campo de estudios a la hora de estimar las capacidades de estos trabajadores.

trabajadores informáticos. Las capacidades de estos trabajadores se construyen de una forma sistémica, donde el dominio de herramientas constituye una variable objetiva para medir estas capacidades, pero en su determinación intervienen además dimensiones como la educación formal, la formación en el trabajo, la utilización de redes, entre otros (Borello et al 2005). No obstante, se optó por agrupar a los trabajadores de acuerdo al tipo de herramientas en la medida en que a partir de ellas era posible identificar y caracterizar el sendero tecnológico recorrido.

Las trayectorias tecnológicas fueron identificadas utilizando la técnica estadística de *análisis factorial de correspondencias múltiples*. A través de esta técnica se pudo agrupar a los individuos de acuerdo a patrones comunes en el tipo de herramientas que dominan, ya que este método permite sintetizar las asociaciones existentes entre las diferentes modalidades de las variables consideradas y así reducir la dimensionalidad del fenómeno estudiado. Para ello, conforma ejes factoriales que permite concentrar el análisis en aquellas variables y modalidades que más aportan. Finalmente, se agrupó a los individuos con características similares mediante la utilización del método *análisis de cluster*. El análisis de *cluster* construye grupos minimizando las distancias euclidianas calculadas con las coordenadas de cada individuo en ejes factoriales que surgen del análisis factorial⁹¹.

El conjunto de variables *activas* seleccionadas para la construcción de los grupos fueron los lenguajes y tecnologías que dominan los trabajadores informáticos. Además, fueron consideradas una serie de variables *ilustrativas* que permitieran la caracterización del sendero de formación de capacidades de estos actores.

De esta forma, los trabajadores fueron agrupados en cinco categorías de trayectorias tecnológicas y de aprendizaje (una incierta y cada una de las otras cuatro, asociada a un paradigma específico de lenguajes y tecnologías): (i) sin perfil específico; (ii) perfil Mainframes y microcomputadoras; (iii) perfil plataforma cerrada; (iv) perfil plataforma abierta y (v) perfil académico. (Para mayor detalle sobre la conformación de los grupos ver anexo correspondiente). Este tipo de agrupación resultó clave porque permitió

⁹¹ Este estudio construyó los grupos con información correspondiente a los diez primeros ejes factoriales.

diferenciar las capacidades y receptividad en función del *tipo* de herramientas que dominan sin introducir escalas de complejidad para comparar las herramientas entre sí⁹².

El primer grupo resultó ser el más numeroso esta representado por un conjunto de personas **sin perfil específico** en término del tipo de herramientas que domina. Básicamente se definió por el no dominio de las herramientas más complejas y novedosas del grupo de herramientas sobre las que se los interrogó. Este grupo esta constituido por el 48% de la personas entrevistadas y algunas de sus características relevantes son: (i) más de 20 años de experiencia laboral, lo que estaría indicando que se trata de un grupo con edades que entre 35 y 50 años; (ii) la rotación es relativamente baja en comparación con los otros grupos (hasta un máximo de dos veces en cinco años) (iii) trabajan predominantemente en el sector público. En el grupo existe una leve predominancia del género femenino en comparación con la participación del genero en el total del panel. En lo que respecta a la trayectoria tecnológica y al sendero de aprendizaje, los integrantes de este grupo se caracterizan fundamentalmente por no haber seguido un sendero definido en la construcción de sus competencias, es decir, a diferencia de otros actores no ha logrado especializarse en un conjunto de herramientas con patrones comunes. No obstante, que la mayor parte de los entrevistados caiga en este grupo no es una debilidad del estudio, ya que la hipótesis que plantea es que determinado sendero tecnológico se asocia a un uso particular de SL. La ausencia de sendero definido en todo caso puede entenderse como la falta de especialización, lo que no implica que no puedan ser asociados a más de una de las trayectorias que serán definidas más abajo o que estén actualmente construyendo su propio sendero.

El segundo grupo constituido por el 12% del panel presentó especial afinidad por las herramientas tales como Cobol, Cliper y Xbase, entre los lenguajes, y AS 400 y mainframes, entre las tecnologías. Por ello fue definido como el grupo de **perfil Mainframes y microcomputadoras**. Este grupo se caracteriza por ser el de mayor

⁹² Otra opción considerada en el momento del análisis fue categorizar a cada una de las tecnologías por su grado de complejidad y luego a cada grupo por su dominio en tecnologías más o menos complejas. Este análisis fue aplicado en Borello et al (2005).

experiencia laboral y mayor edad, mientras que la rotación laboral es significativamente menor que el promedio para el panel. Trabajan mayoritariamente en organizaciones no dedicadas al desarrollo de software que pertenecen fundamentalmente al sector financiero o al Estado. Predominan en este grupo los trabajadores en relación de dependencia. En cuanto su formación, se caracteriza por haberla adquirido por afuera del sistema educativo formal, con alta presencia de los cursos de formación y formación en el trabajo. Los trabajadores de este grupo pueden asociarse a aquellos que se desempeñan en los departamentos de sistemas de organizaciones de mediano o gran tamaño con fuertes necesidades de informatización, justamente como el Estado y el sector financiero y bancario. Este tipo de organizaciones avanzó en la informatización de sus actividades a partir de la década de los 70 y 80 y aún perduran muchos de los equipos instalados en su momento. De la misma forma que los encargados de administrar el funcionamiento de estas máquinas son los mismos. Las trayectorias en las capacidades tecnológicas de estos trabajadores van de la mano del sendero tecnológico de la organización. En este sentido los costos de migración en estos casos son muy elevados. Se espera que el grado de difusión entre las personas de este grupo sea bajo o nulo ya que se presenta como de reducida receptividad.

El tercer grupo, que se lo denominó **perfil plataforma cerrada** o propietaria, significa el 15% del total. Entre los lenguajes que dominan que caracteriza al grupo están Visual Basic y ASP, mientras que entre las tecnologías Punto Net, COM, DCOM, MTS. Este conjunto de herramientas tienen la particularidad de ser tecnologías propietarias y monoplataforma, ya que corren sólo en Microsoft Windows. Otras características del grupo son que se trata de personas entre 20 y 30 años de edad y con menos de 10 años de experiencia laboral. La rotación laboral para el grupo es relativamente elevada. Las personas que componen el grupo se desempeña tanto en empresas cuya actividad principal es la informática como en otro tipo de organizaciones. Tampoco se detecta un patrón sectorial en la demanda, lo que estaría indicando la transversalidad de esta clase de herramientas. En cuanto a su educación formal predominan las licenciaturas en sistemas. El grupo se caracteriza por considerar a los cursos de capacitación como una de las

fuentes principales en su formación profesional, al tiempo que se destaca, a diferencia del resto, por haber obtenido certificaciones⁹³. Esto da la pauta de que la construcción del sendero tecnológico se dio en el marco de una formación muy vinculada al tipo de herramientas que utilizan. En este sentido se considera que la receptividad en este grupo sea reducida.

El cuarto grupo **perfil plataforma abierta** se caracteriza por la utilización de herramientas asociadas a este tipo de plataformas y representa el 15% del panel. Entre los lenguajes que dominan los miembros de este grupo se destacan: C, C++, Java, Unixshell, Python y Perl. Y entre las tecnologías el grupo se destacó por dominar programación multitier, plataforma web, XML, todas ellas relacionadas con tecnologías de comunicación del tipo intranet/internet. Y UML, Programación orientada a objetos y Extreme Programming, todas ellas relacionadas con las metodologías ágiles. Una cualidad fundamental de este tipo de herramientas es ser relativamente nuevas. Otras características del grupo son la experiencia laboral menor a 20 años, con edades entre 20 y 30, y la muy elevada rotación (tres veces en cinco años), lo que está asociado al dinamismo asociado con estas tecnologías. Asimismo, trabajan mayoritariamente en empresas cuya actividad principal es el desarrollo de software. En cuanto a la educación formal, el grupo se caracteriza por la formación universitaria completa en la carrera de licenciatura en ciencias de la computación. No obstante, este grupo se define porque sus miembros consideran que sus capacidades las obtuvieron de forma autodidacta. Evidentemente se asocia este grupo al de elevada receptividad, dado de el sendero tecnológico recorrido está vinculado al del SL.

El último grupo **perfil académico** (9% del total del panel) se destacó por el dominio de herramientas de mayor interés académico que de mercado. Incluso, de la encuesta surge que no hace uso de ellas en su trabajo actual. Esto indicaría que estos trabajadores han desarrollado capacidades que si bien no aplican directamente a su trabajo amplían su competencias. Los lenguajes que dominan por lo que se caracteriza el grupo son Lisp,

⁹³ En informática, grandes empresas tales como Microsoft, IBM, Oracle extienden títulos de certificación a partir del dictado de cursos sobre las herramientas y tecnologías que estas mismas empresas desarrollan y comercializan.

Smalltalk, C, C++, Python, Assembler, Haskell, Perl. Entre otras características del grupo pueden mencionarse la experiencia laboral menor a 10 años, la rotación laboral elevada (dos veces en cinco años). Este grupo se caracteriza porque sus integrantes se desempeñan mayoritariamente en instituciones académicas. Asimismo destaca la presencia de personas con estudios de postgrado en informática, así como los licenciados en ciencias de la computación. A pesar del elevado nivel de educación formal, los miembros de este grupo, al igual que en el caso anterior, se caracterizan por considerar que sus capacidades las obtuvieron de forma autodidacta. La trayectoria tecnológica en este caso está claramente asociada a la actividad académica que estuvo cerca del desarrollo del SL. Se espera encontrar en este grupo una elevada receptividad al uso de SL.

5.4 Relación entre las trayectorias tecnológicas, uso de redes y difusión de SL

En estos grupos se analizó el uso de SL y de redes, de tal forma se pudo establecer una relación entre la trayectoria tecnológica, la conectividad de los agentes y la difusión de SL, lo que contribuyen a entender sobre qué bases se difunde el SL en la Argentina y cuáles son sus limitaciones.

Se detectó que tanto la receptividad medida como las trayectorias tecnológicas de los agentes y la conectividad, en términos de grado de utilización de herramientas de comunicación digital como o pertenencia a comunidades y grupos virtuales, está asociada a la utilización de SL. Además del grado de utilización también se estudió el tipo de uso que cada grupo hace del SL, es decir, sobre quién recae la decisión de utilización de SL, cuáles son las razones por las que eligen SL, si además de la utilización realiza desarrollos de SL y sobre qué motivaciones y entre quienes no lo utilizan por qué eligen plataformas propietarias. De esta forma se logró una descripción más completa del tipo de difusión que se logró en cada caso.

