

**Novática**, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática), organización que edita también la revista **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software). **Novática** edita asimismo **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (**UPGRADE European Network**).

<<http://www.ati.es/novatica/>>  
 <<http://www.ati.es/reicis/>>  
 <<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ**, **ASTIC**, **RITSI** e **Hispalinux**, junto a la que participa en **Prolinnova**.

**Consejo Editorial**

Antoni Carbonell Nogueras, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte, Francisco López Crespo, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molas i Bertrán, Oliba Palau Gordina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Grifó, Asunción Yturbe Herranz

**Coordinación Editorial**

Llorenç Pagés Casas <[lpages@ati.es](mailto:lpages@ati.es)>

**Composición y autoedición**

Jorge Llácer Gil de Ramalés

**Traducciones**

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lingua-informatica/>> Dpto. de Sistemas Informáticos - Escuela Superior Politécnica - Universidad Europea de Madrid

**Administración**

Tomas Brunete, María José Fernández, Enric Camarero, Felicidad López

**Secciones Técnicas - Coordinadores**

**Acceso y recuperación de la información**

José María Gómez Hidalgo (Optinet), <[jmgomez@yahoo.es](mailto:jmgomez@yahoo.es)>

Manuel J. María López (Universidad de Huelva), <[manuel.maria@diesta.uhu.es](mailto:manuel.maria@diesta.uhu.es)>

**Administración Pública electrónica**

Francisco López Crespo (MAE), <[flc@ati.es](mailto:flc@ati.es)>

**Arquitecturas**

Enrique F. Torres Moreno (Universidad de Zaragoza), <[enrique.torres@unizar.es](mailto:enrique.torres@unizar.es)>

Jordi Tubella Morgadas (DAC-UPC), <[jordit@ac.upc.es](mailto:jordit@ac.upc.es)>

**Auditoría STIC**

Marina Touriño Troilo, <[marinatourino@marinatourino.com](mailto:marinatourino@marinatourino.com)>

Manuel Palao García-Suelto (ASIA), <[manuel@palao.com](mailto:manuel@palao.com)>

**Borracho e tecnologías**

Isabel Hernández Collazos (Fac. Derecho de Donostia, UPV), <[ihernando@legalek.net](mailto:ihernando@legalek.net)>

Elena Davara Fernández de Marcos (Davara & Davara), <[edavara@davara.com](mailto:edavara@davara.com)>

**Economía Universitaria de la Informática**

Joaquín Ezpeleta Mateo (CPS-UZAR), <[ezpeleta@posta.unizar.es](mailto:ezpeleta@posta.unizar.es)>

Oriolada Parga Flores (DSEP-UJM), <[cparga@si.ucom.es](mailto:cparga@si.ucom.es)>

**Entorno digital personal**

Alonso Álvarez García (TID), <[aag@tid.es](mailto:aag@tid.es)>

Diego Gachet Páez (Universidad Europea de Madrid), <[gachet@uem.es](mailto:gachet@uem.es)>

**Estadísticas Web**

Encarnación Duesada Ruiz (Oficina Española del W3C) <[eduesada@w3.org](mailto:eduesada@w3.org)>

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería) <[jcarco@gmail.com](mailto:jcarco@gmail.com)>

**Geometría del Conocimiento**

José Baiget Solé (Carriem Ernst & Young), <[joan.baiget@ati.es](mailto:joan.baiget@ati.es)>

**Informática y Filosofía**

José Ángel Olivares Varela (Escuela Superior de Informática, UCLM) <[josangel.olivares@uclm.es](mailto:josangel.olivares@uclm.es)>

Karim Gherab Martin (Harvard University) <[kgherab@gmail.com](mailto:kgherab@gmail.com)>

**Informáticas Gráficas**

Miquel Chover Sellés (Universitat Jaume I de Castellón), <[chover@lsi.uji.es](mailto:chover@lsi.uji.es)>

Roberto Vivó Hernández (Eurographics, sección española), <[rvivo@dsic.upv.es](mailto:rvivo@dsic.upv.es)>

**Ingeniería del Software**

Javier Dolado Cosín (ISI-UPV), <[dolado@si.ehu.es](mailto:dolado@si.ehu.es)>

Luis Fernández Sanz (PRIS-EI-UEM), <[lufern@dpriis.esi.uem.es](mailto:lufern@dpriis.esi.uem.es)>

**Inteligencia Artificial**

Vicente Boti Navarro, Vicente Julián Inglada (DSIC-UPV) <[vbotti@vmpjades.com](mailto:vbotti@vmpjades.com)>

