

Propiedad intelectual y software. Una mirada cualitativa sobre los mecanismos de apropiabilidad de los inputs y outputs en el sector SIS

Guillermina Yansen 1 y Mariano Zukerfeld 2.

1 Becaria CONICET Equipo de estudios sobre Tecnología, Capitalismo y Sociedad (e-TCS), e-tcs.org
Centro Ciencia, Tecnología y Sociedad (CCTS), Universidad Maimónides. Ciudad de Buenos Aires.
guillerminayansen@e-tcs.org;

2 Investigador CONICET Equipo de estudios sobre Tecnología, Capitalismo y Sociedad (e-TCS), e-tcs.org
Centro Ciencia, Tecnología y Sociedad (CCTS), Universidad Maimónides. Ciudad de Buenos Aires.
marianozukerfeld@e-tcs.org

Abstract Esta ponencia se interesa en estudiar la relación de los diversos procesos productivos de software con la propiedad intelectual y, de manera más general, con los diversos mecanismos de apropiabilidad. Esto implica distinguir dos planos. Por un lado, se trata de estudiar las formas en la que los diversos procesos regulan sus outputs: ¿en qué medida perciben ingresos por las licencias, generan ingresos a través de la venta de servicios, regulan sus productos mediante licencias GPL o similares, carecen de regulación explícita, etc.? En otras palabras, ¿qué importancia tiene la propiedad intelectual como medio para favorecer la apropiabilidad de los productores locales de software? Este primer aspecto ha sido, en cierta medida, estudiado. Sin embargo, el segundo elemento que queremos introducir ha sido menos considerado. Se trata de la relación con la propiedad intelectual del software *que utilizan como insumo* estos procesos productivos. Aquí, partiendo de una distinción tripartita entre software privativo legal, privativo ilegal y de código abierto, nos interesa indagar respecto de qué proporciones de cada uno de esos tipos son usados en los distintos procesos productivos.

Palabras clave: software. apropiabilidad propiedad intelectual

1. Introducción

Esta ponencia se interesa en estudiar la relación de los diversos procesos productivos de software con la propiedad intelectual y, de manera más general, con los diversos mecanismos de apropiabilidad. Esto implica distinguir dos planos. Por un lado, se trata de estudiar las formas en la que los diversos procesos regulan sus outputs: ¿en qué medida perciben ingresos por las licencias, generan ingresos a través de la venta de servicios, regulan sus productos mediante licencias GPL o similares, carecen de regulación explícita, etc.? En otras palabras, ¿qué importancia tiene la propiedad intelectual como medio para favorecer la apropiabilidad de los productores locales de software? Este primer aspecto ha sido, en cierta medida, estudiado. Sin embargo, el segundo elemento que queremos introducir ha sido menos considerado. Se trata de la relación con la propiedad intelectual del software *que utilizan como insumo* estos procesos productivos. Aquí, partiendo de una distinción tripartita entre software privativo legal, privativo ilegal y de código abierto, nos interesa indagar respecto de qué proporciones de cada uno de esos tipos son usados en los distintos procesos productivos. A su vez: ¿qué relación hay entre el tratamiento que se le da a los inputs de software y el que se da a los outputs? Esto es, las empresas que lucran con licencias ¿utilizan software que han adquirido mediante licenciamientos similares? Las firmas que ofrecen software de código abierto, ¿lo utilizan como su insumo? Por otra parte, resulta relevante averiguar en qué medida las esferas de software de código abierto y privativo ilegal colaboran con las ganancias de productividad de las firmas.

Antes estas y otras preguntas similares, nuestro trabajo se plantea dos hipótesis. La primera de ellas es que los tratamientos regulatorios tanto de los inputs como de los outputs varían de acuerdo al tipo de proceso productivo. Nos basamos, para ello, en un trabajo previo (vid. infra). La segunda hipótesis es que, en la mayoría de los casos, el tratamiento que se da al software como input es fuertemente *path*

dependence: no se rige tanto por un cálculo de costos y beneficios, ni por una perspectiva ideológica determinada (que puede o no estar presente) sino, ante todo, por la trayectoria previa –en términos de técnicas o competencias- de los sujetos que integran el proceso productivo.

Metodológicamente, esta investigación se basa en una muestra no probabilística de 25 casos. Se realizaron, entre septiembre y diciembre de 2010 en la Ciudad de Buenos Aires, 24 entrevistas en profundidad a productores de software y una a un informante clave. La muestra fue *intencional* y, privilegió la heterogeneidad del subsector, procurando la *saturación teórica*. En efecto, en otros trabajos señalamos las limitaciones de varios de los abordajes existentes en tanto confunden al amplio y variado universo de la producción de software con alguna de sus manifestaciones particulares (típicamente, la que ocurre en empresas multinacionales). A su vez, propusimos una categorización exhaustiva de esos procesos productivos (Segura, Yansen y Zukerfeld, 2011), distinguiendo entre: 1. Producción Mercantil de Software, (subdividida entre procesos Autónomos –micro, pyme y grandes- y heterónomos –filial, en red o escalonada). 2. Producción Mercantil de Otros bienes y servicios 3. Producción Estatal 4. Producción Académica 5. Producción en ONG's 6. Producción No Laboral (Software Libre y Producción Colaborativa). Aunque no todas las categorías tienen la misma relevancia, y considerando que algunos de los entrevistados participaban en más de un tipo de proceso productivo, la muestra incluyó al menos dos casos de cada una de las categorías mencionadas. A su vez, se priorizó la heterogeneidad en los roles y jerarquías.

La ponencia está organizada como sigue. En la segunda sección presentamos el marco teórico que utilizaremos. La tercera sección analiza el tratamiento dado por los diversos procesos productivos a los inputs de software. La cuarta discute la regulación de los outputs. Finalmente comparamos ambas cuestiones en las conclusiones.