Tabla 5.1 Cluster de herramientas y grado de utilización de SL

Cluster de acuerdo a las herramientas que dominan	Grado de utilización de SL				
	Ninguno	Alguno	Bajo	Medio	Alto
Sin perfil específico	34,1%*	17,00%	23,90%	12,50%	12,50%
Mainframes y minicomputadoras	36,40%*	9,10%	36,40%	13,60%	4,5%*
Plataformas cerradas	22,20%	14,80%	44,4%*	14,80%	3,7%*
Plataformas abiertas	3,6%***	17,90%	14,30%	21,40%	42,9%***
Herramientas de interés académico	6,3%*	18,80%	18,80%	25,00%	31,3%*
Total	25,40%	16,00%	26,50%	15,50%	16,60%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación menor al 1%; * Prueba Z significativa al 10%; *** Prueba Z significativa al 1%.

El primer grupo se caracteriza por no utilizar SL, es decir el porcentaje de individuos del grupo que no utiliza SL es significativamente superior al promedio de no-utilización del panel. A pesar de esto, como puede apreciarse en el cuadro previo un porcentaje elevado del grupo utiliza SL con diferente grado, en estos casos la decisión de utilización recae sobre la organización para la que se desempeña, más que sobre una decisión personal, bajo un criterio de reducción de costos. Entre los que no utilizan SL el motivo es fundamentalmente por política de la empresa, aunque también señalaron la falta de compatibilidad así como la falta de conocimiento de soluciones SL que pudieran aplicar a su tarea. Por último, el grupo también se destaca por no desarrollar SL.

El segundo grupo también se caracterizó por la no-utilización de SL, los motivos fueron por política de la empresa y por la falta de compatibilidad. Entre los miembros del grupo que utilizan SL se destacaron los motivos referidos al bajo costo y la estabilidad. Este último es coherente con su trayectoria tecnológica ya que por trabajar en ambientes de alta disponibilidad otorgan una gran valoración a estabilidad que resulta fundamental. En estos casos la decisión de utilización de SL recae sobre la empresa para la que trabaja bajo criterios económicos (menores costos) pero también técnicos. Este grupo también se caracterizó por no desarrollar SL.

En lo que respecta al tercer grupo, se caracteriza por una utilización de SL baja, a diferencia de los casos anteriores. Entre las razones que condujeron a la elección de SL destacan el bajo costo y la seguridad.

En el cuarto grupo la utilización de SL es elevada. El uso de SL en el ambiente laboral responde a una decisión técnica por parte de la empresa donde trabaja, aunque también la decisión recae directamente sobre los trabajadores. Acorde con esto último, los motivos centrales de utilización de SL son la funcionalidad y la disponibilidad del código fuente. Se puede caracterizar a este grupo por su vocación por la informática⁹⁴ y las nuevas tecnologías. El acceso a Internet les ha permitido desarrollarse profesionalmente de ahí su espíritu autodidacta. El SL se les presenta como una nueva tecnología completamente abierta con la que pueden desarrollar nuevas capacidades: la disponibilidad del código fuente se vuelve para este grupo una cualidad sumamente atractiva. Este grupo se caracteriza y se diferencia de los anteriores en que un porcentaje significativamente elevado desarrolla SL además de utilizarlo. Las motivaciones para ello ésta relacionadas con compartir conocimiento y desarrollar habilidades. Lo que una vez más es coherente con su espíritu autodidacta.

En el último grupo, el grado de utilización de SL es también elevado. No obstante se diferencia del anterior en que en este caso, el uso de SL en el ambiente laboral es una decisión personal ya que en organización donde se desempeña no hay requerimientos específicos al respecto. También se diferencia en que los motivos centrales de utilización de SL se destacan las razones políticas así como el deseo de participación de la comunidad. Al igual que el grupo anterior se caracteriza por desarrollar SL, entre las motivaciones, los integrantes de este grupo mencionan las posibilidades de compartir conocimiento y desarrollar habilidades, pero también participar en la escena del SL. De tal forma, una diferencia central entre este grupo y el anterior es que le otorga a la problemática del SL una dimensión política mientras que el grupo cuatro se limita al problema técnico.

Tabla 5.2 Cluster de herramientas y desarrollo de SL

⁹⁴ El tipo de carrera que siguieron mayoritariamente (Ciencias de la computación) tiene una aproximación a la informática desde una perspectiva científica y académica. Otras carreras ofrecen un acercamiento a la informática desde los negocios.

Cluster de acuerdo a las herramientas que dominan	Desarrolla software libre	
	Si	No
Sin perfil específico	10,50%	89,50%
Mainframes y minicomputadoras	0%**	100,00%
Plataformas cerradas	9,50%	90,50%
Plataformas abiertas	29,6%*	70,40%
Herramientas de interés académico	40%***	60,00%
Total	16,40%	83,60%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación menor al 1%; * Prueba Z significativa al 10%; *** Prueba Z significativa al 1%.

En lo que se refiere a la relación entre difusión de SL y la conectividad, también se detectó una relación positiva. Tal como puede apreciarse en la tabla siguiente, a menor grado de utilización de herramientas de comunicación digital y menor pertenencia a comunidades y grupos virtuales, menor es utilización de SL. La relación positiva no parece sorprendente si se sabe que Internet es el principal medio de difusión del SL.

Tabla 5.3 Grado de conectividad y de utilización de SL

Grado de conectividad	Grado de utilización de SL					
	Ninguno	Alguno	Bajo	Medio	Alto	Total
Muy Bajo	53,3%***	6,7%	26,7%	13,3%	0%*	100%
Bajo	20,7%	19,0%	34,5%	10,3%	15,5%	100%
Medio	32,7%	19,2%	17,3%	17,3%	13,5%	100%
Alto	17,5%	12,5%	32,5%	20,0%	17,5%	100%
Muy Alto	12,5%	12,5%	12,5%	18,8%	43,8%***	100%
Total	25,4%	16,0%	26,5%	15,5%	16,6%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación menor al 4.5%; * Prueba Z significativa al 10%; *** Prueba Z significativa al 1%.

Esta asociación pone de manifiesto la necesidad de umbrales mínimos de vinculaciones en la utilización de SL.

5.5 Síntesis y conclusiones

De acuerdo con los datos empíricos analizados, el uso de SL está asociado a determinadas trayectorias tecnológicas definidas por el tipo de herramientas que los trabajadores informáticos dominan, por lo que estas trayectorias pueden definirse como la receptividad de los agentes al SL. El sendero de aprendizaje transitado a partir de la formación profesional y la experiencia concreta frecuentemente está asociado a un tipo determinado de herramientas de desarrollo. Si la trayectoria tecnológica no es compatible con el tipo de herramientas que predominan en el SL, entonces la probabilidad de receptividad está por debajo del umbral mínimo requerido para que haya percolación. Dicho de otro modo, las competencias del trabajador son incompatibles con la adopción del SL, porque no cuenta con las competencias necesarias para la decodificación necesarias para ingresar a la comunidad.

Por otra parte, la adopción no solo depende de la receptividad sino también de la conectividad. Es decir, cuanto más aislados estén los agentes, por más que cuente con elevadas probabilidades de receptividad, la probabilidad de adopción de SL será menor. Tanto el desarrollo como la utilización de SL requieren de un elevado grado de utilización de las herramientas de comunicación digital como de un compromiso importante hacia comunidades o grupos virtuales. Esta mayor participación en redes virtuales contribuye a la generación de competencias, no obstante, en este caso también es necesario un umbral mínimo inicial en la utilización de redes para que este proceso de creación y circulación de conocimiento se inicie. En este sentido resulta clave entender al conocimiento que circula por estas redes como un bien cuasi-público o como un bien club.

Este tipo de análisis redefine la forma de pensar las políticas de promoción de SL. Las opciones habituales son generalmente del tipo *top-down*. Es decir, presuponen que para lograr una mayor difusión del SL es necesario imponer estándares y obligar uso en ámbitos públicos. La visión que aporta aplicar la idea de percolación al proceso de adopción, lleva a pensar que este tipo de políticas encontraría serias limitaciones debido a la debilidad de las redes virtuales o la falta de compatibilidad entre las trayectorias de las capacidades tecnológicas de los desarrolladores locales y la trayectoria tecnológica propia del SL.

Más allá de la pertinencia de opciones de políticas del tipo *top-down*, de este estudio se desprende la necesidad de complementarlas con políticas del tipo *bottom-up*. Este tipo de políticas debería apuntar a la generación de competencias compatibles con las tecnologías libres así como incentivar al desarrollo de redes virtuales y uso herramientas de comunicación digital. De esta forma se lograría incrementar el grado de difusión de SL al crear las condiciones para que el grado de adopción de sea mayor.

