

# Cooperación y producción inmaterial en el software libre. Elementos para una lectura política del fenómeno GNU/Linux

Laurent Moineau y Aris Papatheódorou

Enero 2000

El jueves 9 de diciembre de 1999 la compañía VA Linux<sup>1</sup> entra en bolsa en el Nasdaq, el mercado público de las nuevas tecnologías. Las acciones VA Linux, que cotizan al salir 30\$ valen al final de la sesión 239,25\$. Una subida récord (698% en una sola sesión) para una entrada en bolsa en los EE.UU., pero sobre todo es la primera vez que en circunstancias similares unas acciones terminan la sesión por encima de la barrera de los 200\$. Más allá de la anécdota, estos hechos merecen una reflexión. En efecto, el entusiasmo repentino de los medios bursátiles por lo que no era, hasta poco antes, más que un sector marginal en la producción de software ¿es sólo un producto de la ideología americana de la *new economy*, la creencia en la reconquista del paraíso perdido gracias a las “nuevas tecnologías” de la comunicación y de lo inmaterial? ¿El capitalismo financiero está cediendo en esto a una simple moda pasajera con la esperanza de obtener algunos beneficios fáciles y rápidos? ¿O quizá es la señal de que el software libre –del que el sistema GNU/Linux es su locomotora– después de haber aumentado espectacularmente contra todo pronóstico el número de usuarios se impone ahora también al conjunto de la economía-mundo? Una de las paradojas de esta situación, y no la menos importante, reside precisamente en la emergencia en el seno mismo de la economía capitalista de un nuevo modelo productivo –el del “software libre”– que se construye por una parte en torno al rechazo de la apropiación privada de las fuentes de innovación y de la producción inmaterial y, sobre todo, por un ciclo de producción totalmente cooperativo.

Es la naturaleza y las modalidades de esta cooperación productiva, excepcional por muchos motivos, lo que nos interesa captar. En efecto, si no nos quedamos en la simple interpretación ideológica que no vería aquí más que una pura (y tradicional) contradicción entre una “alternativa no mercantil” y la potencia recuperadora del mercado, podremos observar en la emergencia del modelo productivo del software libre, en sus contradicciones y en los envites que se anudan alrededor de su devenir, no solamente elementos de interpretación del capitalismo posindustrial, sino sobre todo indicadores de los envites políticos que se dibujan bastante más allá del sector informático. Pero, para comprender lo que se está tejiendo ahora, nos parece que hace falta antes de nada recorrer una serie de procesos de constitución de saberes, de agregaciones comunitarias y de innovaciones tecno-científicas.

## 1 El nacimiento de Unix. Una auténtica revolución del software

La informática –nacida en la continuidad de la competición tecnológica de la II Guerra Mundial y desarrollada bajo el aguijón belicoso de la guerra fría– no está todavía, a principio de los años setenta, más que en una etapa quasi-experimental de una tecnología punta infrecuente y costosa. Los ordenadores son principalmente supercalculadoras imponentes, de uso frágil y a veces incierto pero sobre todo prácticamente incapaces de comunicarse entre sí. Un universo particularmente fragmentado en el límite del autismo, en el que los fabricantes ofrecían cada modelo de máquina

---

<sup>1</sup>VA Linux es una sociedad especializada en dar soluciones informáticas basadas en software libre a las empresas.  
<http://www.valinux.com>

con su propia sistema, su propio software, su propio lenguaje de programación pero también en el que los propios usuarios –en cuyos círculos restringidos a los programadores-pioneros se les empieza a conocer como *hackers*– preferían a menudo simplemente “producir” su propio sistema y sus propios lenguajes de programación adaptados a sus necesidades específicas. . .

En este contexto aparece un nuevo sistema operativo, Unix, que da radicalmente la vuelta al mundo de la informática.

Según una leyenda, el jefe de AT&T, cansado de los numerosos problemas de incompatibilidad entre los diferentes tratamiento de textos utilizados en aquel momento, habría abierto una importante línea de crédito para intentar resolver este quebradero de cabeza permanente. Ken Thompson –experto en materia de sistemas que procede de los laboratorios– y Dennis Ritchie –el inventor del lenguaje de programación C– se ponen manos a la obra. Rápidamente ponen en evidencia que el verdadero envite no está tanto en hacer compatibles los programas de tratamiento de textos con las diferentes arquitecturas de hardware, sino más bien en hacer que el sistema operativo mismo sea “portable” a los diferentes tipos de máquinas. Para ellos la evolución del hardware, que vive ya la aparición de los primeros miniordenadores, así como los auténticos progresos en materia de programación, hacen posible y necesaria la concepción de un sistema operativo completamente nuevo que contenga como cualidad intrínseca la posibilidad de la cooperación.