2. Marco teórico

Desde mediados de la década de 1970 el sistema capitalista mundial exhibe de manera nítida un conjunto de profundas transformaciones. Siguiendo a Manuel Castells, entendemos que ocurre un pasaje desde el Capitalismo Industrial hacia el Capitalismo Informacional (Castells, 2006). Naturalmente, este tránsito implica enormes cambios en los procesos productivos y, ciertamente, en la relación entre los mecanismos de apropiabilidad y tales procesos productivos. Para analizar cómo esas transformaciones afectan a la producción de software en particular, es necesario introducir algunas nociones. Los procesos productivos informacionales, como otros cualesquiera, se han analizado en la bibliografía en base a dos enfoques: el sectorial y el laboral. Sobre ambos hay un conjunto importante de antecedentes. Aquí nos basta, apenas, con recuperar algunos conceptos que hemos discutido en trabajos previos. Ellos parten de una perspectiva materialista respecto de los flujos de conocimientos y de una crítica a otros aportes conceptuales (especialmente a la inclusión de estas modalidades en el sector servicios). Llamamos *bienes informacionales primarios* a aquellos que están hechos puramente de información digital (música, películas, textos, datos y, ciertamente, software). Los *bienes informacionales secundarios*, en cambio, son aquellos que procesan, transmiten y/o almacenan información digital (como PC's, smartphones u otras tecnologías digitales). Denominamos *Sector Información* a aquél compuesto por el conjunto de unidades productivas cuyo output principal son bienes informacionales primarios. Se trata de un cuarto sector, que se adiciona a la división usual entre agricultura, industria y servicios. Algunos ejemplos de esas unidades productivas son los estudios de grabación de música, las consultoras, los call centers, las microempresas dedicadas al diseño gráfico, las firmas abocadas al diseño industrial, y las empresas de software (pero también las unidades productivas no mercantiles, como las redes que elaboran software libre), entre otras. En cambio, usamos la categoría de *Trabajo Informacional* para referir a las *tareas* productivas de aquellos trabajadores que utilizan como principal medio de trabajo un bien informacional secundario y que obtienen como producto un bien informacional primario. Diseñadores gráficos, 'data entry', músicos, periodistas y, por supuesto, programadores, representan ejemplos de este tipo de trabajadores.

Por supuesto, toda la producción que tiene lugar en el sector información registra problemas de apropiabilidad. En una línea que, entre otros, incluye a Schumpeter (1950), Arrow (1962), Nelson y Winter, (1982) y, específicamente, Teece (1986), Levin et al (1987) y Cohen et al (2000), los modos a través de los cuales las firmas logran apropiarse –o no- de los frutos de sus de las innovaciones han sido ampliamente conceptualizados. Numerosos estudios empíricos se han hecho desde esas perspectivas (un resumen de esos estudios puede verse en López, 2009).

Sin embargo, el software y los bienes informacionales en general no han sido los ejes centrales de esos trabajos, más enfocados en procesos industriales (Davis, 2002). Desde otras corrientes, el trabajo de Varian y Shapiro (2000) constituye un mojón importante en este análisis estableciendo la relevancia de las

externalidades de redes, el lock in y fenómenos similares. Desde el management, los textos para emprendedores como los de Chris Anderson (2006, 2009) son relevantes para describir las estrategias efectivamente seguidas por los productores de bienes informacionales, especialmente los que lo hacen a través de la web. En particular, algunos fenómenos específicos del mundo del software –como los mecanismos de apropiabilidad en relación a la producción de software libre-, están siendo crecientemente estudiados (Lakhani y Wolf, 2005; Tapscott y Williams, 2007).

La piratería, esto es, la apropiación ilegal e impaga de software, es usualmente discutida en términos legales o morales. No obstante, hay algunos trabajos económicos que arrojan resultados interesantes. Un resumen de la literatura económica sobre la piratería de bienes informacionales puede encontrarse en Peitz y Waelbroeck (2006), aunque tal resumen se restringe a los usos no mercantiles de esos BI (lo que será ubicado más abajo en el cuadrante 4). Respecto de esa piratería *end-user*, varios trabajos encuentran que, *si se consideran las externalidades de redes*, el titular del software podría perjudicarse con leyes de propiedad intelectual muy restrictivas (Conner & Rumelt., 1991; Takeyama, 1994; Slive y Bernhardt, 1998; Shy y Thisse, 1999).

Hay también algunos textos dedicados específicamente a la piratería de BI con fines de lucro (Bernhardt, 1998; Banerjee, 2003). Por ejemplo, Poddar (2005) muestra, a través de un modelo, que la piratería con fines de lucro, a diferencia de la del usuario final, no genera ventajas en ningún caso para los titulares del software.

Con todo, estos textos no lidian en detalle con una forma de piratería intermedia entre la reventa del producto y el uso por parte del consumidor final: la utilización del software ilegal como input, por parte de una firma que luego mercantilizara otro software. Numerosos estudios señalan el vínculo inverso entre el nivel de piratería y el tamaño de la producción local de software. Sin embargo, tienden a acentuar la idea de que desterrar la piratería favorecerá a la industria local cuando, a priori, bien puede sospecharse que es el crecimiento de las firmas locales –cuya productividad aumenta con el software pirata-, el que, en algún momento, las lleva a superar un umbral a partir del cual resulta conveniente reconocer los DPI de terceros, porque se requiere que se reconozcan los propios (Zukerfeld, 2010: Conclusión). De hecho, el tema es usualmente evitado de modo activo. Por ejemplo, en una sistematización de 28 estudios en los que se analizaban las causas de la piratería de software, Karakaya y Ulutürk (2010), encuentran 14 motivos. Varios son redundantes y vagos, pero sólo uno de ellos apunta a los “beneficios de la piratería”, y apenas se menciona un estudio de 1998 apuntalando ese motivo (Gopal y Sanders, 1998)

A su vez, los estudios que analizan las causas de la piratería toman como unidades de análisis a los individuos, mayormente, o a los países, en menor medida. Llamativamente, nuestra limitada revisión bibliográfica no encontró textos que analicen detalladamente los motivos de las firmas para adoptar software ilegal. En este sentido resulta especialmente extraño el tratamiento que se da a las “pérdidas” ocasionadas por la piratería: se mensura con claridad aquello que no perciben los titulares, pero no se discute qué individuos, firmas y países se apropian de esos recursos, y en qué medida lo hacen, con qué resultados, etc.