BIBLIOGRAFÍA

- Ancori B., Bureth A. and Cohendet P., (2000). *The economics of knowledge: the debate about codification and tacit knowledge*, Industrial and Corporate Change, Vol 9, Nro 2.
- Antonelli C. (1997) *Percolation processes, technological externalities and the evolution of technological clubs*, Empirica, Kluwer Academic Publishers. Netherlands
- Antonelli, C., (1999) “*The evolution of industrial organization of the production of knowledge*”, Cambridge Journal of Economics, Vol 23.
- Arroyo, (2002) Sadio Newsletter N° 4.
- Arthur, W. B. (1988), *Competing Technologies: An Overview* En Dosi, Freeman, Nelson, Silverberg y Soete (eds.), Technical Change and Economic Theory, Pinter, London
- Arthur, W. B. (1989) *Competing Technologies, Increasing Returns and Lock-in by Historical Events* Economic Journal, 99, 116-131, 1989.
- Ayala Espino, J. (1999) Instituciones y economía. Una introducción al neo institucionalismo económico. Fondo de Cultura Económica. México.
- Baldwin C.Y. y Clark K.B., (2002) *Empirics of the Bazaar: Discussion*. Mimeo Toulouse Conference.
- Barlow J. P. (1994) *The Economy of Ideas A framework for patents and copyrights in the Digital Age* Wired Issue 2.03 - Mar 1994
- Benkler, Yochai (2002) *Coase's Penguin, or, Linux and the Nature of the Firm*. Yale Law Journal, 112, Winter 2002–2003.
- Bloch, M., (1991). “*Language, Anthropology and Cognitive Science*”, Man, New series, Vol 26 Nro 2.
- Borello, J.; Yoguel, G.; Erbes, A.; Robert, V.; Roitter, S. (2005) *Competencias tecnológicas de los trabajadores informáticos argentinos*. Revista de la CEPAL N° 87 Santiago de Chile.
- Casalet, Cimoli y Yoguel 2005 Redes, jerarquías y dinámicas productivas Ed. Miño y Dávila
- Chandler, A. D. (1977) The visible hand: the managerial revolution in American business The Belknap Press of Harvard University Press.
- Chaparro, E. (2003) *Every breath you take... Tecnología de la Información y Control Social* Sadio Newsletter N° 4.
- Chudnovsky, D., López, A. y Melitsko, S. *El sector de software y servicios informáticos (SSI) en la Argentina: Situación actual y perspectivas de desarrollo* CENIT DT 27/Julio de 2001.
- Coase, R.H. (1937); *The Nature of the Firm* *Economica* The London School of Economics and Political Science

- Cockburn, A. (1999) *Characterizing People as Non-Linear, First-Order Components in Software Development*. Humans and Technology Oct.1999, Salt Lake City, UT USA.
- Covarrubias, I.M.. (2002) Enfoque Sistémico e Individualismo Metodológico: una aproximación Universidad Centro Occidental Lisandro Alvarado Barquisimeto. Venezuela.
- Cowan R., David P., Foray D., (2000) *The explicit Economics of Knowledge Codification and Tacitness*, Industrial and Corporate Change, Vol 9 Nro 2.
- Dalle, J-M y David, PM. (2003) *The Allocation of Software Development Resources In 'Open Source' Production Mode* SIEPR Discussion Paper No. 02-27 March 2003
- Dalle, J-M y Jullien, N. (2003) *Turning Fads Into institution*. Policy Research N°32-2003
- David, P. A. (1985). *Clio and the Economics of QWERTY*, American Economic Review, American Economic Association, vol. 75(2).
- David, P. A. (2005) *The Beginnings and Prospective Ending of "End-to-End": An Evolutionary Perspective On the Internet's Architecture*, Industrial Organization 0502012, Economics Working Paper Archive at WUSTL.
- David, P. A. and Foray, D., (1994), *Percolation Structures, Markov Random Fields and the Economics of EDI Standard Diffusion*, en Global Telecommunications Strategies and Technological Changes Pogorel (Editor). North-Holland, Amsterdam.
- Dewatripont, M. Jewitt, I. y Tirole, J. (1999) *The Economics of Career Concerns, Part I: Comparing Information Structure*. Review of Economic Studies. Vol. 66 1 (January)
- Dosi G, Freeman C, Nelson RR, Silverberg G, Soete L (1988) Technological Change and Economic Theory Pinter, London.
- Dosi G, y Metcalfe, J.S (2003). *Aproximaciones a la irreversibilidad en la teoría económica*. En Ciencia tecnología y crecimiento económico. F.Chesnais y J.C. Neffa (Comps.). CEIL-PIETTE CONICET Trabajo y sociedad. Buenos Aires
- Ernst D. y Lundvall B-A, (1997) *Information Technology in the Learning Economy. Challenges for Developing Countries*, DRUID Working Paper Nro 97/12
- FLOSS (2002) *Free/Libre and Open Source Software: Survey and Study FLOSS: FINAL REPORT* International Institute of Infonomics University of Maastricht, The Netherlands Berlecon Research GMBH Berlin, Germany Junio 2002
- Fowler, M. (2003) *The New Methodology*. <http://martinfowler.com/>
- Freeman, C. y Perez, C.(1988) *Structural crises of adjustment, business cycles and investment behaviour* En Dosi, Freeman, Nelson, Silverberg y Soete (eds.), Technical Change and Economic Theory, Pinter, London.

- Frisch H.L., Hammersley J.M., (1963), *Percolation Process and Related Topics*, Journal of the Society for Industrial and Applied Mathematics, vol. 11, n° 4.
- García J.M.(2001) *INNOVATING WITHOUT MONEY: Linux and the Open Source Paradigm as an Alternative to Commercial Software Development* MSc Dissertation SPRU - Science and Technology Policy Research University of Sussex.
- Goody, J. (1977) *The Domestication of the Savage Mind*, Cambridge: Cambridge University Press.
- Gosh, R., Krieger, B., Glott, R. y Robles, G., (2002) *Free/Libre and Open Source Software: Survey and Study*, Infonomics, NL, June (<http://www.infonomics.nl/FLOSS/report>).
- Griliches, Z. (1957) *Hybrid Corn: An Exploration in the Economics of Technological Change* Econometrica, The Econometric Society
- Habermas, J (1988). *Teoría de la acción comunicativa*. 2 Tomos. Taurus Madrid
- Hann, H, Roberts, J y Slaughter S. (2002) *Delayed Returns to Open Source Participation: An Empirical Analysis of the Apache HTTP Server Project* Mimeo Toulouse Conference.
- Harris, M. (1995). Nuestra Especie Ed. Alianza. Madrid, España.
- Hawkings (2002) *The economics of open source for a competitive firm. Why give it away for free?* <http://opensource.mit.edu/papers>
- Hilbert, M (2001) *Latin America on its path into the digital age: where are we?* Serie: Desarrollo productivo n° 104. Cepal. Chile.
- IDA (2001) (Interchange of Data between Administrations, European Commission, DG Enterprise) *Study into the use of Open Source Software in the Public Sector Part 2 Use of Open Source in Europe* June 2001.
- Johnson B, Lundvall BA (1994) *Sistemas nacionales de innovación y aprendizaje institucional* Comercio Exterior.
- Katz y Shapiro (1968) *Technology Adoption in Presence of Networks externalities* Journal of Political Economics 94 822-41 1986.
- Katz, J. y Hilbert, M. (2003) Los caminos hacia una sociedad de la información en América Latina y el Caribe. Cepal Alfaomega. Bogotá.
- Langlois, R. 2003. *The vanishing hand: the changing dynamics of industrial capitalism* Industrial and Corporate Change, Vol 12, Nro 2.
- Lecocq, X. y Demil (2002) *Open standard: role of externalities and impact on the industry structure*. MIT SL PAPER
- Lerner, J. y Tirole, J. (2000) *The Simple Economics Of Open Source* NBER Working Paper Series, Working Paper 7600 <http://www.nber.org/papers/w7600>" Marzo 2000.

- Lessig, L. (2001) The future of ideas: the fate of the commons in a connected world, Random House. New York
- Lundvall A and Johnson B (1994) *Learnig economy*, Journal of industrial studies, Vol 1, Nro 2, december
- Lundvall, (1996); *The social dimension of learning economy*, DRUID Woking Paper Nro 96/01, Aalborg.
- Malerba F. y Orsenigo L., 2000. *Knowledge, Innovative activities and industrial evolution* Industrial and Corporate Change, Vol 9, Nro 2.
- Martinez, E. (2003) *Crear, tomar o comprar* Newsletter Sadio - Número 6
- Maus. M. (1960) The gift: Forms and functions of exchange in archaic societies Norton. New York.
- Metcalf (1988) *The diffusion of innovation: an Interpretative Survey*. En Dosi G, C. Freeman R. Nelson, G. Silverberg and L. Soete (eds.), Technical Change and Economic Theory, Pinter, London
- Micheli, Jordy (2003), *El trabajo de digitofactura en la economía postindustrial* en Espacios globales: espacios del capitalismo, Ed. Carmen Bueno. México.
- Moineau, L. y Papatheodorou, A. (2004) *Cooperación y producción inmaterial en el software libre* (<http://sindominio.net/biblioweb/>)
- Nelson, R. R., and Winter, S.G. (1982) An Evolutionary Theory of Economic Change. Cambridge: Harvard University Press.
- Nonaka I. y Takeuchi H., 1995 The knowledge creating company: How japannesse companies create the dynamics of innovation, London: Oxford University Press.
- Nooteboom, B. (1999) *Innovation, Learning and Industrial Organization* Cambridge Journal of Economics; N. 23.
- Polanyi, M., (1967) The tacit dimension , New York: Doubleday,.
- Raymond, Eric S. (1999a) *The Cathedral and the Bazaar*. En The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly & Associates, Inc., Sebastopol, CA.
- Raymond, Eric S. (1999b) *Homesteading the Noosphere*. En The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly & Associates, Inc., Sebastopol, CA.
- Raymond, Eric S. (1999c) *Brief History of Hackedom*. En The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly & Associates, Inc., Sebastopol, CA.
- Rosenberg, N. (1976), Perspectives on Technology, Cambridge University Press, New York.
- Rossi C y Bonaccorssi, (2004) *Comparing Motivations of individual programmers and firms to take part in the open source movement*. From community to business. Sant' Anna School of Advanced Studies. Pissa, Italia.

- Rullani E. 2000, El valor del conocimiento, en Territorio, conocimiento y competitividad de las empresas. El rol de las instituciones en el espacio global, Boscherini-Poma comp., Miño y Dávila ed., Madrid.
- Schank, R.C. and Abelson, R.P. (1977) Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures, Hillsdale, NJ. Lawrence Erlbaum.
- Silverberg, G. Dosi, G. y Orsenigo, L. (1988) *Innovation, Diversity and Difusión: a self-organisation model*. The Economic Journal N° 98. Dec. Gran Bretaña.
- Simon H A. (1962). *The Architecture of Complexity* Proceedings of the American Philosophical Society, Vol.106 No.6.
- Simon, H.A. (1982) *Models of bounded rationality*. MIT Press
- Stallman R. (1983) *FSF Manifesto*. www.fsf.org
- Stallman R.M. (2002) Free Software, Free Society: Selected Essays GNU Press
- Storper, M. (1989). *The Transition to Flexible Specialisation in the U.S. Film Industry: External Economies, the Division of Labour, and the Crossing of Industrial Divide*, Cambridge Journal of Economics, Oxford University Press, vol. 13(2).
- Valduriez 2002 Business models for OSS. Presentado en la Conferencia de Toulouse.
- Varian H. (2002) *System Reliability and Free Riding* University of California, Berkeley Mimeo Toulouse Conference.
- Villanueva Núñez 2002 Correspondencia entre el Señor J. A. González, Gerente General de Microsoft del Perú y E. D. Villanueva Núñez, congresista peruano. <http://www.gnu.org.pe/rescon.html>
- Von Bertalanffy, L. (1987) Historia y situación de la teoría general de sistemas. Ed. Alianza Universidad. Madrid.
- Von Hippel, Eric (2002) *Open source software projects as user innovation networks* MIT Sloan School of Management
- West, J. (2003) How open is open enough? Melding proprietary and open source platform strategies *Research Policy* 32 (2003) 1259–1285
- Williamson, O.E. (1985); The economic institutions of capitalism, Collier Macmillan Publishers. LONDON
- Yoguel G, Novick M, Milesi D, Roitter S y Borello J (2003), *Información y conocimiento: la difusión de TICs en la industria manufacturera argentina* *Información y conocimiento: la difusión de TICs en la industria manufacturera argentina*, Revista de la CEPAL N° 86, Santiago de Chile
- Yoguel, G. y Boscherini, F. (2001) *El desarrollo de las capacidades innovativas de las firmas y el rol del sistema territorial*. Desarrollo Económico - Revista de Ciencias Sociales. Vol. 41, N° 161, abril-junio 2001. IDES, Buenos Aires.