El proyecto Unix, iniciado en 1971, se impone en gran medida a partir de 1974, año en el que se aplica con éxito en numerosos modelos de ordenadores diferentes en los que funciona de manera casi idéntica. El sistema Unix se muestra por lo tanto como un agenciamiento de herramientas informáticas concebidas para combinarse eficazmente las unas con las otras y constituir así un “entorno informático común”. Se impone rápidamente por sus cualidades nuevas: una simplificación para el usuario de la estructura lógica de programación, una concepción modular del sistema que aumenta las posibilidades de adaptación (por ejemplo, en diferentes arquitecturas de hardware). Y sobre todo, Unix es el primer sistema operativo multiusuario y multitarea que se basa en el respeto de estándares abiertos.

## 2 El laboratorio de inteligencia de Berkeley

El interés de la comunidad universitaria por el sistema Unix cristaliza particularmente en la Universidad de Berkeley, California. En 1974 un grupo de investigadores y estudiantes de Informática adquiere el Unix de AT&T. La empresa les ha cedido por una cantidad razonable el código fuente del sistema <sup>2</sup>

Pero el sistema AT&T requiere para funcionar correctamente en las máquinas de Berkeley algunas mejoras y adaptaciones que aportarán en particular dos jóvenes estudiantes de Informática, Chuck Halley y Bill Joy, con la ayuda del mismo Ken Thompson. De este trabajo –estructurado en torno a a un pequeño equipo de pioneros– saldrá, a principios de 1977, la Berkeley Software Distribution (BSD), una versión del sistema Unix mejorada por los hackers de la Universidad de Berkeley. El sistema BSD, que conoce un verdadero éxito entre los equipos universitarios de todo el mundo, se pondrá al día regularmente durante los años siguientes gracias a las múltiples contribuciones y aportaciones de los mismos usuarios.

En seguida dos versiones Unix, pero sobre todo dos culturas Unix, corren paralelamente para luego enfrentarse. La de *Unix System Labs* de AT&T por un lado y la de Computers Systems Research Group (CSRG) de Berkeley por el otro; por un lado los mecanismos de innovación de empresa, por otro un proceso cooperativo de innovación en el seno de una comunidad (si bien restringida). El éxito de BSD plantea un problema en seguida a AT&T, que en 1992 lleva al CSRG

---

<sup>2</sup>Teniendo en cuenta la diversidad de hardware, era corriente en aquel momento que las empresas facilitaran el código fuente de los programas o del sistema (es decir, “el secreto de fabricación”), sin el que hubiera sido imposible adaptarlos o hacerlos funcionar. La universidad, que no posee más que rudos sistemas compartidos entre varios departamentos, está interesada en equipar a a cada uno de ellos con miniordenadores bajo Unix, más apropiados y de uso más flexible.

a los tribunales por haber sobrepasado los términos de la licencia acordada y haber hecho públicos “secretos industriales”. Será el comienzo de una larga batalla jurídica que no acabará hasta 1994, con la venta por parte de AT&T de los Unix System Blacks a la empresa Novell, que se apresura a llegar a un acuerdo. El proyecto BSD no sobrevive sin embargo a la aventura judicial. El acuerdo al que se llegó entonces permite de todos modos sacar una última versión del sistema BSD llamada “4.4 BSD Lite”, bajo licencia libre y desembarazada de toda huella de código perteneciente a AT&T.<sup>3</sup>

El final del desarrollo del Unix BSD, a causa de la epopeya judicial del copyright, pone fin a una época. La informática pasa de la era de los pioneros a la de la producción industrial de mercado. Numerosos hackers históricos abandonan los laboratorios para formar su propia empresa o se integran en los mayoristas del sector. La empresa Sun, por ejemplo, surge directamente del vivero de la Universidad de Berkeley donde fue concebida la versión BSD de Unix.

Esta situación tiene dos consecuencias principales. En primer lugar, la de conducir a una verdadera balcanización de los diferentes sistemas basados en Unix, hasta las diferentes tentativas de relanzar el proyecto BSD fuera del marco universitario (NetBSD, FreeBSD, OpenBSD). Cada una desarrolla sus propias variantes, sus propias implementaciones en detrimento la mayor parte de las veces de la compatibilidad. Y, en segundo lugar, encierra al desarrollo informático en lo que será llamado “sistema propietario”, es decir, la aplicación estricta por parte de las empresas del principio de propiedad privada a toda producción de software, excluyendo así toda publicidad del código y por lo tanto toda participación de los usuarios en su desarrollo. Lo que, en otros términos, lleva a impedir cualquier innovación en el seno de un producto terminado.