En este trabajo nos resulta importante sistematizar una organización que está parcialmente presente en la síntesis bibliográfica anterior. En trabajos previos (Zukerfeld, 2007: cap 3) hemos sugerido la conveniencia de analizar la circulación de bienes informacionales en base a dos variables: legalidad y mercantilización, con categorías dicotómicas. Las cuatro esferas que resultan de la combinación de estas variables, entendemos, son útiles para analizar la circulación y distribución de música, películas, textos y, lo que nos ocupa en este caso, software. En la tabla nro 1 encontramos una adaptación en ese sentido.

Tabla nro 1
Esferas de circulación de software

	Mercantil	No mercantil
Legal	1. Software mercantil (privativo) empaquetado o a medida, y algunas formas de SL/CA.	2. Software producido por fuera de la actividad laboral –SL/CA-, en la producción académica, ONG’s, y diversos niveles del Estado.
Ilegal	3. Software privativo copiado y comercializado con fines de lucro.	4. Software privativo copiado y compartido sin fines de lucro.

Fuente: Adaptación de Zukerfeld, 2007.

Pero en esta ponencia no nos interesa pensar cómo circula el software en todo el aparato productivo –lo cual sería interesante para futuras indagaciones-, sino específicamente la relación que tienen los procesos productivos de software con estas esferas. Esa relación tiene dos lados: el software como input y el software como output. En efecto, todos los procesos productores de software no sólo tienen al software como output,

sino que tienen la particularidad de que lo utilizan también como input. En lo que hace a los outputs, el cuadro no ofrece elementos relevantes. Los outputs de los procesos productivos que nos interesan *comienzan* a circular en las esferas 1 y 2. Sin embargo, los inputs no sólo provienen de esas dos esferas, sino, en algunos casos, también de 3 y 4. En qué medida ello ocurre y cómo se relaciona este consumo con el tratamiento dado a los outputs son cuestiones en las que intentaremos indagar en lo que sigue.

3. Los inputs de software

Respecto de la utilización de software como insumo de los procesos productivos, nuestro cuestionario indagó diferencialmente en sistemas operativos, lenguajes y aplicaciones o herramientas. Aunque una discusión detallada de esos tres niveles en cada proceso productivo excede a los alcances de esta ponencia, conviene asentar algunas cuestiones.

De manera general, aun sin distinguir entre procesos productivos, el sistema operativo más nombrado y utilizado por nuestros entrevistados es, a todas luces, Windows, en sus diferentes versiones. Es decir, un sistema operativo privativo. En menor medida, en varios procesos se utiliza también Unix. En cualquier caso, claramente la variabilidad en el uso de sistemas es mucho menor que en el uso de los lenguajes y las herramientas.

En el caso de los lenguajes de programación ya no es tan nítida la generalidad. Así, respecto de los primeros, si los más nombrados combinan opciones privativas y abiertas o libres: Java (SL/CA), PL/SQL (P), MySQL (SL/CA), PHP (SL/CA) y C++ (P), Visual Basic (P). Por último, en relación las herramientas o aplicaciones, constituyen el insumo más difícil de generalizar. La razón estriba en que aquí sí existe una diferencia muy marcada entre los procesos productivos de mayor tamaño y los más pequeños. Mientras que los primeros cuentan usualmente con herramientas propias, desarrolladas *in house* (sistemas de comunicaciones, facturaciones electrónicas, programas de auditorías internas, etc.); los segundos no solamente no cuentan con ellas; sino que, en ocasiones, tampoco las necesitan, utilizando en cambio herramientas básicas tales como el navegador y servicios de correo privativo de uso “no profesional”.

Luego de la descripción de las esferas apuntadas en el cuadro 1, preguntábamos a los entrevistados qué peso relativo tenía cada una de ellas dentro del proceso productivo en cuestión. Previsiblemente, las respuestas sugirieron que el tratamiento de los de inputs varía, en buena medida, en función del proceso productivo. Así, en las empresas multinacionales o grandes locales (tanto las que venden software como las que lo producen *in house*) sólo se usa software legal y, casi exclusivamente, privativo.

- Nada ilegal. Herramientas 100% licenciado. (GH, gerente empresa multinacional)
- Trabajamos todo tecnología Microsoft. Puede que haya algún producto de comunicaciones, o alguna cosa media específica, que no sea Microsoft, pero básicamente todo Microsoft. (...) Está el .Net es de Microsoft, básicamente estamos programando todo en .Net. Un poco de Visual Basic y base de datos SQL. (...) Software libre no tenemos nada. (...) Todo licenciado. Esta es una compañía que opera todo en blanco. Nada en negro. (PH, socio fundador, empresa grande local)
- Todo es legal (...) Nada ilegal. Son bastante delicados con lo que instalás. (BA, Desarrollador en empresa multinacional)
- Todo sw que probamos o usamos o está metido en algún lado tiene que ser totalmente legal. (ML, Administrador de servidores, empresa multinacional)

Esto se debe a distintas razones, pero la que más enfatizan los entrevistados es la de diversas formas de control (auditorías, multas) que se ejercen sobre los programas utilizados.