Información Persona-Computador

Julio Abascal González (FI-UPV), <[julio@si.ehu.es](mailto:julio@si.ehu.es)>

**Lenguaje e Informática**

M. del Carmen Ugarte García (IBM), <[cugarte@ati.es](mailto:cugarte@ati.es)>

**Lenguajes Informáticos**

Andrés Marín López (Univ. Carlos III), <[amarin@it.uc3m.es](mailto:amarin@it.uc3m.es)>

J. Anxo Velázquez Buriel (ESCT-URJC), <[a.velazquez@esct.urjc.es](mailto:a.velazquez@esct.urjc.es)>

**Lingüística computacional**

Xavier Gómez Guinovart (Univ. de Vigo), <[xgg@uvigo.es](mailto:xgg@uvigo.es)>

Manuel Patomar (Univ. de Alicante), <[mpatomar@dsi.ua.es](mailto:mpatomar@dsi.ua.es)>

**Mundo estudiantil y jóvenes profesionales**

Federico G. Mon Trotti (RITSI) <[gnu.fede@gmail.com](mailto:gnu.fede@gmail.com)>

Mikel Salazar Peña (Área de Jóvenes Profesionales, Junta de ATI Madrid), <[mikelxbo\\_uni@yahoo.es](mailto:mikelxbo_uni@yahoo.es)>

**Problemas Informáticos**

Rafael Fernández Castro (ATI), <[rftcalvo@ati.es](mailto:rftcalvo@ati.es)>

Miquel Sàrries Grifó (Ayto. de Barcelona), <[msarries@ati.es](mailto:msarries@ati.es)>

**Redes y servicios telemáticos**

José Luis Marzo Lázaro (Univ. de Girona), <[jesseluis.marzo@udg.es](mailto:jesseluis.marzo@udg.es)>

Germán Santos Booda (UPC), <[german@ac.upc.es](mailto:german@ac.upc.es)>

**Seguridad**

Javier Areltío Bertolin (Univ. de Deusto), <[jareltio@eside.deusto.es](mailto:jareltio@eside.deusto.es)>

Javier López Muñoz (ETS Informática-UMA), <[jlm@lcc.uma.es](mailto:jlm@lcc.uma.es)>

**Sistemas de Tiempo Real**

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <[alalmonso.puentej@dit.upm.es](mailto:alalmonso.puentej@dit.upm.es)>

**Software Libre**

Jesus M. González Barahona, Pedro de las Heras Quirós (GSYC-URJC), <[jmgh.pheras@gsyc.esct.urjc.es](mailto:jmgh.pheras@gsyc.esct.urjc.es)>

**Tecnología de Bibliotecas**

Jesus Garcia Molina (DS-UM), <[jmolina@um.es](mailto:jmolina@um.es)>

Gustavo Rossi (LIFIA-UNLP, Argentina), <[gustavo@sol.info.unlp.edu.ar](mailto:gustavo@sol.info.unlp.edu.ar)>

**Tecnologías para la Educación**

Juan Manuel Doderro Berrido (UCM), <[doderro@inf.uc3m.es](mailto:doderro@inf.uc3m.es)>

César Pablo Córcoles Briongo (UOC), <[ccorcoles@uoc.edu](mailto:ccorcoles@uoc.edu)>

**Tecnologías y Empresa**

Didac López Vilas (Universitat de Girona), <[didac.lopez@ati.es](mailto:didac.lopez@ati.es)>

Francisco Javier Gaitiás Sánchez (Indra Sistemas), <[jjcantaia@gmail.com](mailto:jjcantaia@gmail.com)>

**TIC y Turismo**

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga) <[aguayo.guevara@lcc.uma.es](mailto:aguayo.guevara@lcc.uma.es)>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción, sin ánimo de lucro, de todos los artículos, a menos que lo impida la modalidad de **©** o **copyright** elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

**Coordinación Editorial, Redacción Central y Redacción ATI Madrid**

Padilla 66, 3º, dcha., 28006 Madrid  
 Tlf: 914029391; fax: 913093685 <[novatica@ati.es](mailto:novatica@ati.es)>

**Composición, Edición y Redacción ATI Valencia**

Av. del Reino de Valencia 23, 46005 Valencia  
 Tlf./fax: 963330392 <[creceval@ati.es](mailto:creceval@ati.es)>