### 3 Una necesidad no satisfecha de innovación y de libertad

Anticipándose a ese callejón sin salida, en 1984 Richard M. Stallman lanza el movimiento del software libre (“Free Software Foundation”) con el objetivo político de oponerse a la apropiación privada de la inteligencia informática (el software) por parte de las empresas del sector. Pero ¿qué es eso del software libre? Podemos decir que un programa es libre si “tienes la libertad para ejecutarlo, sea cual sea el motivo por el que quieres hacerlo; tienes la libertad de modificar el programa para adaptarlo a tus necesidades (en la práctica, para que esta libertad tenga efecto, tienes que poder acceder al código fuente, ya que introducir modificaciones en un programa del que no se dispone del código fuente es un ejercicio extremadamente difícil); dispones de la libertad de redistribuir copias, ya sea gratuitamente o a cambio de una cantidad de dinero; tienes libertad para distribuir versiones modificadas del programa, de tal manera que la comunidad pueda beneficiarse de tus mejoras”.<sup>4</sup>

Con la constitución de la FSF y del proyecto GNU<sup>5</sup>, Stallman y otros hackers se empeñan en el objetivo de construir un sistema operativo de tipo Unix completamente basado en software libre... “Un sistema operativo permite hacer muchas cosas; sin sistema el ordenador ni siquiera puede funcionar. Un sistema operativo libre haría de nuevo posible una comunidad de hackers trabajando de modo cooperativo –e invitar a que se nos uniera quien quisiera. Todos y cada uno podrían utilizar un ordenador sin tener que sumarse a una conspiración que buscaba privar a tus propios amigos de los programas que utilizas.”<sup>6</sup> Una tarea descomunal que pasa primero por el desarrollo de programas libres (bajo etiqueta GNU) que correspondan a las funciones esenciales requeridas por un sistema operativo (compiladores, editores de texto, etc.). Pero también una

---

<sup>3</sup>Para un relato detallado de la aventura del Unix BSD, ver el artículo de Marshall McKusick, que se titula “Twenty Years of Berkeley Unix. From AT&T-Owned to Freely Redistributable”, en VV.AA. *Open Sources. Voices from the Open Source Revolution*, Editions O’Reilly, 1999. Hay disponible una versión íntegra en línea en el sitio del editor estadounidense <http://www.oreilly.com/catalog/opensources/book/toc.html>

<sup>4</sup>Richard Stallman, “The GNU Operating System and the Free Software Movement” en VV.AA. *Open Sources. Voices from the Open Source Revolution*. [Existe traducción castellana de este importante artículo en: <http://www.fsf.org/gnu/thegnuproject.es.html>]

<sup>5</sup>Creado según el principio del acrónimo recursivo, GNU significa “Gnu’s Not UNIX”, es un guiño de los tiempos en los que el copyright sobre el término *Unix* obligaba a los sistemas tipo Unix a denominarse *Unix like* o *Un\*x*.

<sup>6</sup>R. Stallman, obra citada.

formalización del estatuto del software libre de manera que lo que se desarrolle como código libre no pueda ser después transformado en “software propietario”, privando de este modo a los usuarios de toda libertad de uso. Se trata de la creación del principio de *copyleft* –por oposición al copyright– y sobre todo de su puesta en marcha bajo la forma de la licencia libre GPL. Esto plantea al mismo tiempo la libertad de uso y de modificación del código bajo GPL y la imposición de que todo programa que integre todo o una parte tenga que ser imperativamente a su vez software libre.<sup>7</sup>

## 4 Internet, la estructura que conecta

No es posible hablar de lo que ha ocurrido en el mundo de la informática y del software sin evocar el papel determinante de Internet. A partir de la experimentación de las primeras redes telemáticas, a fines de los años cincuenta bajo la égida de la investigación militar avanzada (ARPA), esta forma de comunicación seduce a los investigadores y a los informáticos por sus potencialidades en materia de intercambio de datos de manera horizontal y por la posibilidad de la cooperación a distancia.

Sin duda no es casualidad que el sistema Unix incluya de principio sus propios protocolos de red (UUCP e IP), que permiten a dos ordenadores funcionando bajo Unix intercambiarse mensajes electrónicos a través de líneas telefónicas ordinarias. Las máquinas Unix crearán muy pronto una red en la red, de la que saldrá en 1980 la red Usenet (la de los *newsgroups*, los forums electrónicos hoy parte integrante de Internet), que en seguida superará en amplitud a ARPAnet<sup>8</sup>. Este sistema de relaciones sociales a distancia<sup>9</sup>, a través de la escritura transportada de un ordenador a otro, no podía quedar confinada al mundo de la investigación. Los investigadores mismos son frecuentemente jóvenes y tienen inquietudes diversas, desde la política a la música, de la filosofía zen al deporte, gradualmente el uso de la red comienza a “desviarse” de su función primera que es la de conectar los superordenadores de los laboratorios de investigación y crean áreas de discusión dedicadas a los temas más diversos.