- Hay multas de unas cifras terribles si te llegan a encontrar alguna cosa rara. Es totalmente crítico que siempre tengan sw legal no existe la posibilidad de otra cosa. (ML, Administrador de servidores, empresa multinacional).
- Hay auditorías a nivel del software y te sale “tu máquina está compliance o no compliance. Tenés que arreglar esto, te falta tal parche, etc.”. Y cada tanto hay auditorías con la Security tool. No podés tener un Ares, etc. (MA, Administradora de bases de datos, empresa multinacional)

Sin embargo, pese a que el predominio del software privativo legal es abrumador, vale la pena insistir con lo que se sugirió más arriba: en algunos niveles —especialmente en las implementaciones de lenguajes— la presencia del software libre o de código abierto (SL/CA) no es desdeñable. Por ejemplo, en una empresa grande local todos los lenguajes mencionados son SL/CA (Java, Html, Bash, Python); para los sistemas operativos se usa, alternativamente, Windows y Ubuntu; las aplicaciones, finalmente, están asociadas a los lenguajes, por lo que

también son SL/CA (SQL Developer, bugzilla, Eclipse, Testlink). En síntesis, el entrevistado entiende que un 80% del software usado es SL/CS, un 15% privativo legal y un 5% privativo ilegal (entrevista con VR, team leader en empresa grande local).

Otra excepción al dominio del software privativo legal lo configura una empresa grande local altamente innovadora. Allí los empleados utilizan enormes cantidades de programas, tanto libres como privativos. Como nos cuenta CA, investigador de la empresa, se utilizan distintas implementaciones de lenguajes, sistemas operativos y herramientas, en función de las necesidades del proceso productivo. La empresa compra todas las licencias que los empleados solicitan. No obstante, el entrevistado nos indica que, aunque de modo marginal, en la firma se utiliza algo de software privativo ilegal. El grueso, sin embargo, es de software privativo legal y SL/CA, que son indispensables para los procesos productivos en curso.

Cuando pasamos a las empresas PYME, el cambio es notable. Por ejemplo, tomemos el caso de Tyresias, una empresa que en el momento de la entrevista contaba con una decena de empleados y una importante oficina en el microcentro de la ciudad de Buenos Aires. RA, socio gerente, explica sintéticamente el tratamiento que dan a los inputs:

- Nos movemos con: 50% privativo ilegal y 50% libre. O son libres y en otros casos son truchas. No pagamos.

En el caso de las microempresas, esta tendencia se acentúa. Por ejemplo, DJ es socio de una de estas pequeñas unidades productivas. A su vez, es activista del software libre. Sin embargo, resulta notable que su firma está lejos de utilizar sólo o predominantemente ese tipo de programas. Por ejemplo, entre los lenguajes utilizados además del SL PHP, recurre a C#, que utiliza sin licencia. Lo mismo sucede con los sistemas operativos, donde distribuciones de Linux y Windows (sin licencia) conviven, y con las aplicaciones (Photoshop, Visual studio, net beans, Ninja Manager). En síntesis, el entrevistado nos indica que recurren a dos esferas de software: 70% privativa ilegal y 30% SL/CA. Todo o casi todo el software que utilizan muchas de las restantes microempresas es privativo ilegal (Windows, Internet Explorer, Dream Weaver, Visual Basic, Visual Studio).

¿Cuáles son los motivos de adopción de estas proporciones de software? Simplificadamente, podemos enumerar razones, e intentar algún vínculo distinguiendo entre empresas grandes locales y multinacionales (GE-MN), de un lado, y micro, pequeñas y medianas (MPYME), de otro.

i) Los motivos vinculados a los *costos* varían según el tipo de empresa y de esfera de circulación. Las empresas grandes locales afrontan probabilidades altas de ser fiscalizadas y recibir multas, por lo que pagar las licencias del software privativo es conveniente, salvo cuando se trata de programas que se usarán de modo lateral y breve, para lo cual resulta más práctico tolerar la descarga ilegal, bajo responsabilidad del individuo que lo hace, usualmente con fines investigativos. El SL/CA en general no se adopta en estas firmas por motivos de costos, sino por los otros que se enumeran más abajo. No obstante, hay excepciones:

-¿Y vos cuál suponés que es el criterio?

No sé, obviamente lo libre es gratis, así que no están gastando guita. Pero no sé de qué manera lo deciden. O sea vos trabajás con las aplicaciones que tenés disponibles. Por ejemplo, si vos trabajás con uno libre y creés que otro pago es mejor, ¿lo podés ir a pedir? Sí sí. De ahí a que lo compren es otro tema. (BA, programador, empresa multinacional de software)

Para las MPYME no tiene sentido adoptar software privativo legal. No se trata tanto de que no puedan pagarlo, sino de que es un gasto innecesario: la probabilidad de ser detectados es bajísima y, dado que operan informalmente también en términos fiscales, prefieren en varios casos evitar toda inscripción como empresas productoras de software. La elección de SL/CA sí ocurre por temas de costos en varias de estas empresas. Parecería que en algunos procesos productivos más cercanos a la formalización –por su tamaño, por el tipo de clientes, etc.-hay una búsqueda de reemplazar, allí donde fuera posible, los software privativos ilegales por SL/CA con miras a los costos futuros.

ii) Los motivos *path dependence* son bastante homogéneos en las distintas clases de software. Incluyen, por un lado, las herencias de adopciones previas que generan altos costos para cambiar: “Muchas veces son legados que quedan... o alguien que les gustó.” (CA, Investigador empresa grande local). Estas adopciones pueden haber sido racionalizadas en algún momento o no. Pero el punto es que se escogen a los recursos humanos en función del trayecto que viene recorriendo la empresa. Esta modalidad es la típica de las GE-MN. En cambio, en las MPYME suele pesar más la trayectoria de los aprendizajes que portan los trabajadores-socios: “yo porque aprendí, me enseñaron ese (ASP Clásico), el que ahora es mi socio. Y después fui evolucionando en eso, y ya viene con las herramientas en eso, que no voy a agarrar Linux, que no tengo idea...no hubo una gran discusión sobre que lenguajes, sistemas o aplicaciones usar”(GRT, socio microempresa).

