

**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). **Novática** edita también **Upgrade**, revista digital de CEPIS (Council of European Professional Informatics Societies), en lengua inglesa.

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

ATI es miembro de CEPIS (Council of European Professional Informatics Societies) y tiene un acuerdo de colaboración con ACM (Association for Computing Machinery). Tiene asimismo acuerdos de vinculación o colaboración con AdaSpain, AI2 y ASTIC

#### CONSEJO EDITORIAL

Antoni Carbonell Nogueras, Francisco López Crespo, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molins i Bertrán, Roberto Moya Quiles, César Pérez Chirinos, Mario Piattini Velthuis, Fernando Píera Gómez (Presidente del Consejo), Miquel Sarries Griñó, Carmen Ugarte García, Asunción Yturbe Herranz

**Coordinación Editorial**  
Rafael Fernández Calvo <[rfoalvo@ati.es](mailto:rfoalvo@ati.es)>

**Composición y autoedición**  
Jorge Llácer

**Traducciones**  
Grupo de Lengua e Informática de ATI  
Coordinadas por José A. Accino (Univ. de Málaga) <[jalfonso@ieev.uma.es](mailto:jalfonso@ieev.uma.es)>

**Administración**  
Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

#### SECCIONES TÉCNICAS: COORDINADORES

**Administración Pública Electrónica**  
Gumersindo García Arribas, Francisco López Crespo (MAP)  
<[gumersindo.garcia@map.es](mailto:gumersindo.garcia@map.es)>, <[flc@ati.es](mailto:flc@ati.es)>

**Arquitecturas**  
Jordi Tubella (DAC-UPC) <[jordit@ac.upc.es](mailto:jordit@ac.upc.es)>  
Victor Vihals Yuferra (Univ. de Zaragoza) <[vyuferra@unizar.es](mailto:vyuferra@unizar.es)>

**Auditoría SITIC**  
Marina Touriño, Manuel Palao (ASIA)  
<[marinatourino@marinatourino.com](mailto:marinatourino@marinatourino.com)>, <[manuel@palao.com](mailto:manuel@palao.com)>

**Bases de Datos**  
Coral Calero Muñoz, Mario G. Piattini Velthuis  
(Escuela Superior de Informática, UCLM)  
<[Coral.Calero@uclm.es](mailto:Coral.Calero@uclm.es)>, <[mpiattin@inf-cr.uclm.es](mailto:mpiattin@inf-cr.uclm.es)>

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

Isabel Davara Fernández de Marcos (Davara & Davara)  
<[isdavara@davara.com](mailto:isdavara@davara.com)>

**Enseñanza Universitaria de la Informática**  
Joaquín Ezpeleta Mateo (CPS-UIZAR) <[ezpeleta@posta.unizar.es](mailto:ezpeleta@posta.unizar.es)>

Cristóbal Pareja Flores (DSIP-UCM) <[cpajef@sisip.ucm.es](mailto:cpajef@sisip.ucm.es)>

**Informática y Filosofía**  
Josep Corco (UIC) <[jcorco@unica.edu](mailto:jcorco@unica.edu)>

Esperanza Marcos (ESSET-URJC) <[euca@esset.urjc.es](mailto:euca@esset.urjc.es)>

**Informática Gráfica**  
Roberto Vivo (Eurographics, sección española) <[rvivo@dsic.upv.es](mailto:rvivo@dsic.upv.es)>

**Ingeniería del Software**  
Javier Dolado Cosin (DLSI-UPV) <[dolado@si.ehu.es](mailto:dolado@si.ehu.es)>

Luis Fernández (PRIS-EL-UEM) <[lufern@pris.esi.uem.es](mailto:lufern@pris.esi.uem.es)>

**Inteligencia Artificial**  
Federico Barber, Vicente Botti (DSIC-UPV)  
<[fvbotti@barber@dsic.upv.es](mailto:fvbotti@barber@dsic.upv.es)>

**Interacción Persona-Computador**  
Julio Abascal González (PI-UPV) <[julio@si.ehu.es](mailto:julio@si.ehu.es)>

Jesús Lorés Vidal (Univ. de Lleida) <[jesus@eup.udl.es](mailto:jesus@eup.udl.es)>

**Internet**  
Alonso Álvarez García (TID) <[alonso@ati.es](mailto:alonso@ati.es)>

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

**Lengua e Informática**  
M. del Carmen Ugarte (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. Angel Velázquez (ESSET-URJC) <[a.velazquez@esset.urjc.es](mailto:a.velazquez@esset.urjc.es)>

**Libertades e Informática**  
Alfonso Escolano (FIR-Univ. de La Laguna) <[aescolan@ull.es](mailto:aescolan@ull.es)>

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

Manuel Palomar (Univ. de Alicante) <[mpalomar@dlsi.ua.es](mailto:mpalomar@dlsi.ua.es)>

**Mundo estudiantil**  
Adolfo Vázquez Rodríguez  
(Rama de Estudios del IEEE-UCM) <[a.vazquez@iee.org](mailto:a.vazquez@iee.org)>

**Profesión informática**  
Rafael Fernández Calvo (ATI) <[rfoalvo@ati.es](mailto:rfoalvo@ati.es)>

Miquel Sarries Griñó (Ayto. de Barcelona) <[msarries@ati.es](mailto:msarries@ati.es)>