La red deviene un instrumento autónomo en manos de comunidades de investigadores. Un instrumento que funciona como una máquina de guerra para la circulación de procesos cognitivos pero también como máquina subjetiva para la constitución de espacios comunitarios virtuales. Así, lo que circula a través de las líneas telefónicas (correo, newsgroups) es más que una suma de simples datos académicos: la comunicación electrónica se convierte en el vector de agregación de microcomunidades de intereses, de cooperación en proyectos comunes. De hecho el sistema GNU/Linux llega en un momento en el que la cooperación se ha hecho posible a gran escala a través de la red. Podemos incluso decir que es de alguna manera la primera producción “palpable” de Internet, es decir, la primera vez que la red de redes materializa otra cosa diferente que ella misma.

## 5 El núcleo viene del frío

A principios de los años noventa, un joven estudiante finlandés, Linus Torvalds, sorprende proponiendo casi por azar un nuevo sistema operativo construido alrededor de un núcleo<sup>10</sup> muy pronto bautizado como “Linux”. Linus deseaba simplemente poder ejecutar en su casa, en su propio PC,

---

<sup>7</sup>La versión oficial de la GPL está disponible en [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html). [Existen varias traducciones castellanas no oficiales en [lucas.hispalinux.es/Otros/gples/gples.html](http://lucas.hispalinux.es/Otros/gples/gples.html)]

<sup>8</sup>ARPAnet es una red electrónica experimental puesta en marcha por la Advanced Research Project Agency (ARPA), dependencia del Departamento de Defensa estadounidense. En términos estrictos, de la fusión de ARPAnet entre otras con las redes universitarias y Usenet nacerá Internet.

<sup>9</sup>Sobre la naturaleza de las relaciones humanas en red, ver Jean-Louis Weissberg.

<sup>10</sup>El núcleo o *kernel* es prácticamente el corazón de un sistema de tipo Unix, el elemento alrededor del cual se agencian los módulos, los comandos, las bibliotecas y los programas para componer un sistema operativo en su conjunto. El término *Linux* designa de hecho el núcleo de un sistema llamado GNU/Linux.

los programas que él mismo realizaba o utilizaba en su carrera. Como su universidad estaba equipada con ordenadores de tipo Unix, Linus se propuso crear un sistema Unix libre de derechos que pudiera funcionar con una arquitectura hardware de tipo Intel (los PC), los microordenadores más corrientes.

De hecho, en ese momento, la tarea no era tan dura como parece. El proyecto GNU de la Free Software Foundation (FSF) se había ocupado ya de redesarrollar la mayor parte de los comandos Unix, así como cierto número de software fundamental como el compilador GCC. Un Unix libre, diferente del sistema BSD, ya había sido casi realizado, pero la FSF no había anticipado correctamente el desarrollo formidable del PC. Los programas de GNU eran instalados por administradores del sistema atentos al rendimiento y a la portabilidad sobre los miniordenadores que funcionaran con Unix. La idea genial de Linus fue centrarse en la concepción y realización de un núcleo capaz de interactuar con los programas de GNU, llevando el conjunto a los ordenadores personales.

Eric S. Raymond relata así su descubrimiento de GNU/Linux: “El encuentro con Linux fue un shock. A pesar de llevar bastantes años de actividad en la cultura hacker, todavía arrastraba la idea no contrastada de que los hackers amateurs, aun los más dotados, no podrían juntar jamás los recursos suficientes ni el talento para producir un sistema operativo multitarea utilizable[...]. Observar este exceso de código maravilloso desplegado ante mis ojos fue una experiencia bastante más potente que contentarme con saber, desde un punto de vista únicamente intelectual, que todas las trozos existían probablemente ya en alguna parte. Es como si me hubiera estado paseando durante años en medio de montones de piezas de recambio desaparejadas y me encuentro de repente enfrente de esas mismas piezas juntadas bajo la forma de un Ferrari rojo y rutilante, con las puertas abiertas, las llaves bailando en el contacto y el motor rugiendo promesas de potencia. . . Las tradiciones hackers que yo venía observando durante los últimos veinte años cobraban de repente vida de un modo nuevo y vibrante.”<sup>11</sup>