iii) En función del *producto/cliente*: En una empresa grande local, por ejemplo, nuestro entrevistado nos comentó que como sistemas operativos utilizaban tanto Windows como Linux. El primero para el producto empaquetado que realiza la firma y que funciona en ese sistema. El segundo, para nuevos desarrollos basados en software libre. Ambos sistemas están, en algunos casos, en “máquinas virtuales” por lo que los empleados los utilizan alternativamente. Pero las demandas de los clientes corporativos también inciden en las adopciones de las multinacionales:

- *¿Cuáles fueron los criterios de adopción de esos sistemas?* Es lo que estaba, el OS390 es el sistema operativo de mainframe. LA empresa que contrató el proyecto usa eso, lo tiene ese hace un montón y es como la arquitectura de procesamiento estándar de las empresas de servicios que maneja volúmenes gigantescos, como puede ser una telefónica.

En las MPYME es más usual (aunque no exclusivo) que sean las demandas de clientes corporativos las que determinen las elecciones de lenguajes y aplicaciones. Así, las esferas privativa ilegal y SL/CA se adoptan de acuerdo a los proyectos en curso. Por supuesto, esto genera a su vez una serie de aprendizajes *path dependence* que puede prolongar esta adopción cuando el vínculo con el cliente ha desaparecido.

iv) En las empresas grandes y multinacionales, especialmente las que prestan servicios a otras firmas, emergió como un elemento importante, por parte de las ofertas de software privativo, la existencia de servicios técnicos siempre disponibles para solucionar cualquier falla¹. Por ejemplo:

- *¿Cuáles fueron los criterios de selección de los sistemas?* Fundamentalmente por el soporte. Si vos estás trabajando con cualquiera de estas tecnologías abiertas, en definitiva es una engaña pichanga. Porque estos sistemas, el open, al final, si vos tenés necesidad de dar soporte a empresas y tenés un problema tanto con una base de datos o un lenguaje, si yo trabajo con redhat, al final tenés que terminar contratando a alguien si tenés un problema. En definitiva, tenés que pagar. Y si no tenés eso, no podés pretender dar servicios a empresas, porque la empresa requiere niveles de seriedad y de resolución que no pueden ser dilatados. No es php, unos pibes. Si yo tengo un problema, tengo la licencia, llamo por el teléfono y me lo resuelve la empresa. (PH, socio fundador, empresa grande local)

v) Complementariamente, la elección de SL/CA en varios casos está influenciada por la opinión de que se trata de una alternativa más segura –en términos de la prevención de virus, de control de los procesos críticos, etc.). Así, por ejemplo, nos comentaron que se decidió en una empresa grande local utilizar Free BSD en los servidores de la firma.

Tabla nro 2.
Motivos de adopción de inputs de software

Motivo de adopción	Privativo legal	Privativo Ilegal (con o sin fines de lucro)	SL/CA
Costos	GE-MN: Los costos de las probables multas exceden a los de las licencias).	GE-MN: El costo de transacción de adquirir legalmente productos menores es alto. MPYME: Pueden evitar las sanciones por el pago de licencias.	MPYMES: Firmas que consideran la formalización en el horizonte eligen por costos frente a la eventual esfera privativa legal.
Path dependence	GE-MN: Adopciones previas de la firma (racionalizadas o no) MPYME: Competencias de los socios-trabajadores		
En función del producto /cliente	GE-MN: En función del producto o del cliente	MPYME: En función del cliente	GE-MN: En función del producto o del cliente MPYME: En función del cliente

¹ Aunque entre las firmas relevadas no emergió, es importante señalar que las empresas que comercializan SL/CA también ofrecen servicios de asistencia técnica.

Soporte técnico	GE-MN: Actualizaciones y asistencia ante fallas	No	No
Seguridad	No	No	GE-MN y MPYME: menor exposición

Fuente: Elaboración propia en base a entrevistas.

4. Los outputs de software

En los procesos productivos relevados, los outputs de software son heterogéneos. Quizás sea útil tomar en cuenta el tipo de producto (entre los que hacen software empaquetados y a medida), de un lado, y por el tipo de regulación (ejercicios de derechos de autor, licencias “libres”, contratos específicos, sin licencia específica). Los procesos productivos relevados, por lo general, no ocupan un casillero específico, sino que combinan dos o más de ellos.

Tabla nro. 3
Tipos de output de software

	Servicios	A medida	Empaquetados
Derechos de autor	1.	2.	3.
Contratos específicos	4.	5.	6.
Sin regulación explícita	7.	8.	9.
SL/CA	10.	11.	12.

Fuente: Elaboración propia.