Yoguel, G., Novick M. y Fuchs, M. (2003) *Desarrollo de Redes de Conocimiento. El Caso de Lifa, Laboratorio de Investigación y Formación en Informática Avanzada de la Universidad Nacional de la Plata*. En Nuevas Tecnologías de Información y Comunicación. Los Límites en la Economía del Conocimiento. F. Boscherini, M. Novick y G. Yoguel (Comps.) Miño Dávila 2003.

Zeitlyn, David (2003) *Gift economies in the development of open source software: anthropological reflections*. Policy Research N° 32.

ANEXOS

Fuentes de información

Para la realización de este trabajo se contó con diversas fuentes de información de distinto grado de sistematicidad. No obstante, considerando la novedad de la problemática a nivel internacional y tanto más a nivel local, todas las fuentes fueron calificadas como de alto valor y por lo tanto todas ellas fueron tenidas en cuenta para la elaboración del presente trabajo.

La información puede agrupada en cuatro grandes grupos: (i) estudios previos, (ii) datos empíricos de fuentes primaria y secundaria, (iii) recopilaciones de noticias publicadas en diarios y revistas especializadas, sitios de Internet especializados, así como documentos oficiales y (iv) charlas y entrevistas no estructuradas con personas involucradas con el sector de SSI en general y con el SL específicamente.

Antecedentes

Dentro de este grupo cabe diferenciar entre los estudios previos que se hayan realizado sobre la problemática del SL (a nivel mundial) de los trabajos sobre el sector de SSI en el país. Actualmente no se han realizado trabajos locales cuya problemática central sea el SL desde una perspectiva de la economía o desde el desarrollo económico. Por lo tanto los antecedentes que se tuvieron en cuenta en la búsqueda bibliográfica se refieren principalmente a trabajos extranjeros. El fenómeno del SL involucra un amplio espectro de disciplinas (economía, sociología, antropología, psicología, derecho, entre otras) y por lo tanto las formas de afrontarlo son diversas. En la medida que el presente trabajo intenta hacer un abordaje a la problemática desde la economía se consideraron como antecedentes los estudios desde esta misma disciplina.

La difusión de estos trabajos vía Internet ha sido amplia, no obstante, caben mencionar algunas de ellas que se destacan. Por una parte el MIT, inspirado en el movimiento del

SL desarrollo un sitio desde el cual promueve el intercambio de investigaciones vinculadas con el SL. El sitio cuenta con de una base *on-line* trabajos y referencias de trabajos sobre SL. La página del MIT representa una fuente ineludible de consulta para el estudio de este tema (<http://www.opensource.mit.edu>).

Por otra parte, el instituto de industria de la Universidad de Toulouse organizó durante el 2002 un congreso sobre la economía del SL, presentándose en el mismo trabajos pioneros en esta problemática, como el de Lerner y Tirol (2002) que aparece como uno de los trabajos más citados referidos a SL.

El instituto Infonomics de la Universidad de Maastricht en conjunto con la Comisión Europea desarrolló un relevamiento electrónico a desarrolladores de SL para indagar sobre la cuestión de las motivaciones (FLOSS), también recopiló información sobre código fuente libre y sobre experiencias de instrumentación de SL en el Estado (fundamentalmente en los países de la UE). Sobre la base de este relevamiento se han escrito una serie de trabajos que fueron consultados. Este estudio ha tenido continuidad durante el 2004 y 2005 ampliándose a otros países y regiones del mundo. Si bien el proyecto tenía alcance global, está fuertemente asociado a la Unión Europea. El objetivo de este proyecto fue construir un repositorio de información sobre el fenómeno del SL y desarrollar indicadores que permitan evaluar el nivel actividad. En este caso, la muestra fue de 2208 respuestas. El BCG (Boston Consulting Grup) y OSTG (Open Source Technology Group), también llevaron a cabo una encuesta a desarrolladores en los EEUU durante 2001 con el objetivo de caracterizar la conducta de programadores que participan en desarrollos libres. Obtuvieron 684 respuestas válidas.

Por último cabe mencionar que en 2003 el Journal Research Policy ha dedicado un número especial a la temática del SL publicando un número importante de trabajos. Si bien estas cuatro fuentes de constituyen un marco de referencia obligatorio, existen otros aportes atomizados que pueden encontrarse en la red.

En lo que se refiere a los estudios locales, recientemente se han realizado importantes aportes sobre la dinámica de la industria del software, desde miradas multidisciplinarias. Estos trabajos constituyen un aporte valiosísimo para la comprensión de la dinámica del sector en la Argentina en tanto no existen estadísticas sistemáticas que describan

periódicamente su estructura y dinámica. Entre ellos cabe mencionar los trabajos realizados por el grupo de CENIT, que incluyeron diversas encuestas dirigidos al estudio del sector

Datos empíricos de fuentes primarias y secundarias

Al respecto cabe mencionar la realización de una encuesta electrónica a trabajadores informáticos que fue realizada en conjunto con el grupo de trabajo de la UNGS y la Sociedad Argentina de Informática e Investigación Operativa (SADIO). Esta encuesta que tenía como objetivo central el estudio de las capacidades de los trabajadores informáticos argentinos, contó con un apartado dedicado a evaluar la difusión de SL entre estos trabajadores. Las preguntas de este apartado fueron elaboradas en función de la prueba de hipótesis. Al mismo tiempo, las preguntas fueron construidas de forma de que pudieran ser comparables con los relevamientos realizados por la Universidad de Maastricht junto a la Comisión la Europea (FLOSS, 2002) y por el BCG durante el 2001. Ver más detalle más adelante en estos anexos.

Recopilaciones de noticias publicadas en diarios y revistas y sitios especializadas y documentos oficiales

En esta categoría se encuentran trabajos sobre el fenómeno del SL con perspectivas distintas al análisis económico. Estos trabajos pueden diferenciarse de acuerdo a los autores y al tipo de publicación que corresponda. Por un lado están los documentos escritos por personas que pertenecen a la comunidad de desarrolladores y han participado en la fundación de las instituciones que organizan y promueven el movimiento. Estos escritos tiene singular valor porque se trata de las primeras descripciones del movimiento y de documentos fundacionales de la comunidad. Mucho de ellos siguen las mismas reglas de producción ‘libre’ y ‘evolutiva’ donde los documentos finales resultan de contribuciones de diferentes personas. Entre ellos desatacan los documentos escritos por R. Stallman y E. Raymond (líderes de los movimientos de SL y OSS, respectivamente). Por otra parte, se encuentran algunos trabajos encarados desde el *business* que relatan

experiencias de migraciones hacia SL, modelos de negocios basados en SL, o la incursión de grandes empresas de SSI en el mundo de SL y sus estrategias de negocios. Estos trabajos ponen de manifiesto que el SL no es un fenómeno marginal circunscrito a informáticos y *hackers* sino que tiene fuertes incidencias sobre el mundo de los negocios y afecta al desempeño económico. Por otra parte, se encuentran los documentos oficiales como proyectos de ley, correspondencia entre las cámaras productoras y legisladores, cartas abiertas, discursos, etc. En estos casos, se trata de documentos que muestran la incidencia del fenómeno en la esfera política y los reclamos de los diferentes actores a que el Estado asuma una postura frente al movimiento. Por último, cabe mencionar la literatura que de aborda esta problemática desde su aspecto jurídico. Estos trabajos relacionan la problemática de los derechos de propiedad intelectual con el SL, y está fuertemente vinculados con el grupo anterior.

Charlas y entrevistas no estructuradas con personas involucradas con el sector de SSI en general y con el SL específicamente

Por último, se consideran las entrevistas y encuentros realizados con personas con informáticos (académicos y relacionados con la industria). Se han realizado en el marco de la presente investigación numerosas entrevistas con personas directa o indirectamente vinculadas al sector de desarrollo de software y al movimiento del SL, entre ellas, personas que protagonizaron el desarrollo del sector de software desde sus inicios en la Argentina, con los que diseñaron e instrumentaron planes de estudio y enseñanza de informática en el país, con empresarios y trabajadores del sector, con docentes universitarios y alumnos, con directores de institutos de investigación, con quienes diseñan y discuten actualmente la política científica y tecnológica del sector en la Argentina y con los fundadores de las instituciones que promueven el desarrollo y difusión de SL en la Argentina, con usuarios de SL, con empresarios que decidieron adoptarlo, con empresario que brindan servicio sobre SL. En las JAIIO 2004 (Jornadas de Informática e Investigación Operativa organizadas) por SADIO se presentó una versión preliminar de esta investigación que recibió valiosos comentarios.

Elección de categorías y variables aplicadas en el capítulo empírico de la investigación

Elección de la unidad de análisis: para el testeo de la cuarta hipótesis referida al procesos de difusión se consideró como unidad de análisis al usuario experto, es decir, el programador, desarrollador o, en términos más generales, trabajador informático.

Categorías y variables. La encuesta fue diseñada de modo que proveyera información relevante para la contrastación de las hipótesis a partir de una idea previa de los factores explicativos de la difusión de SL entre los trabajadores informáticos argentinos. Las categorías y las variables fueron difíciles de delimitar a priori. Sin embargo, a partir de los trabajos previos y teniendo en mente un modelo teórico se pudieron prever algunos rasgos de la categorización. Entre las variables que serían utilizadas en la construcción de indicadores destacan como variable dependiente: el grado de difusión de SL; mientras que en el grupo de las variables independientes estaría (i) la receptividad medida por el tipo de herramientas de desarrollo que dominan los trabajadores y (ii) la conectividad en términos de el uso de herramientas que favorezcan el desarrollo de redes virtuales, el grado e importancia del uso de herramientas de comunicación digital y el grado de pertenencia a comunidades virtuales.