Concebido de manera casi confidencial en 1990, el sistema GNU/Linux se revela realmente en 1993. En nada de tiempo, se contagia lo que queda de las comunidades de hackers dispersas en el mundo y la joven comunidad de los nuevos usuarios de PC, que pasan así del bricolaje en MS-DOS a la programación en Linux. Algunos han dicho a propósito de Linux que era “el summum del hack”. Tras el énfasis de esta expresión, se constata que efectivamente el sistema construido en torno al núcleo linux potencia el conjunto del patrimonio del código acumulado durante dos decenios por toda la comunidad. Se revela como el sistema operativo más portable, puesto que funciona no solamente sobre arquitecturas Intel, sino también con PowerPC (Macintosh, IBM, Amiga), Sparc (Sun). GNU/Linux es también una plataforma opcional para sistemas empotrados: en pocos años, equipamientos tan diversos como autorradios, lavadoras o cafeteras llevarán este sistema incorporado.

## 6 El proceso de desarrollo del software libre

La verdadera innovación del sistema GNU/Linux no reside solo en su dimensión “tecnológica” (el núcleo portable), sino en los mecanismos sociales de producción de la innovación que se ponen en juego alrededor suyo. En efecto, una de las mayores fuerzas de este sistema operativo –que puede explicar ampliamente su éxito actual– es no solamente su fuerte contenido innovador, sino sobre todo haberlo basado en el potencial creativo existente en el software libre, y después en la utilización de la red Internet como espacio en el que se elaboran nuevos proyectos y en el que se pone en marcha una cooperación masiva y abierta. Retomando en gran medida los métodos de trabajo “comunitario” puestos en marcha durante un tiempo en torno a la rama BSD/Unix, o del Massachusetts Institute Technology (MIT), el desarrollo del código no será ya un acto aislado, ni como actividad personal ni en el marco empresarial, sino un verdadero proceso de cooperación en el que la relación producción/utilización tiende a diluirse. Los primeros usuarios del sistema son los

---

<sup>11</sup>Eric Raymond, “The revenge of the hackers”, en VV.AA. *Open Sources. Voices from the Open Source Revolution*.

mismos que lo desarrollan; del mismo modo podríamos decir que los primeros que lo desarrollan son sus propios usuarios.

El proceso de creación del software libre –contrariamente al “software propietario”– escapa a cualquier aproximación estilo marketing, basándose más bien en la noción de utilidad social. Alguien utiliza un programa y constata que no corresponde exactamente a sus necesidades, ya sea porque faltan algunas funcionalidades, ya sea porque su concepción no le satisface. Hace entonces el inventario de los programas existentes para comprobar si puede utilizar otro producto. Si este no es el caso, puede intentar la modificación del código fuente de un programa ya desarrollado o bien crear un software completamente nuevo, y en este último caso recurrir, vía listas de correo o grupos de news, a todos aquellos que expresen la misma necesidad para unir sus fuerzas y dar cuerpo a un nuevo software, por lo general retomando el código libre existente. Por tanto, la participación en un nuevo proyecto puede adoptar múltiples formas. Un usuario aguerrido puede aportar documentación pero también exponer una modificación del código –llamado *patch* o “parche”– al “mantenedor” del software.<sup>12</sup> Pero el usuario final también puede expresar sencillamente su opinión sobre tal o cual problema particular no anticipado por los desarrolladores o corregir un “bug” (fallo) menor. La capacidad de los autores de software libre para recurrir a la comunidad tanto para proponer mejoras como para reforzar el equipo de desarrollo es un factor determinante. Y de hecho, se constata que cuanto más abierto es un proyecto, más presiona la comunidad a los desarrolladores para que mejoren (o completen) su “producto” y para hacer frente a los riesgos de los inevitables abandonos de miembros del equipo.

En un texto que se ha hecho famoso, “La catedral y el bazar” (1998)<sup>13</sup>, Eric S. Raymond ha propuesto lo que denomina el “bazar” frente a la “catedral”, el modelo comercial tradicional de desarrollo. Raymond señala que, como las fuentes del código se pueden descargar libremente, la corrección de los bugs (o errores) se hace a un ritmo sostenido. Según Raymond, esto explica por qué los programas son generalmente fiables. En el ciclo de vida tradicional del software, la fase de pruebas representa frecuentemente más de un tercio del tiempo de trabajo consagrado a un producto. Al contrario, en la economía del software libre son los usuarios quienes se encargan de este tipo de tareas y quienes proponen las correcciones.