Los usuarios finales suelen asociar, implícitamente, al mercado del software con el casillero 3, esto es, con el software empaquetado cuya regulación se basa en las leyes de copyright y derechos de autor. El ejemplo de Windows es, en este sentido, paradigmático. Sin embargo, en países como la Argentina la producción de este tipo de software es escasa. Justamente por tratarse de programas que pueden ser copiados con costos cercanos a 0, producir localmente implica competir con productores internacionales, comúnmente con desventaja. Algo relativamente similar ocurre con otras formas de software empaquetado²: el SL/CA del casillero 12 –como alguna distribución de Linux- suele ser desarrollada a nivel internacional. Luego empresas como Red Hat pueden vender servicios complementarios (10). No obstante, por un lado, también hay adaptaciones locales y no mercantiles de Linux, como Ciberlinux.³

Retomando el software empaquetado con regulación basada en derechos de autor (3), aunque es minoritario, vale la pena detenerse en un ejemplo de este tipo –que posee interés propio- que ha surgido en nuestro trabajo de campo. Se trata de una empresa que tiene, como principal producto, un software para seguridad de grandes organizaciones (grandes empresas, estados). No se trata de un simple antivirus, sino de un programa que puede usarse incluso ofensivamente para hackear computadoras ajenas. Los clientes de esta firma son el Gobierno de los EE.UU. y numerosas firmas multinacionales. Es interesante resaltar que el precio del producto también muestra que no se trata de un software para consumidores finales: la versión completa cuesta USD 30.000 por año. Al igual que numerosos software empaquetados, el cálculo de su precio no depende de la cantidad de horas de trabajo que demanda y la copia no sólo se impide por vías legales, sino también a través de medios técnicos. Esto es, el propio software contiene “mecanismos muy poderosos” para impedir la reproducción ilegal. Intuitivamente, podría parecer que esta estrategia busca impedir la pérdida de beneficios oara la firma que implicaría la reproducción no autorizada. Sin embargo, nuestro entrevistado en la firma relativiza esa presunción:

² La producción que ocurre en los casilleros 6 y 9 es marginal y la obviamos aquí.

³ “Ciberlinux es una distribución Linux de origen argentino, derivada y “adaptada” de ubuntu, compartiendo paquete de software y archivos con esta última, creada y orientada para ser utilizada en cibercafés, administraciones públicas, empresas, etc” Ciberlinux, en Wikipedia.org.

-Por un lado, por una cuestión de imagen, porque se puede usar para romper todo, para hacer desastres. El otro es porque quieren cobrar, pero me parece que más importante es la primera. Es mantener el control de algo que es un arma.(CA, investigador empresa grande local)

Así, el mecanismo de apropiabilidad se combina con una estrategia de posicionamiento de la marca. El riesgo principal de la copia ilegal es la pérdida del prestigio de la marca, fundamental en el rubro de la seguridad informática. En otras palabras, lo que vende esta empresa es una ventaja relativa para grandes organizaciones respecto de otras organizaciones, por lo que restringir el acceso a éstas últimas parece ser un rasgo fundamental del negocio. De manera coherente con esto, la firma ofrece partes de su software de modo gratuito (12), a la vez que permite que los papers que surgen de las investigaciones que realizan sus trabajadores estén disponibles. En ambos casos busca tanto aumentar el prestigio como exhibir las debilidades de seguridad de los softwares que compiten con el propio. Naturalmente, la firma ofrece, complementariamente, servicios y programas a medida bajo distintas modalidades (1, 2, 4, 5).

Ahora bien, como señalamos, el grueso de los procesos productivos de software elaboran productos a medida y servicios. Por ejemplos, las PYME, y en buena medida la Micro, confeccionan sistemas de gestión y para la web. Por caso, 2/3 de la producción de la mencionada Tiresias son sistemas de gestión y el tercio restante, sistemas y aplicaciones para la web. La firma factura unos USD 120.000 anuales por estas actividades y los servicios aledaños. Por lo general, realiza contratos de cesión de derechos, por lo que, como muchas otras, habita la zona de los casilleros 4 y 5. Previsiblemente, cuando se pregunta a su socio gerente por la relevancia dada a las copias posteriores de los desarrollos de la firma, éste les quita toda importancia.

-Nosotros vemos cada cosa que hacemos como un proyecto, una vez que hicimos el proyecto, es muy específico de una necesidad puntual. Difícilmente se copie de manera completa. Se puede copiar partes y no nos interesa, la verdad. Nuestro trabajo lo cobramos y punto. Me parece además que el valor no está tanto en el producto, sino en el servicio, que es lo que nosotros hacemos.(RA, socio gerente PYME)

Las microempresas ofrecen un amplio abanico de opciones. Un caso interesante es el de Turistemas, que basa sus negocios en un único sistema de gestión web de paquetes turísticos. Lo ofrece a agencias de viajes. Esto es, lucra con la implementación del sistema propio (que podría verse como un empaquetado) y con los servicios de mantenimiento y actualizaciones que ofrece a sus clientes. Aunque sólo cuenta con cuatro socios y un empleado, factura unos USD 160.000 al año. El 85% de esa facturación se debe a un único cliente. Con este y con el resto de los demandantes de su producto, Turistemas firma un contrato que implica el pago de un abono (fee) mensual, que incluye actualizaciones y solución de diversos problemas.

-Le decimos tomá este sistema es tuyo y lo instalamos en tal lugar, vos me tenés que pagar para que te lo mantenga. Ellos no pueden ingresar al sistema y modificarlo. Hay un contrato que se firma sobre el sistema: de quien es, los datos de quien son, y que pasa si no se paga, y se acabó (GRT, socio microempresa)

De este modo, Turistemas parece moverse entre los casilleros 4,5 y 6. Esto la coloca en una situación intermedia respecto de los derechos de propiedad intelectual sobre su software.

- Yo copio, pero no se puede hacer copia de mi producto porque me enoja, pero eso es así. No, no es algo (la copia ilegal) de lo que nos preocupamos diariamente, porque en realidad el sistema es lo de menos. Para mí nosotros somos buenos en atender a los clientes. En los servicios, en darles ideas. De hecho yo también copio partes de otros sistemas. Copiar algo del código fuente, en nuestro ámbito, no veo complicaciones con eso.(GRT, socio microempresa)

Aquí, al menos en la representación del entrevistado, la falta de temor a la copia no surge tanto de la imposibilidad práctica, sino de los rasgos del nicho de mercado en el que funciona y del valor provisto por los servicios complementarios.