**Administración y Redacción ATI Cataluña**

Via Llobregat 45, ppal. T.: 08003 Barcelona  
 Tlf: 934125235; fax: 934127713 <[secregen@ati.es](mailto:secregen@ati.es)>

**Redacción ATI Andalucía**

Isaac Newton, s/n, Ed. Sadiel,  
 Isla Cartuja, 41092 Sevilla. Tlf./fax: 954460779 <[secreand@ati.es](mailto:secreand@ati.es)>

**Redacción ATI Aragón**

Lagascá 9, 3-B, 50006 Zaragoza  
 Tlf./fax: 976235181 <[secreara@ati.es](mailto:secreara@ati.es)>

**Redacción ATI Asturias-Cantabria**

<[gp-astucant@ati.es](mailto:gp-astucant@ati.es)>

**Redacción ATI Castilla-La Mancha**

<[gp-clmancha@ati.es](mailto:gp-clmancha@ati.es)>

**Subscripciones y Ventas** <<http://www.ati.es/novatica/interes.html>>, ATI Cataluña, ATI Madrid

**Publicidad**

Padilla 66, 3º, dcha., 28006 Madrid

Tlf: 914029391; fax: 913093685 <[novatica@ati.es](mailto:novatica@ati.es)>

Impresión: Dierra S.A., Juan de Austria 66, 08005 Barcelona

Depósito legal: B 15.154-1975 - ISSN: 0211-2124. CODEN NOVAEC

Partida: "Salida de la habitación 101" - Concha Arias Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

**editorial**

**Estudiantes y jóvenes profesionales, clave del futuro de ATI** > 02  
**en resumen**

**El poder de laas comunidades** > 02  
*Llorenç Pagés Casas*

**monografía**

**Sortware libre: investigación y desarrollo**

(En colaboración con **UPGRADE**)

Editores invitados: *Manuel Palomo Duarte, José Rafael Rodríguez Galván, Israel Herraiz Tabernero y Andrea Capiluppi*

**Presentación. Software libre: investigación y desarrollo** > 03

*Andrea Capiluppi, José Rafael Rodríguez Galván, Manuel Palomo Duarte, Israel Herraiz Tabernero*

**La necesidad de investigar sobre software libre en Europa** > 06

*Israel Herraiz Tabernero, Rafael Rodríguez Galván, Manuel Palomo Duarte*

**De la catedral al bazar: un estudio empírico del ciclo de vida de los proyectos basados en comunidades de voluntarios** > 09

*Andrea Capiluppi, Martin Michlmayr*

**Los bienes comunes como nueva economía y lo que esto significa para la investigación** > 17

*Richard P. Gabriel*

**Software libre para la gestión de proyectos de investigación** > 20

*Israel Herraiz Tabernero, Juan José Amor Iglesias, Álvaro del Castillo San Félix*

**Innovación tecnológica en comunicaciones móviles desarrollada con Software Libre: Campus Ubicuo** > 25

*Javier Carmona Murillo, José Luis González Sánchez, Manuel Castro Ruiz*

**El modelo de la Oficina de Software Libre de la Universidad de Cádiz en la universidad española** > 31

*José Rafael Rodríguez Galván, Manuel Palomo Duarte, Juan Carlos González Cerezo,*

*Gerardo Aburruga García, Antonio García Domínguez, Alejandro Álvarez Ayllón*

**Aprendiendo a introducir una innovación en un proyecto basado en Software Libre** > 36

*Christopher Oezbek, Lutz Prechelt*

**Optimización del proceso de render 3D distribuido con software libre** > 41

*Carlos González Morcillo, Gerhard Weiss, David Vallejo Fernández,*

*Luis Jiménez Linares, Javier Albusac Jiménez*

**secciones técnicas**

**Mundo estudiantil y jóvenes profesionales**

**SWAML, Semantic Web Archive of Mailing Lists** > 49

*Sergio Fernández López, Diego Berrueta Muñoz, José Emilio Labra Gayo*

**TCOS: uso de terminales ligeros en las aulas** > 52

*Mario Izquierdo Rodríguez*

**Porting de GCC al microcontrolador Microchip PIC16F877** > 55

*Pedro José Ramírez Gutiérrez*

**SubDownloader** > 58

*Iván García Cortijo*

**Software Libre en la Enseñanza: primeras jornadas organizadas por OuSLi en el ámbito de la educación** > 61

*José Ramón Méndez Reboredo, Enrique Estévez Fernández, Florentino Fernández Riverola,*