El tipo de organización descrita por Raymond se ha ido afinando desde entonces. Como la cooperación en red ha borrado las fronteras, los programadores de todos los países y de todos los medios están invitados a participar en los proyectos más diversos. Pasamos aquí a otra escala, lejos de los modelos de producción empresarial clásica. Conscientes de los riesgos de desviación del modelo “bazar” hacia un modelo “consejo municipal” (a fuerza de hablar, ya no hacemos nada)<sup>14</sup>, las comunidades han respondido de manera pragmática adoptando procedimientos técnicos que modifican las estructuras organizativas: el desarrollo se hace en red, con la ayuda de herramientas que automatizan las fastidiosas actualizaciones del código fuente. Surgen sofisticados sitios web dedicados al desarrollo en grupo y a la cartografía de bugs, siendo uno de los más interesantes el del proyecto Gnome (software de escritorio completamente libre). Tan seductor como devorador de tiempo ¿cómo podría resistir este modelo a los empujones de las grandes empresas informáticas, que desde 1998 invierten masivamente en el suministro de software para GNU/Linux? ¿Cómo por otra parte asegurar una remuneración correcta a los desarrolladores de software libre? En fin, ¿la comunidad que hace desarrollar este sistema ha tomado precauciones contra un control externo?

---

<sup>12</sup>Cada programa libre tiene una persona (a veces dos) que asume la tarea del “mantenimiento” del código y esto resulta ciertamente crucial. Tiene que añadir constantemente nuevas funcionalidades al software, al mismo tiempo que vela por su calidad y su estabilidad. Tiene que responder sobre todo a las peticiones de los usuarios, incorporando mejoras o modificaciones propuestas por estos últimos. Cuanto más importante es el proyecto, más evoluciona hacia funciones de coordinación o de arbitraje, siendo estos conflictos entre desarrolladores bastante frecuentes. Es por ejemplo la función actual del creador de Linux, Linus Torvalds, cuya autoridad en el desarrollo del núcleo es incontestable.

<sup>13</sup>Una traducción castellana se encuentra disponible en: <http://lucas.hispalinux.es/Otros/catedral-bazar/>

<sup>14</sup>Ver el texto de Alan Cox, uno de los principales hackers del núcleo Linux, “La catedral, el bazar y el consejo municipal”, publicado originalmente en <http://www.slashdot.org/features/98/10/13/1423253.shtml>

## 7 Business is business

El sistema GNU/Linux será de ahora en adelante un “blanco” privilegiado para los productores de programas. IBM, Oracle, Inprise (antes Borland), Sun (inventor del lenguaje Java), sin olvidar el precursor Netscape y más recientemente Matra proponen portar sus productos de referencia a GNU/Linux. Lo importante aquí es subrayar que estos productores han escogido estrategias diferentes e incluso opuestas para conquistar este mercado que, hay que recordar, no es para nada “cautivo”. La mayor parte de las versiones GNU/Linux de los programas que ofrecen estas empresas son gratuitos. Se puede descargar libremente el producto completo desde Internet o, como mínimo, una versión básica totalmente correcta<sup>15</sup>. La disponibilidad (liberación) del código fuente choca sin embargo con la concepción tradicional de los “decididores” de la economía del sector: los secretos de fabricación tienen que estar bien guardados cuando tan siquiera los beneficios surgen de la venta de programas (con la notable excepción de Microsoft), sino del suministro de servicios para las empresas. La presión de la comunidad de usuarios de programas es sin embargo bastante fuerte, por eso IBM ha anunciado que el código de su base de datos DB2 estará disponible gratuitamente.

La apuesta más audaz la ha llevado a cabo la empresa Netscape, persuadida por Eric Raymond. Para hacer frente a serias pérdidas de mercado, Netscape ha decidido liberar a la comunidad de desarrolladores el código de su software de navegación web, rebautizado al efecto como Mozilla. Netscape redistribuye su producto bajo una nueva licencia, la NPL (Netscape Public License), ya que según Eric Raymond las licencias GPL y BSD no podían aplicarse en este caso pues el código de Mozilla contiene partes desarrolladas por terceras empresas que todavía detentan copyright cerrado.