Indicador de receptividad

Se aplicó la técnica estadística de análisis factorial de correspondencias múltiples.

El conjunto de variables activas seleccionadas para la construcción de los grupos fueron los lenguajes y tecnologías que dominan los trabajadores informáticos⁹⁵. Además, fueron

⁹⁵ Lenguajes: Ada, Asp, Delphi-Kylix, Assembler, C, C++, C#, Clipper - Xbase, Cobol, Fortran, Java, Perl, Visual basic, Python, Smalltalk, Lisp, Javascript, Haskell, Unixshell y otros. Tecnologías: PuntoNet, COM/DCOM/MTS, J2EE, Plataforma web, Web Services, XML, Programación multitier, J2ME u otro para microcomputadores, Mainframes, AS400, Middleware / mom, Transaction Servers (MQ series, biztalk, etc.), Programación orientada a objetos, Extreme Programming (métodos ágiles), Métodos formales, Function points, UML, Proceso Unificado,

consideradas una serie de variables ilustrativas que permitieran la caracterización del sendero de formación de capacidades de estos actores⁹⁶.

Salida Análisis de cluster:

Análisis estructural. De la lista de lenguajes y tecnologías sobre los que fueron consultados solamente fueron eliminadas aquellas herramientas con una tasa de respuesta inferior al 3%.

⁹⁶ La edad, el género, la ciudad de residencia, el tipo y sector de la organización para la que trabaja, el tipo de relación laboral, la rotación laboral, la experiencia laboral, el nivel de complejidad de los proyectos en los que se vieron involucrados, el dominio de inglés, la educación formal, la importancia de la autoformación, entre otras.

DESCRIPTION DE LA COUPURE 'b' DE L'ARBRE EN 5 CLASSES
 CARACTERISATION DES CLASSES PAR LES MODALITES

CARACTERISATION PAR LES MODALITES DES CLASSES OU MODALITES
 DE COUPURE 'b' DE L'ARBRE EN 5 CLASSES
 CLASSE 1 / 5

V.TEST	PROB.	POURCENTAGES		GLOBAL	MODALITES	
		CLA/MOD	MOD/CLA		CARACTERISTIQUES DES VARIABLES	
				48.62	CLASSE 1 / 5	
5.84	0.000	58.00	98.86	82.87	no domina	Programación multitier
5.67	0.000	61.36	92.05	72.93	no domina	XML
5.57	0.000	67.68	76.14	54.70	no domina	Plataforma web
5.36	0.000	59.29	94.32	77.35	no domina	javascript
5.29	0.000	74.63	56.82	37.02	no domina	Programación orientada a objetos
5.13	0.000	56.13	98.86	85.64	no domina	Extreme Programming
5.12	0.000	61.79	86.36	67.96	no domina	java
4.64	0.000	62.73	78.41	60.77	no domina	visual basic
4.42	0.000	55.13	97.73	86.19	no domina	J2EE
4.23	0.000	53.33	100.00	91.16	no domina	Mainframes
4.20	0.000	54.04	98.86	88.95	no domina	COM/DCOM/MTS
3.97	0.000	61.11	75.00	59.67	no domina	C++
3.78	0.000	55.41	93.18	81.77	no domina	asp
3.78	0.000	57.04	87.50	74.59	no domina	Web Services
3.74	0.000	54.14	96.59	86.74	no domina	PuntoNet
3.69	0.000	52.38	100.00	92.82	no domina	lisp
3.65	0.000	58.97	78.41	64.64	no domina	UML
3.58	0.000	53.80	96.59	87.29	no domina	smalltalk
3.37	0.000	54.73	92.05	81.77	no domina	Assembler
3.13	0.001	52.10	98.86	92.27	no domina	python
2.82	0.002	54.68	86.36	76.80	no domina	cobol
2.35	0.009	57.14	63.64	54.14	no domina	C

CLASSE 2 / 5

V.TEST	PROB.	POURCENTAGES		GLOBAL	MODALITES	
		CLA/MOD	MOD/CLA		CARACTERISTIQUES DES VARIABLES	
				12.15	CLASSE 2 / 5	
7.32	0.000	87.50	63.64	8.84	si domina	Mainframes
6.62	0.000	45.24	86.36	23.20	si domina	cobol
5.51	0.000	68.75	50.00	8.84	si domina	AS400
3.68	0.000	20.41	90.91	54.14	no domina	C
3.30	0.000	17.95	95.45	64.64	no domina	UML
2.99	0.001	26.67	54.55	24.86	si domina	Cliper xbase
2.61	0.005	17.59	86.36	59.67	no domina	C++

CLASSE 3 / 5

V.TEST	PROB.	POURCENTAGES			MODALITES	CARACTERISTIQUES DES VARIABLES
		CLA/MOD	MOD/CLA	GLOBAL		
				14.92		CLASSE 3 / 5
6.98	0.000	60.61	74.07	18.23	si domina	asp
6.27	0.000	66.67	59.26	13.26	si domina	PuntoNet
5.59	0.000	33.80	88.89	39.23	si domina	visual basic
5.38	0.000	65.00	48.15	11.05	si domina	COM/DCOM/MTS
4.33	0.000	39.02	59.26	22.65	si domina	javascript
3.95	0.000	26.83	81.48	45.30	si domina	Plataforma web
3.90	0.000	34.78	59.26	25.41	si domina	Web Services
2.98	0.001	21.05	88.89	62.98	si domina	Programación orientada a objetos
2.49	0.006	19.08	92.59	72.38	no domina	unixshell

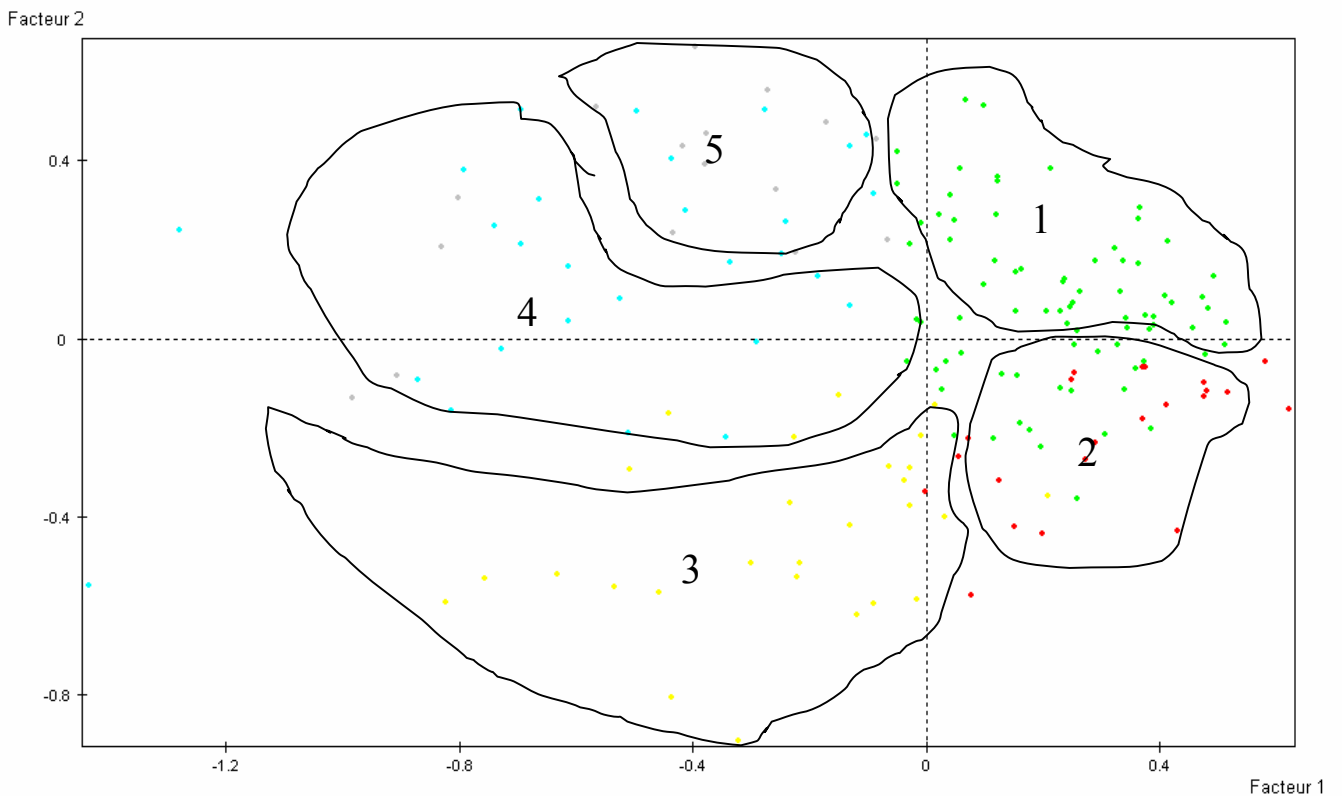
CLASSE 4 / 5

V.TEST	PROB.	POURCENTAGES			MODALITES	CARACTERISTIQUES DES VARIABLES
		CLA/MOD	MOD/CLA	GLOBAL		
				15.47		CLASSE 4 / 5
7.21	0.000	44.83	92.86	32.04	si domina	java
6.98	0.000	72.00	64.29	13.81	si domina	J2EE
5.66	0.000	54.84	60.71	17.13	si domina	Programación multitier
4.90	0.000	34.38	78.57	35.36	si domina	UML
4.87	0.000	53.85	50.00	14.36	si domina	Extreme Programming
4.52	0.000	28.92	85.71	45.86	si domina	C
4.34	0.000	36.73	64.29	27.07	si domina	XML
4.28	0.000	30.14	78.57	40.33	si domina	C++
4.20	0.000	23.68	96.43	62.98	si domina	Programación orientada a objetos
4.13	0.000	28.05	82.14	45.30	si domina	Plataforma web
3.31	0.000	34.15	50.00	22.65	si domina	javascript
2.98	0.001	30.00	53.57	27.62	si domina	unixshell
2.94	0.002	50.00	25.00	7.73	si domina	python
2.38	0.009	36.36	28.57	12.15	si domina	perl

CLASSE 5 / 5

V.TEST	PROB.	POURCENTAGES			MODALITES	CARACTERISTIQUES DES VARIABLES
		CLA/MOD	MOD/CLA	GLOBAL		
				8.84		CLASSE 5 / 5
6.90	0.000	84.62	68.75	7.18	si domina	lisp
5.99	0.000	52.17	75.00	12.71	si domina	smalltalk

3.95	0.000	18.07	93.75	45.86	si domina	C
3.38	0.000	42.86	37.50	7.73	si domina	python
3.38	0.000	27.27	56.25	18.23	si domina	Assembler
3.34	0.000	35.00	43.75	11.05	si domina	haskell
3.23	0.001	17.81	81.25	40.33	si domina	C++
2.88	0.002	20.41	62.50	27.07	si domina	XML
2.59	0.005	13.16	93.75	62.98	si domina	Programación orientada a objetos
2.52	0.006	27.27	37.50	12.15	si domina	perl



Indicador de conectividad

El indicador de conectividad se construyó considerando las siguientes preguntas del cuestionario referidas a uso de herramientas de comunicación digital y pertenencia a comunidades virtuales.