*Daniel González Peña*

**Referencias autorizadas** > 64

**sociedad de la información**

**Nueva Economía**

**Las TIC y la Ciencia, Ingeniería y Gestión de los Servicios** > 69

*Gregorio Martín Quetglas, Vicente Cerverón Lleó, Francisco J. Gálvez Ramírez*

**Programar es crear**

**Todas las palabras son capicúas (CUPCAM 2006, problema F, solución)** > 73

*Oscar Martín Sánchez*

**Las luces de la escalera (CUPCAM 2006, problema G, enunciado)** > 74

*Julio Mariño Carballo*

**Permutaciones con un número dado de inversiones (CUPCAM 2006, problema H, enunciado)** > 75

*Manuel Abellanas Oar, Luis Hernández Yáñez*

**asuntos interiores**

**Coordinación Editorial / Programación de Novática** > 76

**Normas para autores / Socios Institucionales** > 77

**Monografía del próximo número: "Gobierno de las TIC"**

Richard P. Gabriel  
IBM Research

<rp@{dreamsongs.com | us.ibm.com}>

# Los bienes comunes como nueva economía y lo que esto significa para la investigación

## 1. Introducción

A veces aparecen novedades a una velocidad que ni investigadores ni profesionales son capaces, o están dispuestos, a observar. Un ejemplo ocurrido recientemente ha sido la *aparición de la aparición*<sup>2</sup> como un campo de estudio, en forma de ciencia de la complejidad. Durante siglos, fenómenos que actualmente se consideran como el centro de muchas disciplinas científicas no fueron observados o ni siquiera se consideraron algo serio.

Los investigadores y los profesionales del software libre están enamorados de las licencias, de las herramientas y su uso, del desarrollo comunitario, y de la eficacia y eficiencia con la que la metodología del software libre está produciendo software. Sin embargo, algo mucho mayor podría estar cambiando el panorama de la computación, y no sólo añadiendo algunos conocimientos a la ingeniería del software.

Durante los últimos 10 años, las empresas han estado contribuyendo con una considerable cantidad de software al mundo del software libre. Por ejemplo, Sun Microsystems ha calculado que, usando medios convencionales para la asignación de valor monetario al código, ha contribuido con más de mil millones de dólares en código. IBM y, posiblemente, otras grandes compañías no están muy atrás. Especialmente interesante es que Sun ha tomado la decisión de abrir todo el código de su software, y parece que sigue el camino correcto. Al mismo tiempo, no está depositando todas sus expectativas de beneficio en su hardware: también esperan ganar dinero con su software.

<sup>1</sup> **Nota del Editor:** la mejor traducción del término inglés *commons*, un término cada vez más en boga, de la mano del conocimiento abierto y del software libre, se halla todavía en discusión. La traducción más habitual es "bienes comunes", aunque hay quien opina que bastaría con traducirlo por "comunes", dada la primera acepción del término "común" en el diccionario de la Real Academia: *Dicho de una cosa: Que, no siendo privativamente de nadie, pertenece o se extiende a varios. Bienes, pastos comunes.*

<sup>2</sup> **Nota del Traductor:** "emergence of emergence", se trata de un juego de palabras.

**Traducción:** Alejandro Álvarez Ayllón (Oficina de Software Libre, Universidad de Cádiz).

©IEEE Este artículo fue publicado previamente en las actas del *First International Workshop on Emerging Trends in FLOSS Research and Development*, 2007. FLOSS '07. ISBN: 0-7695-2961-5. Digital Object Identifier: 10.1109/FLOSS.2007.14. Se publica con los correspondientes permisos de los autores y de IEEE.

**Resumen:** *suponga que toda la estructura social y comercial que está detrás de la creación de software esté cambiando, refiriéndonos a que se esté convirtiendo en bienes comunes<sup>1</sup>, haciendo caer drásticamente sus costes. ¿Como podría el mundo cambiar y cómo podríamos reconocer esos cambios? El software no sería continuamente recreado por diferentes organizaciones, por lo que la "eficiencia" global de la producción de software se incrementaría espectacularmente; por lo tanto sería posible crear valor sin despilfarrar medios, la experimentación y el riesgo serían asumibles (y probablemente necesarios debido a que las compañías no podrían cobrar por duplicar su infraestructura), y el tamaño y complejidad de los sistemas existentes se incrementaría, quizá más allá de la comprensión humana. Tanto o más importante sería que las actividades de creación de software tendrían su origen en personas, organizaciones y disciplinas que, hoy en día, se consideran consumidores. Habría, en un sentido literal, un único sistema software existente, continuamente creciente y sería ésta una ecología gestionada por economistas, sociólogos, gobiernos, clubes, comunidades y multitud de disciplinas. Aparecerían nuevos modelos de negocio, quizá a una velocidad alarmante. ¿Cómo deberíamos diseñar nuestra investigación de forma que podamos observar y comprender estos cambios? Existen algunas evidencias de que el cambio está en proceso, como resultado de la adopción de software libre por parte de compañías que no son simplemente receptores agradecidos de los "regalos" procedentes de los evangelizadores del software libre, sino que son inteligentes ladrones que reenfocan las ideas y crean algunas nuevas de su propia cosecha.*