¿Cabría decir que actualmente, como parecen pensar los promotores del *open source*<sup>16</sup>, las estructuras y los modos de organización preconizados por la FSF constituyen un obstáculo para el desarrollo de Linux cuando limita la inversión de productores de software? No pensamos que esto sea así. Al contrario, existen muchas empresas que distribuyen ya desde hace tiempo sus productos bajo licencia GPL. El contrato está claro: la comunidad se encarga de coordinar el desarrollo del producto y la empresa le asegura que el software esté siempre disponible y sea adaptable. Este es el caso de la compañía AbiSoft, que desarrolla un procesador de textos llamado Abiword. También es el caso de la empresa Cygnus –recientemente adquirida por Red Hat– que desarrolla el compilador egcs, sucesor de GCC, adoptado entre otras por la NASA. El software libre ejerce muchas veces de vector de reconocimiento para la empresa que gana su dinero de otra manera (venta de CDs, manuales, asistencia, etc.). El modelo estadounidense de las *start-up* permite recaudar fondos para financiar desarrollos de software libre. La empresa parisina Mandrake Soft emplea así programadores cuya tarea consiste en participar en la realización de una *suite* de oficina bajo licencia GPL (KOffice). El desarrollo de este software no está bajo la responsabilidad de Mandrake Soft, que pone simplemente recursos humanos y materiales a disposición del proyecto. Aunque este tipo de cooperación no es fundamentalmente nuevo –el servidor gráfico de Unix, XWindow, ha sido desarrollado por un consorcio que reagrupaba a sus principales vendedores–, su originalidad reside en el hecho de que los proyectos asocian desarrolladores asalariados pero también altruistas. Estos últimos se ven ampliamente recompensados: además de la satisfacción personal y su valoración por

---

<sup>15</sup>El caso de la suite de oficina StarOffice es bastante emblemática: su uso se ha hecho totalmente gratuito después de la compra por parte de Sun de la empresa que lo había desarrollado... sin embargo sus fuentes siguen sin estar disponibles [N. de la T. En agosto de 2000 Sun publicó bajo licencia libre GPL las fuentes de esta suite ofimática multiplataforma, un clon del Office de Microsoft, rebautizada ahora como OpenOffice.]

<sup>16</sup>La Open Source Initiative (OSI) surgió como propuesta de algunos hackers –Eric Raymond entre ellos– para acabar con una ambigüedad (*free* en inglés, significa “libre” pero también “gratis”) y con un término que al parecer podía disuadir a las empresas, pero a cambio ha introducido otras ambigüedades tal vez peores: con el concepto *open source* (“fuente abierta”) que proponen como sustituto a “software libre” se pone solo el acento en que el código fuente esté disponible, sin incidir en las otras tres libertades (poder copiar, poder modificar y poder redistribuir libremente), por mucho que en su definición también lo incluyan. Pero lo peor es que ha servido sobre todo para crear confusión, pues algunas empresas usan la marca “open source” como estrategia de marketing para etiquetar software que no lo es. [N. de la T.]

parte de la comunidad, la participación en el desarrollo de software es un triunfo innegable que se menciona en un currículum. Añadamos para terminar que numerosos proyectos informáticos de final de carrera conciernen a sistemas GNU/Linux.

Bob Young, el patrón de Red Hat, estima en un 70% la proporción de desarrolladores GNU/Linux remunerados.<sup>17</sup> Aunque este porcentaje resulta un poco excesivo, conviene considerar que muchas personas que trabajan en la enseñanza superior y la investigación participan en el desarrollo o en la mejora de GNU/Linux en sus lugares de trabajo: el software libre se beneficia por lo tanto de una financiación indirecta muy importante.

## 8 El contrapoder de los usuarios

El devenir del sistema GNU/Linux suscita ya muchísimas interrogaciones e inquietudes, concretamente sobre el riesgo supuesto de una “recuperación por parte del mercado”. Pensamos que hay tres razones principales que demuestran de qué manera este sistema ofrece serias garantías de independencia.

La primera es de orden jurídico: el sistema se distribuye bajo licencia GPL. Eso significa, como ya hemos visto, que todo producto que contiene una parte, por mínima que sea, de código bajo GPL, ha de ser él mismo distribuido exclusivamente bajo licencia GPL. Como el corazón del sistema ha sido realizado pro cientos de programadores, esto significa que tendrían que ponerse todos de acuerdo para un cambio eventual del estatus jurídico. Aunque el reconocimiento legal de la GPL no sea evidente, sí bloquea cualquier tentativa de captación. Además, la marca “linux” –en sí misma un contrasentido– ha sido depositada por un grupo que defiende ferozmente la identidad del sistema: Linus Torvalds es por el momento su depositario. La segunda razón es de orden técnico: contrariamente a Windows o a MacOS (antes de la futura versión X), el sistema GNU/Linux –heredado de Unix– está construido en forma de capas sucesivas que se apoyan en normas y estándares abiertos. Tenemos el núcleo, la capa de red, los comandos desarrollados por GNU, la capa de base gráfica de la que se ocupa el consorcio XFree, los innumerables “gestores de ventanas” (de fvwm a enlightenment, pasando por WindowMaker y AfterStep), los programas de oficina (Gnome, KDE, etc.) y, para terminar, todas las aplicaciones. Un conjunto complejo del que evidentemente no es propietaria ninguna empresa ni ningún organismo.