El grado de utilización de las herramientas de comunicación digital (GUHCD) fue calculado como la suma ponderada de la utilización: de Foros especializados (FE), Listas de correo internacionales (LCI), Listas de correo nacionales (LCN) y Newsletters (NL). Estas cuatro variables estaban en una escala de 1 a 5.

$$\text{GUHCD} = (\text{FE} + \text{LCI} + \text{LCN} + \text{NL}) / 20$$

El grado de pertenencia a las comunidades virtuales (GPCV) se calculó como una suma ponderada de la pertenencia a comunidades a través de internet (CV) y la participación en comunidades o grupos de usuarios (GU). Estas dos variables tomaban valores de 1 a 4.

$$\text{GPCV} = (\text{CV} + \text{GU}) / 8$$

El indicador de conectividad (IC) final surgió de la suma entre ambos indicadores previos.

$$(\text{IC}) = (\text{GUHCD} + \text{GPCV}) / 2$$

Variables de corte

Por último, cabe destacar un conjunto de variables de corte que contribuyeron a caracterizar la difusión de SL pero que no estuvieron especificadas en el modelo. Estas variables pueden ser consideradas como variables exógenas, entre ellas, la edad, el género, la ciudad de residencia, el tipo y sector de la organización para la que trabaja, el tipo de relación laboral, el nivel de complejidad de los proyectos en los que se vieron involucrados, el dominio de inglés, la educación formal, la importancia de la autoformación, entre otras.

Diseño de desarrollo del trabajo de campo

El capítulo empírico de esta investigación está basado en una encuesta electrónica auto administrada. El cuestionario (ver el apartado 0 de estos anexos) estuvo disponible en un sitio web para que fuera completado *on-line* por los desarrolladores. Esta metodología fue pensada como la mejor opción para recolectar la información dada la familiaridad de la

herramienta para el público objetivo y teniendo en cuenta la aplicación metodologías similares en las experiencias del proyecto FLOSS y del BCG. La encuesta estuvo orientada a trabajadores informáticos de firmas privadas, del sector público, de *training centers* y de universidades.

El desarrollo del cuestionario fue realizado de una forma multidisciplinaria en la que participaron investigadores con experiencia en el sector y actores vinculados estrechamente con la actividad. Básicamente participaron investigadores de la UNGS (Universidad Nacional de General Sarmiento) y miembros de SADIO (Sociedad Argentina de Informática e Investigación Operativa). En lo que respecta a la sección de la encuesta destinada al análisis de la difusión del SL, se consideraron además los relevamientos del FLOSS y del BCG de modo de poder extraer alguna conclusión comparable.

La encuesta estuvo compuesta por ocho partes: La primera parte apunta a recabar información sobre la situación laboral del encuestado. La segunda parte busca información acerca de la empresa para la que trabaja. La tercera parte apunta a lograr una descripción de las actividades específicas que realiza la persona encuestada. El cuarto apartado está enfocado a evaluar los proyectos que los trabajadores de la informática llevan adelante en su actividad laboral. La quinta parte indaga sobre la difusión de SL. El sexto apartado está orientado a medir las vinculaciones institucionales de los trabajadores informáticos y su participación en redes reales y virtuales. El séptimo apartado considera aspectos referidos a la formación de los trabajadores. Por último, el octavo apartado indaga sobre algunos datos personales del encuestado.

Como el relevamiento fue realizado por la Universidad en colaboración con SADIO. Ambas instituciones fueron utilizadas para promover la encuesta. Fueron contactadas telefónicamente cientos de empresas y miles de desarrolladores recibieron correos electrónicos invitándolos a participar del estudio.

Validación externa de la muestra

El alcance de la encuesta en términos de respuestas válidas obtenidas fue inferior al que se había supuesto inicialmente. El capítulo 5 de esta investigación está sustentado en 181 respuestas válidas. No obstante, diferentes criterios revelan que en la muestra existe suficiente heterogeneidad, lo que limita la presencia de sesgos.

La localización geográfica y el tipo de organización para la que trabaja fueron las dos dimensiones principales a tener en cuenta para asegurar la diversidad. Las respuestas recibidas no se circunscribieron a Buenos Aires y sus alrededores, sino que también llegaron respuestas del interior de la provincia y de otras provincias del país. Incluso contestaron argentinos residentes en el extranjero.

La variedad de organizaciones también estuvo garantizada. Las respuestas provinieron de departamentos de sistemas de oficinas del gobierno y de grandes empresas, de consultores independiente, de firmas dedicadas al desarrollo de software y servicios informáticos, de pequeñas y medianas firmas manufactureras y de firmas proveedoras de servicios públicos (agua, etc.). Como consecuencia, la estructura sectorial de la muestra y de del universo resultaron bastante similares. De acuerdo con el censo poblacional de 2001 el software desarrollado *in-house* alcanzaba al 56% de las personas que trabajaban en IT mientras que en nuestra muestra dicho porcentaje asciende al 51%.

En lo que respecta a la composición por nivel educativo existe un sesgo en la muestra a favor de los trabajadores más calificados: los desarrolladores con educación universitaria representan dos tercios de la muestra pero un tercio de la población. En la misma dirección la proporción de personas jóvenes (menos de treinta años) está sub-representada.

Encuesta sobre las competencias y capacidades del trabajador informático argentino. Proyecto TICs UNGS - SADIO⁹⁷

Parte I Datos sobre la situación laboral del encuestado

101: Su situación actual es:

Estoy temporariamente desocupado

Me dedico exclusivamente a estudiar pero tengo experiencia laboral

Me dedico exclusivamente a estudiar y no tengo experiencia laboral

Trabajo en la profesión

102: Indique cuántos años de experiencia laboral tiene como trabajador informático

103: ¿Cuántas veces ha cambiado de trabajo en los últimos 5 años?

104a: Indique los cargos que ocupa en la empresa o institución que constituye su actividad PRINCIPAL y las áreas en las que trabaja:

104b: Indique los cargos que ocupa en la empresa o institución que constituye su actividad SECUNDARIA y las áreas en las que trabaja:

105a: Tipo de organización para la que trabaja en su ocupación PRINCIPAL

Empresa privada

Institución académica o tecnológica

ONG

Sector público municipal

Sector público nacional

Sector público provincial

105b: Relación con la organización para la que trabaja su ocupación PRINCIPAL

Contratado por obras

Contratado por servicios

Empleado en relación de dependencia

Propietario o socio

105c: Tiempo de dedicación a su actividad PRINCIPAL (en horas semanales aproximadas)

106a: Tipo de organización para la que trabaja en su ocupación SECUNDARIA

Empresa privada

Institución académica o tecnológica

⁹⁷ La versión presentada en este anexo de la encuesta está ligeramente modificada por motivos de extensión (por ejemplos se obviaron la serie de condicionantes que se aplican a cada pregunta). No obstante, los cambios introducidos no alteran el sentido general de la misma.

ONG
Sector público municipal
Sector público nacional
Sector público provincial

106b: Relación con la organización para la que trabaja en su ocupación SECUNDARIA
Contratado por obra
Contratado por servicios
Empleado en relación de dependencia
Propietario o socio

106c: Tiempo de dedicación a su actividad SECUNDARIA (en horas semanales aproximadas)

Parte II Datos de las empresas o instituciones en las que trabaja

201a: Indique los siguientes datos de la empresa/institución que señaló como ocupación PRINCIPAL

201b: Indique los siguientes datos de la empresa/institución que señaló como ocupación SECUNDARIA

202a: ¿Cuántas personas en su actividad principal se dedican a desarrollo de software?

202b: ¿Cuántas personas en su actividad principal se dedican a servicios informáticos?

203: ¿La actividad principal de esta empresa o institución es proveer software y/o servicios informáticos?

Sí

No

204: ¿Cuál es el servicio o producto ofrecido por esta empresa?

Desarrollo de componentes
Desarrollo y venta de paquetes de aplicación
Desarrollos a medida
Hardware
Mantenimiento de hardware
Otros servicios
Servicios de asesoramiento
Servicios sobre hardware propio (ISP, Hosting)
Software de base
Soluciones integrales

205: ¿Qué tipos de aplicaciones desarrollan? (Por ejemplo software administrativo, contable, para control de procesos, etc)

206: Indique qué sector es el principal demandante software y servicios informáticos que ofrece la empresa o institución en la que trabaja. De existir más de uno mencione el que considere más importante

207: ¿En qué lugar desarrolla usted sus actividades?