**Palabras clave:** *bienes comunes, código fuente, estrategia de negocio, licencia de software, nuevos modelos de negocio, producción de software, propiedad intelectual, software libre.*

## Autor

**Richard P. Gabriel** se doctoró en Informática por la Universidad de Stanford en 1981 y recibió un Master of Fine Arts en Poesía por el Warren Wilson College en 1998. Es un Ingeniero Distinguido en IBM Research, especializado en la arquitectura, diseño e implementación de sistemas ultragrandes y autosustentados, así como en las técnicas de desarrollo para construirlos. Fue Presidente y es actualmente miembro del Consejo de Dirección de Hillside Group, una organización sin ánimo de lucro que apoya a la comunidad de patrones de software mediante conferencias, publicación de libros y concesión de becas. Es autor de 4 libros y un libro de cordel de poesía. Ha ganado varios premios, incluyendo el AAAI/ACM Allen Newell. Es guitarrista líder en una banda de *rock and roll* y poeta.

## 2. Sun: Un caso de estudio (Breve visión general)

Sun comenzó en 1982 como una compañía basada en estándares abiertos y productos de consumo: BSD Unix, procesadores Motorola 68000, y TCP/IP. A finales de los 90 comenzó a experimentar con ideas relacionadas con el código abierto, y con verdadero código abierto: Jini (no es auténtico código abierto, pero es un experimento interesante sobre los conceptos y prácticas de código abierto, combinados con estrategias de mercado), Netbeans, Juxta, y OpenOffice fueron los primeros pasos, seguidos por Glassfish, Grid Engine, OpenSparc, OpenSolaris, Open Media Commons, y, más recientemente, Java. Si a esto añadi-

mos Java.net se abren interesantes posibilidades. Sun está claramente experimentando con el concepto global de bienes comunes. OpenSparc es un diseño hardware que fue licenciado bajo una licencia libre, con el objetivo de crear mercado; Open Media Commons es ante todo un proyecto abierto de DRM, pero también enfoca la pregunta de qué significa la propiedad intelectual en el siglo XXI. Java.net es una especie de meta-comunidad dirigida a la creación de mercados alrededor de Java. Se considera a Solaris y Java las joyas de la corona de Sun.

Durante esta etapa experimental en Sun (que todavía continúa) se puso especial énfasis en los modelos de negocio y de gobierno.

Sun está impulsando cuatro estrategias de negocio relacionadas con el software libre:

- Incrementar el volumen comprometiendo a los desarrolladores de software y reduciendo los obstáculos a su adopción.
- Compartir el desarrollo con desarrolladores externos y con proyectos libres ya establecidos que sean requeridos por los sistemas de Sun.
- Dirigirse a mercados emergentes cuyos gobiernos o intereses demanden software libre, como Brasil, partes de la Unión Europea, Rusia, India y China
- Irrumpir en mercados cerrados suministrando alternativas de software libre.

Sun ha realizado un interesante conjunto de observaciones sobre los cambios ocurridos desde que se monetizó el software. En los años 70, el software era fundamentalmente parte de un paquete hardware. Se compraban sistemas completos: hardware y software. En muchos casos, las compañías que vendían hardware suministraban el código de las aplicaciones a sus clientes para que lo adaptaran a sus necesidades, y no se consideraba extraño.

Entre las dos décadas entre los años 1980 y 2000, las compañías productoras de hardware comenzaron a desvincular su software, y empresas dedicadas a éste empezaron a vender software que realizara todo tipo de cosas, incluyendo sistemas operativos. Lo que estos dos periodos tienen en común es que el software era pagado cuando era adquirido. Y parecía no haber otra opción: si querías usar algo, primero tenías que comprarlo.