Algunas empresas grandes locales tienen modalidades parecidas, combinando la provisión de un sistema en particular con su implementación y servicios complementarios. Por ejemplo, Sistegas provee sistemas de gestión a través de la web hechos a medida, para comercialización de gas natural. Sus clientes son, naturalmente, empresas hidrocarburiíferas. La firma trabaja con la venta de licencias anuales y con contratos para la implementación y los servicios complementarios. Respecto de la preocupación por la copia de sus productos, el entrevistado, que coordina un equipo, entiende que no es ni central ni desdeñable:

-¿Qué tan importante es para quienes comandan el proceso productivo que se impida efectivamente la copia de programas protegidos bajo derechos de autor?" -Más o menos. -¿Por qué? -No sé, si lo pensaron o no. Sí a nivel contrato, lo habrán cubierto y demás, pero no hacemos algo tecnológico para evitar que pueda copiarse algo. -¿Qué tan importante es para vos que se impida efectivamente la copia de programas protegidos bajo derechos de autor? -No, a mi no me importa. -¿Por qué? -Todo lo que hacemos está hecho en software libre y no me parece mal que puedan tener acceso a lo que está hecho.

El fragmento muestra, además, que en este caso no se utilizan mecanismos técnicos para impedir la copia y, más importante, que el entrevistado entiende que el hecho de usar software libre como input sería contradictorio con impedir la difusión del producto obtenido (idea que, claro, imbuye el espíritu de las licencias GPL). En fin, la firma parece ocupar la zona de los casilleros 1,2,4 y 5.

5. Comentarios finales

Este ha sido un trabajo preliminar, exploratorio, incapaz de sugerir conclusiones. Van apenas, algunos comentarios para ordenar futuras versiones menos atropelladas que la presente. Partimos de una revisión bibliográfica de trabajos relativos a los mecanismos de apropiabilidad con el fin de estudiar los modos en que los distintos procesos productivos de software intentan aprovechar los conocimientos que producen. Tanto para entender el tratamiento que los procesos productivos dan a sus inputs como a sus outputs de software, nos servimos parcialmente de un esquema que distingue dos variables: legalidad y mercantilización, con categorías dicotómicas. Las cuatro esferas que resultan de la combinación de estas variables, entendemos, son útiles para analizar la circulación y distribución de música, películas, textos y, lo que nos ocupa en este caso, software. No pudimos estudiar, en esta ponencia introductoria, la diversidad de situaciones que presenta ese cuadro, y sólo nos enfocamos en las empresas que producen software con fines mercantiles. Con todo, considerar las dos variables, mercantilización y legalidad, es sugerente no tanto para analizar los outputs de estos procesos productivos como para dar cuentas de los inputs. Para ello, por comodidad, unimos las dos esferas ilegales, respecto de la no mercantil legal nos centramos en el SL/CA, y en cuanto a la mercantil legal nos quedamos sólo con el software privativo. Así utilizamos en el resto de esta ponencia un esquema de tres tipos de software.

En efecto, cuando analizamos los insumos de software, vimos que, simplificada, las firmas grandes locales y las multinacionales adoptan casi enteramente software legal. Mayormente, se trata de software privativo, aunque en algunos procesos productivos el software libre o de código abierto está presente. Las empresas micro y PYME, en cambio, combinan el software privativo ilegal –mayoritariamente–, con el SL/CA –en menor medida–. Parecería que en estos casos a mayor cercanía con la formalización del procesos productivo, mayor software libre y menor software privativo ilegal, aunque esta es una hipótesis que debería testearse y no una conclusión.

Respecto de los motivos para recurrir a las distintas esferas de circulación de estos inputs, encontramos los siguientes: i) Los motivos vinculados a los *costos* varían según el tipo de empresa y de esfera de circulación. Las empresas grandes locales afrontan probabilidades altas de ser fiscalizadas y recibir multas, por lo que pagar las licencias del software privativo es conveniente, salvo cuando se trata de programas que se usarán de modo lateral y breve, para lo cual resulta más práctico tolerar la descarga ilegal, bajo responsabilidad del individuo que lo hace, usualmente con fines investigativos.

Para las MPYME no tiene sentido adoptar software privativo legal. No se trata tanto de que no puedan pagarlo, sino de que es un gasto innecesario: la probabilidad de ser detectados es bajísima y, dado que operan informalmente también en términos fiscales, prefieren en varios casos evitar toda inscripción como empresas productoras de software. La elección de SL/CA sí ocurre por temas de costos en varias de estas empresas. Parecería que en algunos procesos productivos más cercanos a la formalización –por su tamaño, por el tipo de clientes, etc.– hay una búsqueda de reemplazar, allí donde fuera posible, los software privativos ilegales por SL/CA con miras a los costos futuros. ii) Los motivos *path dependence* son bastante homogéneos en las distintas clases de software. Incluyen, por un lado, las herencias de adopciones previas que generan altos costos para cambiar. Esta modalidad es la típica de las GE-MN. En cambio, en las MPYME suele pesar más la trayectoria de los aprendizajes que portan los trabajadores-socios. iii) En función del *producto/cliente*: En GE-MN la decisión parece depender del tipo de producto que se elabora y, en algunos casos, de las particularidades del cliente.

En las MPYME que hemos estudiado es más usual (aunque no exclusivo) que sean las demandas de clientes corporativos las que determinen las elecciones de lenguajes y aplicaciones. Así, las esferas privativa ilegal y SL/CA se adoptan de acuerdo a los proyectos en curso. Por supuesto, esto genera a su vez una serie de aprendizajes *path dependence* que puede prolongar esta adopción cuando el vínculo con el cliente ha desaparecido. iv) En las empresas grandes y multinacionales, especialmente las que prestan servicios a otras

firmas, emergió como un elemento importante, por parte de las ofertas de software privativo, la existencia de servicios técnicos siempre disponibles para solucionar cualquier falla. v) Complementariamente, la elección de SL/CA en varios casos está influenciada por la opinión de que se trata de una alternativa más segura –en términos de la prevención de virus, de control de los procesos críticos, etc.).