En las oficinas de la empresa / institución para la que trabaja

En las oficinas del cliente

En su casa

208: Indique si existe y cuantas personas integran las siguientes áreas en la empresas en la que trabaja

Aseguramiento de la calidad

Desarrollo (programación)

Documentación

Implementación

Investigación

Marketing

Personas que efectúan más de una función

Soporte

Testing

Training

209: Indique a qué sector pertenece la empresa para la que trabaja:

Parte III Descripción de sus capacidades y actividades

301a: Marque el nivel de sus competencias técnicas en el área de la PROGRAMACIÓN (superior, alta, media, baja, no posee)

Área de aplicación

Capacidad de desarrollar estrategias de resolución de problemas

Capacidad para el diseño de programas o sistemas

Dominio de lenguajes o características de ambiente

Interfaces con el usuario

Manejo de técnicas de documentación

Manejo de técnicas de prueba y verificación

301b: Marque el nivel de sus competencias técnicas en el área de ANÁLISIS (superior, alta, media, baja, no posee)

Áreas de aplicación

Capacidad de comunicación con los demás

Capacidad de interpretar y abstraer situaciones, de modelizar de la realidad o de relacionar situaciones con modelos predeterminados

Capacidad de síntesis y diseño y para desarrollo de arquitectura de sistemas

Conocimiento de tecnologías (BD, OS / sist distribuidos)

Dominio de conceptos científicos de computación

301c: Marque el nivel de sus competencias técnicas en el área de la GERENCIAMIENTO (superior, alta, media, baja, no posee)

Capacidad de planificación y de control de procesos y sus etapas

Capacidad de relacionar situaciones con modelos tecnológicos o productos predeterminados

Capacidad para la toma de decisiones

Pensamiento orientado a relaciones económicas de costo-beneficio

301d: Marque el nivel de sus competencias técnicas en el área de la ADMINISTRACIÓN: (superior, alta, media, baja, no posee)

Actitud orientada al seguimiento y activación de actividades en pos de resultados

Capacidad de establecer y comunicar objetivos

Capacidad de liderazgo de otros

302a: ¿Qué actividades realiza actualmente en su ocupación principal referidas al desarrollo de software?

Análisis funcional de las necesidades del cliente
Análisis técnico de las necesidades del cliente
Control de la configuración
Diseño (de la arquitectura del sistema, de programas)
Diseño o administración de base de datos
Documentación
Gerenciamiento de proyecto
Implementación (instalación, capacitación, apoyo técnico)
Otros (asesoramientos o intervención sobre temas específicos)
Programación (escritura de código fuente)
Testing

302b: ¿Qué actividades realiza actualmente en su ocupación principal referidas los servicios informáticos?

Administración de redes (incluye también back-up)
Administración de seguridad y/o auditoria de sistemas
Asesoramiento tecnología (planificación, asesoramiento, determinación de estándares)
Capacitación
Customización de productos de otras empresas (SAP, IBM, etc)
Customización de productos propios
Gestión de producción (planificación de capacidad)
Mantenimiento de bases de datos
Servicios de asesoramiento
Soporte (help desk/ mesa de ayuda)

303: ¿Cómo consiguió su último trabajo?

por aviso publicado en el diario
por aviso publicado en Internet
por búsqueda de conocidos
por medio de la universidad
por recomendación de colegas
otros

304: Indique cuáles son los sistemas operativos que usa habitualmente (lo uso siempre, lo uso mayormente, lo uso muchas veces, lo uso a veces, no lo uso)

BSDs
DOS
GNU/Linux
Mac OS
OS para Mainframe
Unix comercial
Windows 95/98/Me
Windows NT/2000/XP
Otros

305: Marque cuáles de los siguientes lenguajes de programación domina y cuales utiliza en su trabajo:

ADA
ASP
Assembler
C
C#

C++
Clipper / xbase
Cobol
Delphi / Kylix
Fortran
Haskell
Java
Javascript
Lisp
Perl
Python
Smalltalk
Unix-shell
Visual Basic

306: Marque cuáles de las siguientes tecnologías domina y cuales utiliza en su trabajo:

Análisis estructural
AS400
COM/DCOM/MTS
Extreme Programming
Function points
J2EE
J2ME u otro para microcomputadores
Mainframes
Messaging (MQ series, biztalk, etc.)
Métodos formales
Middleware / mom
Plataforma web
Proceso Unificado
Programación multitier
Programación orientada a objetos
PuntoNet
UML
Web Services
XML

307: ¿Conoce los motivos por los que se eligió la plataforma (lenguaje, sistema operativo, bases de datos) que utiliza en su trabajo?

Sí
No

308: En caso positivo, indique si se explica por:

Capacidad de las personas que trabajan en la empresa / institución (conocimiento previo)
Costo global de la plataforma (licencias, costos laborales, etc)
Falta de alternativas
Historia previa (legacy)
Nivel tecnológico de la plataforma
Partner de la empresa que provee la plataforma
Requerimientos del cliente
Restricciones del mercado local
Soporte y servicios que ofrece el proveedor de la plataforma
Otros

309: ¿En que grado considera usted que su trabajo es estimulante para su desarrollo profesional?

nada estimulante

poco estimulante

algo estimulante

medianamente estimulante

muy estimulante

Parte IV Descripción de un proyecto de desarrollo

Responda este apartado con la información correspondiente al proyecto de desarrollo que usted considere más importante en el que se haya visto involucrado en los últimos 8 años.

401: ¿Entre las actividades laborales que realiza en el campo de la informática, alguna se encuadra dentro de algún proyecto de desarrollo?

Sí

No

402: Caracterice brevemente el objeto del proyecto elegido y sus principales rasgos.

403: ¿Qué función desempeña (o desempeñó) dentro del proyecto?

Análisis funcional de las necesidades del cliente

Análisis técnico de las necesidades del cliente

Control de la configuración

Diseño (de la arquitectura del sistema, de programas)

Diseño o administración de base de datos

Documentación

Gerenciamiento de proyecto

Implementación (instalación, capacitación, apoyo técnico)

Otros (asesoramientos o intervención sobre temas específicos)

Programación (escritura de código fuente)

Testing

404: ¿En qué año se inició el proyecto?

405: ¿Cuál es (o fue) el sector demandante del proyecto?

406a: ¿Se trata de un proyecto interno o desarrollado para terceros?

406b: ¿Fue desarrollado para un cliente en la Argentina o para un cliente en el extranjero?

406c: ¿Fue desarrollado para ser posteriormente comercializado o por un encargo específico de un cliente?

407: Indique cuál es (o fue) la criticidad del software en relación al objeto al que está destinado por ejemplo: severidad del daño que podría producirse ante una eventual falla

algo crítico

medianamente crítico

muy crítico

nada crítico

poco crítico

408: Indique cuál fue la duración planificada del proyecto (en meses)

409: Responde solo en casos de proyectos concluidos. Indique estimativamente cuál fue la duración efectiva del proyecto (en meses)

410: Cuántas personas estuvieron (o están) involucradas en el proyecto:

411: Indique el grado de dificultad técnica del proyecto seleccionado (considerando, entre otras cuestiones, el diseño, la elegancia, la eficiencia y la modularización del software).

Dificultad alta
Dificultad media
Dificultad media alta
Dificultad reducida
Sin dificultad

412: Indique el grado de complejidad de las competencias técnicas requeridas para el desarrollo del proyecto (considerando desde el análisis de las necesidades del cliente, hasta la venta y los servicios posventa).

algo complejo
medianamente complejo
muy complejo
nada complejo
poco complejo

413: Si el proyecto exigió vincularse con productos o servicios de grandes empresas de informática, en qué medida considera usted que esto fue importante para actualizarse tecnológicamente.

algo importante
medianamente importante
muy importante
nada importante
no requirió vincularse
poco importante

Parte V Difusión de software libre o de código abierto en la Argentina

Entendemos que software libre (free software) no es sinónimo de software de fuente abierta (open source software). Sin embargo, a los fines de este estudio, no resulta conveniente tal diferenciación, por lo tanto nos referiremos a este dominio con las siglas OS/FS (open source/free software) para indicar ambos casos.

501: ¿Cómo calificaría usted al grado de utilización de soluciones OS/FS?

Ninguno
Bajo
Alguno
Medio
Alto

502: La utilización de OS/FS en su trabajo, es una decisión de:

consecuencia de la infraestructura de sistemas ya existente del cliente o mercado para el cual es el software
la empresa/institución para la que trabaja (decisión económica)
la empresa/institución para la que trabaja (decisión técnica)
la empresa/institución para la que trabaja (política comercial)
suya porque no hay requerimientos específicos al respecto

otros

503: ¿Qué tipo de software libre utiliza? Nombre cuáles en cada caso

504: La decisión de utilizar soluciones OS/FS se basa en criterios múltiples, ¿podría ordenarlos de acuerdo a su importancia? Ordenar de 1 a 11 (uno es más importante)

- Bajo costo
- Disponibilidad del código fuente
- Estabilidad
- Existencia de redes de usuarios
- Funcionalidad
- Interoperabilidad
- Pertenencia a una comunidad
- Razones políticas
- Respeto los estándares
- Seguridad
- Otros

505: ¿Usted diría que el OS/FS satisface los requerimientos de hoy de mejor o peor forma que el software propietario?

El OS/FS resulta más efectivo

El software propietario resulta más efectivo

No sé

No tiene nada que ver una cosa con la otra

506: ¿Usted participa en desarrollos OS/FS?

Sí

No

507: ¿En qué año comenzó a desarrollar software abierto?

508: ¿Cuántas horas por semana aproximadas le dedica Ud. al desarrollo de OS/FS?

510: Ordene de acuerdo a su importancia las razones por las que usted dedica tiempo y esfuerzo al desarrollo de OS/FS. Ordenar de 1 a 11 (uno es más importante)

- para compartir mis conocimientos y habilidades
- para desarrollar nuevas habilidades
- para hacer dinero
- para mejorar mis oportunidades de trabajo
- para mejorar productos OS de otros desarrolladores
- para participar en la escena del open source
- para poner limite al poder de las grandes compañías
- para resolver un problema que no pudo resolver el software propietario
- porque creo que el software no debería ser un producto propietario
- porque creo que es el sistema que predominará en el futuro
- no sé

511: ¿Usted gana dinero (directa o indirectamente) desarrollando esta actividad?

No

Sí, directamente: me pagan para desarrollar OS/FS

Sí, directamente: me pagan por administrar OS/FS

Sí, directamente: me pagan por hacer soporte a OS/FS

Sí, directamente: por otras razones

Sí, indirectamente: las tareas en mi trabajo no incluyen el desarrollo de OS/FS pero desarrollo OS/FS en mi trabajo

Sí, indirectamente: obtuve mi trabajo por mi experiencia previa en OS/FS

Sí, indirectamente: por otros motivos

512: ¿En cuántos proyectos de OS/FS ha participado?

513: ¿En cuántos proyectos de OS/FS participa en la actualidad?