Con el software libre y la búsqueda de los modelos de negocio apropiados, esto puede cambiar, y ese cambio comenzó a principios de esta década de los 2000. El uso de software libre suele ser, por lo general, gratuito. No obstante, hay empresas y, a veces, individuos dispuestos o ansiosos por pagar a cambio de: soporte y mantenimiento, actualizaciones periódicas y solución de errores, indemnizaciones por responsabilidad y protección de patentes. En estos casos, la monetización puede ocurrir cuando se implanta el producto final. Esto implica que, en esos casos no cuesta nada explorar una idea para un producto concreto hasta que ésta sea completamente integrada para su venta o distribución. Entonces, si el productor lo desea, podrían comprarse uno o varios de estos servicios.

Aplazando algunos de los costes derivados de la aparición de nuevos productos y, posiblemente, nuevas empresas, pueden explorarse muchas más ideas considerando todo el mercado. Las barreras para la experimentación son muy bajas.

El repertorio completo de modelos de negocio que Sun ha identificado son:

- Suscripción (como ya hemos mencionado), incluyendo indemnizaciones y protección de patentes, extendiendo un paraguas de seguridad en lo referente a la propiedad intelectual que cubre a las partes que suscriben los servicios.
- Licencia dual, vendiendo las nuevas versiones del código, mientras que las antiguas son libres.
- Administración o custodia, esto es emplear un estándar para atraer a desarrolladores que lo empleen, y venderles otros productos y servicios a ellos mismos.
- Embebido, donde el código es parte de otra cosa, normalmente hardware, que es comercializada.
- Consultoría, donde se vende la experiencia de una persona o empresa en un determinado código, típicamente servicios de liderazgo de grupos de programación.
- Alojamiento, donde los servicios proporcionados por el software libre son ejecutados en servidores, vendiéndose el acceso a dichos servicios, o empleando otras maneras de obtener beneficio (como anuncios).
- Entrenamiento y formación sobre el código o sobre las metodologías de software libre.

Los teóricos del software libre de Sun ven en estas observaciones un ciclo virtuoso donde, encontrando un lugar para añadir valor en el código de estos bienes comunes, una compañía (o persona) puede crear un punto de monetización sin tener que invertir por sí sola en el código, produciendo de este modo un producto o servicio a un coste menor.

### 3. Lo que esto significa para el software

Suponga que Sun no está en una situación aislada, y que las compañías y otro tipo de organizaciones (incluyendo los individuos) se están preparando para cambiar sus modelos de negocio y de desarrollo de software basándose en el ciclo virtuoso descrito por Sun. ¿Cómo podrían las iniciativas completas de producción de software cambiar, y qué significaría esto para la ingeniería del software?

Analícemos el escenario. La gran mayoría del software sería un bien común, y estaría disponible para ser usado. No quedaría mucho de propietario. Habría presiones por parte de los clientes para que se realizaran unificaciones o simplificaciones. Por ejemplo, ¿por qué habría necesidad de tener múltiples sistemas operativos si no es por los diferentes requerimientos de escala, tiempo real, y distribución? Por otro lado, encontrar nuevo valor podría provocar que las empresas crearan bifurcaciones a partir del código base con el objetivo de crear plataformas o puntos de partida para categorías enteras de nuevas fuentes de valor. ¿Cómo podría romperse este equilibrio?

Como las barreras para emprender casi cualquier empresa serían tan bajas, habría muchos jugadores (incluyendo pequeñas empresas e individuos) capaces de ser un factor a tener en cuenta en cualquier área de negocio. Con más participantes, habría más oportunidades para que surgieran nuevas ideas e innovaciones. ¿Cómo irrumpirán estos en el mercado? ¿Intentarán quizás las empresas convertirse en repositorios de propiedad intelectual para ofrecer las mejores indemnizaciones? ¿Lograrán ser otras entidades, como universidades privadas o laboratorios de investigación, jugadores significados debido a que pueden ofrecer un potente portafolio de patentes que pueden usar para proteger a sus clientes? Teniendo en cuenta los enormes portafolios existentes, como los pertenecientes a IBM o Microsoft, parecería que éstos continuarían dominando; sin embargo, en nuevas áreas o nichos de mercado las pequeñas organizaciones, incluso individuos, podrían mantener las patentes clave.

Inmediatamente surgen algunas consideraciones obvias. ¿Qué pasa con las licencias? Actualmente, los grandes sistemas se obtienen a través de la unión de "subsistemas" (por expresarlo de alguna forma) publicados bajo distintas licencias. Lo que no se permite es mezclar piezas licenciadas con diferentes bases fuente. ¿Existirá una presión por poner todo el código bajo la misma licencia o la presión irá en otro sentido, es decir en crear nuevas licencias especializadas?