Respecto de los outputs de software, vimos como, tal como señala la literatura, la producción de software empaquetado está directamente relacionada con el uso de los derechos de autor como mecanismo de apropiabilidad. Así, como el grueso de la producción bajo estudio es de software a medida y de servicios informáticos, otros fueron los mecanismos privilegiados, dado que el temor a la reproducción impaga es generalmente bajo. Los contratos de prestación de servicios hoc son la forma preferida por las empresas estudiadas. No obstante, algunas licencian ciertos productos como SL/CA, y otras omiten todo licenciamiento específico.

Finalmente, en cuanto a la relación entre los inputs y los outputs parece haber bastante correspondencia entre dos modalidades prototípicas: por un lado, los usos de software legal y el pedido de cumplimiento de los derechos de autor. Por otro, el uso de software ilegal o libre y la despreocupación por las copias ilegales que pudieran hacerse.

Referencias bibliográficas:

- Anderson C. (2006). *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion.
- Anderson, C (2009) *Free: The Future of a Radical Price*. Hyperion,
- Arrow, K.J., (1962) Economic Welfare and the Allocation of Resources for Invention. En: Nelson, R. (Ed.), *The Rate and Direction of Inventive Activity*. Princeton University Press, Princeton, NJ, pp. 609–625.
- Banerjee, D. S. (2003), “Software Piracy: A Strategic Analysis and Policy Instruments”,
- Brandenburger, A y, Stuart, H., (1996). Value-based business strategy. *Journal of Economics and Management Strategy* 5, 5–24.
- Cohen, W., Nelson, R. y Walsh,(2000) ‘Protecting Their Intellectual Assets: Appropriability Conditions and Why US Manufacturing Firms Patent (or Not)’ *Working Paper N. 7552*, National Bureau of Economic Research, Cambridge, US, 2000.
- Conner, K R y Rumelt, R. (1991) "Software Piracy: An Analysis Of Protection Strategies," *Management Science*, informs, vol. 37(2), pages 125-139, february.
- Dasgupta, P., Stiglitz, J. (1980). Uncertainty, industrial structure, and the speed of R&D. *The Bell Journal of Economics* 11 (1), 1–28.
- Gopal, R.D. y Sanders G. Lawrence (1998). International software piracy: Analysis of key issues and impacts. *Information Systems Research*, 9, 4, 380-397.
- Hirshleifer, J., (1971). The private and social value of information and the reward to inventive activity. *American Economic Review* 61, 561–574.
- International Journal of Industrial Organization* 21, 97-121.
- Karakaya M y Ulutürk B (2010) Individual And Social Reasons Behind Software Piracy: An Analysis Of Previous Studies En *Uluslararası Güvenlik Ve Terörizm Dergisi • Cilt: 2 (2)*
- Klepper, S., (1996) Entry, exit, growth and innovation over the product life cycle. *American Economic Review* 86, 562–583.
- Lakhani K R. & Wolf, RG. (2005) “Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects” en J. Feller, B. Fitzgerald, S. Hissam, & K. R. Lakhani, *Perspectives on Free and Open Source Software MIT Press, Mass.*
- Levin, R.C., A.K. Klevorick, R.R. Nelson, S.G. Winter, R. Gilbert and Z. Griliches, (1987) ‘Appropriating the Returns from Industrial Research and Development’, *Brookings Papers on Economic Activity*, Vol. 1987, N. 3, Special Issue on Microeconomics, pp. 783-831.
- López, A (2009) *Innovation and Appropriability: Empirical Evidence and Research Agenda*. WIPO. January 2009. 1–40
- Peitz, M y Waelbroeck, P (2006) Piracy of Digital Products: A Critical Review of the Theoretical Literature (February 2006). Available at SSRN: <http://ssrn.com/abstract=889128> or <http://dx.doi.org/10.2139/ssrn.889128>
- Poddar, S., (2005), “Network Externality and Commercial Software Piracy”, *mimeo*, National University of Singapore.
- Schumpeter, J.A., (1950). *Capitalism Socialism and Democracy*. Harper and Row, New York.
- Shy, O. y Thisse, J. (1999), “A Strategic Approach to Software Protection”, *Journal of Economics and Management Strategy* 8, 163-190.
- Slive, J. and D. Bernhardt (1998), “Pirated for Profit”, *Canadian Journal of Economics* 31, 886-899.

11vo Simposio sobre la Sociedad de la Información, SSI 2013

- Takeyama, L. N. (1994), "The Welfare Implications of Unauthorized Reproduction of Intellectual Property in the Presence of Network Externalities", *Journal of Industrial Economics* 42, 155-166.
- Tapscott, D y Williams, A (2007) *Wikinomics La nueva economía de las multitudes inteligentes*, Paidós Empresa, Barcelona.
- Teece, D.J. (1986) Profiting from technological innovation. *Research Policy* 15 (6), 285–305.
- Varian, H y Shapiro, C (2000) *El dominio de la Información*. Antoni Bosch, Madrid.
- Zukerfeld, Mariano (2007) "La teoría de los Bienes Informacionales: Música y Músicos en el Capitalismo Informacional" en Perrone y Zukerfeld, *Disonancias del Capital*, Buenos Aires, Ediciones Cooperativas.
- Zukerfeld, M. (2010) *Capitalismo y Conocimiento: Materialismo Cognitivo, Propiedad Intelectual y Capitalismo Informacional*, Tesis doctoral, Volúmenes I, II y III. Disponibles en: capitalismoyconocimiento.wordpress.com