514: Por favor, ordene de acuerdo a su importancia las razones por las cuales no utiliza soluciones OS/FS en su trabajo. Ordenar de 1 a 9 (uno es más importante)

Desconfianza

El costo de utilizar OS/FS (costo de migración, costo de aprendizaje) es muy alto

Falta de personal capacitado para la instrumentación de este tipo de soluciones

Falta de servicios de ayuda y mantenimiento

No conozco soluciones que puedan aplicarse OS/FS a mi trabajo

No es compatible con la plataforma propietaria que utilizo

No estoy interesado

No sé qué es el OS/FS

Política de la empresa

Parte VI Redes y vinculaciones institucionales

601: Indique si se apoya o encuentra información útil para el desarrollo de sus tareas en: (lo uso siempre, lo uso mayormente, lo uso muchas veces, lo uso a veces, no lo uso)

Comunidades temáticas más allá de Internet

Foros especializados

IRCs (Internet Relay Chat)

Listas de correo internacionales

Listas de correo nacionales

Newsletters

602: Por favor, indique las publicaciones técnicas que lee con más frecuencia para mantenerse actualizado:

603: Por favor, indique qué buscadores o meta buscadores utiliza con mayor frecuencia para mantenerse actualizado:

604: Por favor, indique los sitios de Internet específicos que utiliza con mayor frecuencia para mantenerse actualizado:

605: El Intercambio de experiencias con colegas y amigos le resulta de utilidad para:

Ordene de acuerdo a su importancia. Ordenar de 1 a 8 (uno es más importante)

conseguir empleos

conseguir recursos humanos

dimensionar el mercado (salarios, condiciones de trabajo)

enterarse de nuevas tecnologías

enterarse de oportunidades de capacitación

saber que usan las empresas locales

no intercambio experiencias de ese tipo
otras

606: Indique en la siguiente tabla las instituciones en las que forma parte, en las que formó parte pero ya no y en las que tiene interés en asociarse (Formó parte pero ya no, Forma parte, Está interesado en asociarse)

AADS

Asociaciones de profesionales de desarrollo

Asociaciones y/o cámaras de empresas de software

Comunidades virtuales (a través de Internet)

Consejos profesionales

Grupos de usuarios de productos (ie: LUG.ar)

IEEE-CS

SADIO

607: Desde que año esta asociado a SADIO

608: ¿Conoce las actividades que lleva a cabo SADIO?

Si

No

609: ¿Alguna vez participó de alguna de las siguientes actividades organizadas por SADIO?

Bibliotecas

Conferencias

Cursos de capacitación profesional

Debates

Foros

Grupos o comités técnicos

Jornadas JAIIO

Lista infosociedad

Otras reuniones técnicas nacionales o internacionales

Otras

610: ¿Con qué frecuencia consulta los siguientes servicios brindados por SADIO? (lo uso siempre, lo uso mayormente, lo uso muchas veces, lo uso a veces, no lo uso)

Bolsa de trabajo

EJS (Electronic Journal of SADIO)

Lista infosociedad

Newsletter

Web

Otras

611: ¿Cuáles de las siguientes actividades le parece que debería desarrollar SADIO?

Debe actuar como "nodo" conector de intereses específicos

Debe ayudar a sus miembros a realizar contactos internacionales

Debe enviar información de divulgación (Talleres/ Congresos, etc)

Debe enviar información especializada sobre el sector informático en la Argentina

Debe enviar información sobre nuevas publicaciones

Otras

Parte VII Formación

701: Por favor, complete con información sobre sus estudios secundarios:

702: Por favor, complete con información sobre sus estudios terciarios:

703: Por favor, complete con información sobre sus estudios universitarios:

704: Por favor, complete con información sobre sus estudios de posgrado:

705: ¿Ha obtenido certificaciones?

Cisco

Microsoft

Novel

Oracle

Otros (IBM, Sans, IT-college, etc)

Sunmicrosystems

706: ¿En qué ámbito considera que obtuvo la mayor parte de los conocimientos que utiliza en su trabajo principal actual? Ordenar de 1 a 6 (uno es más importante)

Autodidacta

En los estudios mencionados en las preguntas anteriores

En otros cursos de capacitación

En su trabajo actual

En sus trabajos anteriores

Otro

707: ¿Cuál es su nivel de inglés?

Bueno

Excelente

Malo

Muy bueno

Regular

Parte VIII Datos personales

801a: Nombre y apellido:

801b: Teléfono:

801c: Dirección de correo electrónico:

802: Genero:

803: Edad:

21-30

31-40

41-50

más de 50

804: Ciudad y país de residencia:

805: ¿Estaría dispuesto a colaborar a posteriori para responder alguna entrevista en profundidad en temas específicos?

Sí
No

Anexo estadístico al capítulo 5

Tabla 0.1 Relación entre grado de utilización de SL y edad

Grado de utilización de SL	Edad				Total
	menos de 30	de 31 a 40	de 41 a 50	más de 50	
Ninguno	28,3%	30,4%	28,3%**	13,0%	100%
Alguno	51,7%	37,9%	3,4%**	6,9%	100%
Bajo	33,3%	39,6%	20,8%	6,3%	100%
Medio	28,6%	53,6%*	14,3%	3,6%	100%
Alto	66,7%***	20,0%	10,0%	3,3%	100%
Total	39,8%	35,4%	17,1%	7,2%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 2,7%; * Prueba Z significativa al 10%; ** Prueba Z significativa al 5%; *** Prueba Z significativa al 1%.

Tabla 0.2 Relación entre grado de utilización de SL y género

Grado de utilización de SL	Género		
	Femenino	Masculino	Total
Ninguno	21,7%	78,3%	100%
Alguno	13,8%	86,2%	100%
Bajo	14,6%	85,4%	100%
Medio	17,9%	82,1%	100%
Alto	6,7%	93,3%	100%
Total	15,5%	84,5%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS.

Tabla 0.3 Relación entre grado de utilización de SL y ciudad de residencia

Grado de utilización de SL	Ciudad de residencia			
	Buenos Aires y Gran Buenos Aires	Interior del país	Residentes en el exterior	Total
Ninguno	69,6%	28,3%	2,2%	100%
Alguno	44,8%	34,5%	20,7%	100%
Bajo	68,8%	29,2%	2,1%***	100%
Medio	39,3%	57,1%***	3,6%	100%
Alto	53,3%	40,0%	6,7%	100%
Total	58,0%	35,9%	6,1%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 0.4%; *** Prueba Z significativa al 1%.

Tabla 0.4 Relación entre grado de utilización de SL y experiencia laboral

Grado de utilización de SL	Experiencia laboral				Total
	menos de 10	entre 11 y 20	entre 21 y 30	más de 30	
Ninguno	37%*	37,00%	21,7%*	4,30%	100%
Alguno	86,2%***	6,9%***	0%**	6,90%	100%
Bajo	43,80%	39,60%	12,50%	4,20%	100%
Medio	50,00%	32,10%	14,30%	3,60%	100%
Alto	69%*	17,20%	10,30%	3,40%	100%
Total	53,90%	28,90%	12,80%	4,40%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 0.8%; * Prueba Z significativa al 10%; ** Prueba Z significativa al 5%; *** Prueba Z significativa al 1%.

Tabla 0.5 Relación entre grado de utilización de SL y nivel de educación formal

Grado de utilización de SL	Nivel de educación formal			
	Terciaria	Universitaria	Posgrado	Total
Ninguno	13,6%*	77,3%	9,1%*	100%
Alguno	0,0%	82,1%	17,9%	100%
Bajo	10,9%	76,1%	13,0%	100%
Medio	3,6%	60,7%	35,7%***	100%
Alto	3,3%	80,0%	16,7%	100%
Total	7,4%	75,6%	17,0%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 5%; * Prueba Z significativa al 10%; *** Prueba Z significativa al 1%.

Tabla 0.6 Relación entre grado de utilización de SL y tipo de organización

Grado de utilización de SL	Tipo de organización en clases			
	Empresa Privada	Inst. académica o de investigación	Sector público	Total
Ninguno	65,9%	12,2%	22,0%	100%
Alguno	78,3%	0,0%	21,7%	100%
Bajo	71,7%	4,3%	23,9%	100%
Medio	48,1%	22,2%***	29,6%	100%
Alto	61,5%	11,5%	26,9%	100%
Total	65,6%	9,8%	24,5%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 20%; *** Prueba Z significativa al 1%.

Tabla 0.7 Relación entre grado de utilización de SL y tipo de relación laboral

Grado de utilización de SL	Tipo de relación laboral			
	Empleado en relación de dependencia	Propietario	Contratado	Total
Ninguno	57,10%	19,00%	23,80%	100%
Alguno	61,90%	9,50%	28,60%	100%
Bajo	55,60%	22,20%	22,20%	100%
Medio	59,30%	18,50%	22,20%	100%
Alto	52,00%	20,00%	28,00%	100%
Total	56,90%	18,80%	24,40%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS.

Tabla 0.8 Relación entre el grado de utilización de herramientas de comunicación digital y grado de utilización de SL

Uso de herramientas de comunicación digital	Grado de utilización de SL					
	Ninguno	Alguno	Bajo	Medio	Alto	Total
Muy Bajo	46,2%*	0,0%	23,1%	23,1%	7,7%	100%
Bajo	22,4%	23,9%*	29,9%	10,4%	13,4%	100%
Medio	30,2%	15,1%	30,2%	9,4%	15,1%	100%
Alto	20,7%	17,2%	13,8%	34,5%***	13,8%	100%
Muy Alto	15,8%	0%*	26,3%	15,8%	42,1%***	100%
Total	25,4%	16,0%	26,5%	15,5%	16,6%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 0.09%; * Prueba Z significativa al 10%; *** Prueba Z significativa al 0.9%.

Tabla 0.9 Relación entre la pertenencia a grupos y comunidades virtuales y grado de utilización de SL

Pertenenencia a grupos y comunidades virtuales	Grado de utilización de SL					
	Ninguno	Alguno	Bajo	Medio	Alto	Total
Muy Bajo	30,7%	13,3%	29,3%	13,3%	13,3%	100%
Bajo	0%*	22,2%	33,3%	44,4%***	0,0%	100%
Medio	38,1%	23,8%	19,0%	4,8%	14,3%	100%
Alto	24,5%	20,4%	24,5%	18,4%	12,2%	100%
Muy Alto	11,1%*	7,4%	25,9%	14,8%	40,7%***	100%
Total	25,4%	16,0%	26,5%	15,5%	16,6%	100%

Fuente. Elaboración propia en base a la Encuesta SADIO-UNGS. Nota: la probabilidad de aceptar la hipótesis nula de ausencia de asociación es de 0.09%; * Prueba Z significativa al 10%; *** Prueba Z significativa al 1.9%.