### 4. Lo que esto significa para la Ingeniería del Software

Como pocas compañías "poseerían" un sistema completo o aplicación sectorial, podría haber alguna presión en las bases de código para distanciar las APIs, protocolos, formatos de datos, etc. Y si esto ocurriera, ¿de dónde vendría la presión opuesta? ¿Se encargarían las organizaciones de estandarización o se crearían estructuras de gobierno como la Fundación Apache o la IETF? ¿O surgirían empresas para definir las aplicaciones o estructuras de los sistemas tal como pasó con el ordenador personal a principios de los 80? En ese caso, IBM estableció una serie de normas de diseño determinando cuáles eran los componentes de un PC y cómo interactuaban [1].

Esto permitió al mercado formarse alrededor de los diversos componentes, lo que cambió la naturaleza del diseño de los sistemas informáticos. Hoy en día, esta forma de ver el diseño ha creado una nueva aproximación a los problemas de la ingeniería del software: *la ingeniería del software guiada por la economía*.

La enseñanza de la informática y del software cambiaría, ya que todo el código estaría disponible para su estudio (e, incluso, la

mejora formaría parte del proceso). En este sentido, los desarrolladores estarían mejor formados que nunca.

La programación dejaría de ser materia de inteligencia e invención para ser más un proceso de encontrar un código fuente existente que se aproxime y, o bien adaptarlo o adaptarse a él. Las licencias podrían ayudar o entorpecer. Con menos presión para crearlo todo a partir de cero, sería posible crear sistemas cada vez más grandes con un tamaño de equipo asequible. Esto afloraría las cuestiones y problemas asociados a los sistemas de muy gran escala (ULS, *Ultra Large Scale*). Citando una convocatoria de artículos para un taller sobre este tema [2][3]:

*En pocas palabras, un incremento radical en la escala y complejidad demandará nuevas tecnologías y enfoques para todos los aspectos relacionados con la concepción, definición, desarrollo, implementación, uso, mantenimiento, evaluación, y regulación del sistema. Si los sistemas software en los que nos centramos hoy en día son comparados a edificios o a infraestructuras individuales, entonces los sistemas ULS son más parecidos a ciudades o redes de ciudades. Y, como éstas, tendrán nodos individuales complejos (similares a edificios o infraestructuras), así que deberemos continuar mejorando las tecnologías y métodos tradicionales; pero también mostrarán organización, y requerirán tecnologías y visiones fundamentalmente distintas a aquellas que son adecuadas a nivel de nodo. Los elementos software de los sistemas ULS presentan unos retos especialmente abrumadores. Desarrollar las tecnologías y enfoques necesarios a su vez requerirá investigaciones básicas y aplicadas diferentes a las que hemos realizado en el pasado. Facilitar el desarrollo de sistemas ULS (y, particularmente, sus elementos software) requerirá nuevas ideas en muchas disciplinas, incluyendo las ciencias de la computación y la ingeniería del software, pero también en otras como economía, urbanismo y antropología.*

El cambio de software propietario a software basado en bienes comunes aceleraría la época de los sistemas ULS, lo que supondría un cambio cualitativo debido a su escala masiva. Si esto ocurre, se pondría de relieve la insuficiencia de nuestras herramientas, incluyendo metodologías y lenguajes de programación.

## 5. Lo que esto significa para la investigación

El hábito de la investigación en computación es profundizar mucho en las cuestiones planteadas. En cierto sentido, a los investigadores les encantan los rompecabezas. Gregory Tverton escribió esto acerca de los rompecabezas y los misterios en un artículo para y sobre los servicios de inteligencia [4].

*Actualmente, los servicios de inteligencia se hallan en el negocio de la información, no solamente en el ámbito de los secretos, lo que supone un enorme cambio para la profesión. En las circunstancias de la era de la información, es hora para la comunidad de la inteligencia de distinguir entre rompecabezas y misterios. Los rompecabezas tienen soluciones concretas, sólo si tenemos acceso a la información (secreta) necesaria. Fueron la especialidad de los servicios de inteligencia durante la Guerra Fría: ¿cuántos misiles tiene la Unión Soviética? ¿cómo son de precisos? ¿cual es el potencial de guerra de Irak? Lo contrario a los rompecabezas son los "misterios", preguntas que no tienen una respuesta, en principio, definitiva. ¿Provocará Corea del Norte una nueva crisis nuclear? ¿Renunciará a su supremacía nacional el Partido Comunista de China? ¿Cuándo y dónde tendrá lugar el próximo ataque de Al Qaeda? Nadie conoce las respuestas a estas preguntas. Un misterio sólo puede ser "iluminado", pero no resuelto.*