La tercera razón es política. Los usuarios dependen de su sistema, la disponibilidad y la gratuidad del código son para ellos un auténtico empeño, a pesar de que la variable “política” sólo sale a relucir excepcionalmente. Para muchos de ellos, el software libre es una prolongación del espíritu de la contracultura de los últimas décadas.<sup>18</sup> Admitiendo que un depredador llegase a tomar el control de núcleo –aunque no se sabe cómo podría hacerlo– existen sistemas igualmente libres e igualmente basados en Unix, como la familia de los BSD herederos de la rama Berkeley (FreeBSD, OpenBSD y NetBSD) o como Hurd, el núcleo oficial de GNU, que también pueden hacer funcionar los programas Linux en las mismas condiciones. La comunidad abandonaría por lo tanto el barco con armas y equipaje y al cabo de un año todo el mundo habría olvidado Linux. . .

## 9 Para no concluir

El modelo de software libre trastoca la concepción que podíamos tener de la cooperación y de la innovación. Hay que constatar que nos encontramos por primera vez en presencia de una cooperación masiva, planetaria, altamente valorizable, asegurada conjuntamente por empresas, es cierto, pero

---

<sup>17</sup>Entrevista en *Langages & Systèmes*, noviembre 1999.

<sup>18</sup>El mismo Richard Stallman fue, a comienzos de los años setenta cuando era estudiante, uno de los animadores del *free speech movement* en el campus de la Universidad de Berkeley, como nos cuenta en una entrevista el magazine italiano *Decoder*.



sobre todo por un conjunto de sujetos autónomos, a la vez en el corazón del sistema capitalista pero también afuera, es decir, más allá.

Tradicionalmente, se venía diciendo con Schumpeter que el empresario era el motor de la innovación mientras que el marxismo ortodoxo ponía el acento en el “progreso social” como el fruto de las luchas de masas. El *operaismo* italiano efectúa un salto cualitativo sosteniendo que la lucha de clases es lo que determina el desarrollo capitalista: la innovación se plantea entonces a la vez como una apuesta de la lucha y como el producto de la relación de fuerzas entre el capital y la clase obrera.<sup>19</sup> Siguiendo este hilo, esto significa que la cooperación solo puede ser concebida como exclusivamente puesta en marcha en el interior de la esfera de la producción capitalista. El desarrollo del proceso cooperativo de producción de software libre alrededor del sistema GNU/Linux –con su fuerte contenido innovador–, a pesar de no encontrarse fuera de la dinámica de producción capitalista, pone en marcha fuerzas sociales que se determinan en gran medida fuera de la maquinaria económica. La circulación de saberes, la identificación colectiva con una ética de compartir el conocimiento, las prácticas de creación colectiva en red o los intentos de “moralización” de la relación mercantil, etc. sugieren que estamos en presencia de sujetos sociales híbridos, actores de un formidable despegue productivo, pero también actores de una verdadera movilización hacia la conquista de nuevos derechos.

Creemos, con Manuel Castells, que “el paradigma de las tecnologías de la información proporciona las bases materiales de su extensión al conjunto de la estructura social”<sup>20</sup>. Un modelo de cooperación social productiva ha surgido en torno al software libre: todavía no sabemos si podrá extenderse a otros sectores de la producción inmaterial. . . para constituir un nuevo paradigma productivo.

En torno a la necesaria lectura política de esta realidad que va cobrando forma, así como de su necesaria interpretación en términos que vaya más allá de las categorías de la economía política y de la ideología, nos parece necesario interpretar en lo sucesivo el fenómeno “GNU/Linux” o, para ser más precisos, el proceso de cooperación social ampliada que ha cristalizado alrededor de comunidades de desarrolladores y usuarios de este sistema informático libre.

Copyright ©2000 Laurent Moineau y Aris Papatheodorou  
Se permite la copia literal e íntegra de este artículo siempre y cuando esta nota se preserve.

*Traducción castellana:* Marisa Pérez Colina y Miquel Vidal, de sinDominio.net

---

<sup>19</sup>Raniero Panzieri, “Capitalisme et machinisme”, en *Luttes ouvrières et capitalisme d’aujourd’hui*, Colección “Cahiers Libres”, ediciones François Maspéro, 1968.

<sup>20</sup>Manuel Castells, *La société en réseau*, Fayard, 1998 [Traducción castellana: “La sociedad red”, Alianza, 1997]