Encontrar evidencias del profundo cambio desde el software propietario al basado en bienes comunes en el mundo comercial es parte de un misterio, no es un rompecabezas, por lo que nuestros medios tradicionales podrían no ajustarse adecuadamente. Pero, desde luego, estudiar los métodos de ingeniería que los proyectos de software libre usan no iluminará el contexto general, refiriéndonos a cómo cambia la industria del software cuando las corporaciones cambian sus modelos de negocio para ajustarse a los bienes comunes. Las preocupaciones de las compañías no son las mismas que las de alguien que usa una herramienta de seguimiento de errores, edita el código con Emacs, y automatiza una parte complicada del proceso de prueba. Además, como existen ideas preestablecidas adheridas a ciertos ideales de la ingeniería, probablemente veamos nuevas ideas en las que los ingenieros del software no hemos pensado.

Pongamos un ejemplo, de nuevo extraído del caso de estudio Sun. Un fabricante japonés de automóviles contactó con el Open Source Group de Sun con el objetivo de aprender sobre el software libre. El grupo de la empresa japonesa, era el responsable de la creación del grueso de las aplicaciones de la compañía. Decían no tener ni un solo programador empleado directamente, sino que estaban subcontratados, principalmente en la India. Estaban preocupados de que las empresas indias con las que tenían contratos no eran tan expertas interpretando las especificaciones que se les había dado como sería de desear desde un punto financiero. Así que el vicepresidente del grupo estaba interesado en que el Open Source Group de Sun les ayudara a imponer una metodología (que no realidad) de software libre en las

compañías subcontratadas, de forma que los encargados de las aplicaciones pudieran monitorizar el progreso, dirigir el trabajo nocturno, seguir las comunicaciones por email, wikis, etc. para poder juzgar cómo progresaba el proyecto y poder tomar medidas sobre la marcha, tal vez empleando técnicas de software libre.

Ni una sola línea de código sería publicada; no habría licencia. Sería simplemente una herramienta de administración. Los investigadores que tendrían conocimiento y realizarían informes sobre dichas innovaciones y actividades vendrían de una escuela de negocios, o serían economistas. Puede que, incluso, antropólogos. Por tanto, lo que vemos es que se requería un punto de vista más amplio, más interdisciplinar, lo que se adecua a las conclusiones obtenidas por los autores del informe de los sistemas ULS.

Otra parte del cambio es que los investigadores en software podrían realizar "ciencia real", basada en la evolución del software, sistemas, marcos de trabajo, etc. Por ejemplo, empezaría a tener sentido controlar cuántas veces se codifica un conjunto de datos en su camino desde una base de datos hasta un cliente, un número que podría ser muy alto si el sistema que realiza la transmisión en general está hecho con un número determinado de marcos de trabajo desarrollados separadamente. Hoy en día, obtener esta información precisa de una relación especial con una corporación, una relación que, sospecho, es bastante rara de encontrar.

## 6. Conclusiones

Uno puede preguntarse si los movimientos de Sun son predictivos o iconoclastas. Si se trata de lo segundo, entonces Sun es simplemente pura curiosidad; pero si se trata de lo primero, nos compete a aquellos de nosotros que estamos en el límite entre la investigación y la práctica concebir una especie de programa de investigación que nos ayude a darnos cuenta de los cambios, de forma que podamos registrarlos y estudiarlos.

## Referencias

- [1] C. Baldwin, K. Clark. *Design Rules: The Power of Modularity*. MIT Press, 1999. ISBN: 0262024667.
- [2] First ICSE Workshop on Software Technologies for Ultra-Large-Scale (ULS) Systems. <<http://www.cs.virginia.edu/~sullivan/ULS1/>>
- [3] The Software Engineering Institute (SEI). *The Software Challenge of the Future: Ultra-Large-Scale Systems*, Junio de 2006. <<http://www.sei.cmu.edu/uls/>>.
- [4] Gregory F. Tverton. Reshaping Intelligence to Share with "Ourselves". *Commentary No. 82, Canadian Security Intelligence Service*, Julio de 2003. <<http://www.csis-scrs.gc.ca/en/publications/commentary/com82.asp>>.