**Redes y servicios telemáticos**  
Luis Guijarro Coloma (DCOM-UPV) <[lguijar@dcom.upv.es](mailto:lguijar@dcom.upv.es)>

Josep Solé Pareta (DAC-UPC) <[pareta@ac.upc.es](mailto:pareta@ac.upc.es)>

**Seguridad**  
Javier Areitio (Redes y Sistemas, Bilbao) <[jareitio@orion.deusto.es](mailto:jareitio@orion.deusto.es)>

Composicion, Edición y Redacción ATI Valencia  
(ESYC-URJC) <[fgb.pheras@gsyc.esset.urjc.es](mailto:fgb.pheras@gsyc.esset.urjc.es)>

**Sistemas de Tiempo Real**  
Alejandro Alonso, Juan Antonio de la Puente  
(DIT-UPM) <[jaalonso,jpuente@dit.upm.es](mailto:jaalonso,jpuente@dit.upm.es)>

**Software Libre**  
Jesús M. González Barahona, Pedro de las Heras Quirós  
(ESYC-URJC) <[jmg.gheras@gsyc.esset.urjc.es](mailto:jmg.gheras@gsyc.esset.urjc.es)>

**Tecnología de Objetos**  
Jesus Garcia Molina (DIS-UM) <[jmolina@correo.um.es](mailto:jmolina@correo.um.es)>

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

**Tecnologías para la Educación**  
Josep Sales Ruffi (ESPIRAL) <[jsales@pie.xtec.es](mailto:jsales@pie.xtec.es)>

**Tecnologías y Empresa**  
Pablo Hernández Medrano (Bluemat) <[pablohm@bluemat.biz](mailto:pablohm@bluemat.biz)>

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

**TIC para la Sanidad**  
Valentín Masero Vargas (DI-UNEX) <[vmasero@unex.es](mailto:vmasero@unex.es)>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción de todos los artículos, salvo los marcados con © o *copyright*, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

**Coordinación Editorial y Redacción Central (ATI Madrid)**  
Padilla 66, 3º, dcha., 28006 Madrid  
Tlf. 914029391; fax. 913093685 <[novatica@ati.es](mailto:novatica@ati.es)>

**Composicion, Edición y Redacción ATI Valencia**  
Reino de Valencia 23, 46005 Valencia  
Tlf./fax. 963330392 <[secreval@ati.es](mailto:secreval@ati.es)>

**Administración y Redacción ATI Cataluña**  
Via Laietana 41, 1º, 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**  
Lagasca 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)>

**Redacción ATI Galicia**  
Recinto Ferial s/n, 36540 Silleda (Pontevedra)  
Tlf. 986581413; fax. 986580162 <[secregal@ati.es](mailto:secregal@ati.es)>

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

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

**Imprenta:** 9-Impressió S.A., Juan de Austria 66, 08005 Barcelona.  
**Depósito Legal:** B 15.154-1975  
**ISSN:** 0211-2124; CODEN NOVAEC

**Portada:** Antonio Crespo Foix / © ATI 2003

## SUMARIO

En resumen: El procomún del conocimiento **2**  
*Rafael Fernández Calvo*

**Monografía: Conocimiento abierto / Open Knowledge**  
(En colaboración con **Upgrade**)

Editores invitados: *Philippe Aigrain* y *Jesús M. González Barahona*

**Presentación. Propiedad y uso de la información y del conocimiento: ¿privatización o procomún?** **3**

*Philippe Aigrain, Jesús M. González-Barahona*

**La Economía Política del procomún** **6**

*Yochai Benkler*

**El redescubrimiento del procomún** **10**

*David Bollier*

**La lengua en el medio digital: un reto político** **13**

*José Antonio Millán*

**Nota sobre las patentes de software** **16**

*Pierre Haren*

**Sobre la patentabilidad de las invenciones referentes a programas de ordenador** **17**

*Alberto Bercovitz Rodríguez Cano*

**Eligiendo la herramienta legal correcta para proteger el software** **21**

*Roberto Di Cosmo*

**Por favor, ¡pirateen mis canciones!** **24**

*Ignacio Escobar*

**La normativa europea y norteamericana sobre propiedad intelectual en el 2003: protección legal antipiratero y derechos digitales** **26**

*Gwen Hinz*

**'Informática de confianza' y política sobre competencia: temas a debate para profesionales informáticos** **30**

*Ross Anderson*

**Secciones Técnicas**

Lengua e Informática

**El software libre y las lenguas minoritarias: una oportunidad impagable** **36**

*Jordi Mas i Hernández*

Lenguajes informáticos

**Evaluación parcial de programas y sus aplicaciones** **40**

*Pascual Julián Iranzo*

**COMPAS: un compilador para un lenguaje imperativo con aserciones embebidas** **47**

*Joaquín Ezpeleta Mateo, Pedro Gascón Campos, Natividad Porta Royo*

Seguridad

**Ocultación de imágenes mediante Esteganografía** **52**

*David Atauri Mezquida, Luis Fernández Sanz,*

*Matías Alcojor, Ignacio Acero*

**La confianza y la seguridad aspectos vitales para los servicios electrónicos** **58**

*José A. Mañas Argemí*

Sistemas de Tiempo Real

**Sistemas Linux de tiempo real** **63**

*Javier Miqueleiz Álamos*

Referencias autorizadas

**Sociedad de la Información** **69**

Personal y transferible

**Locos por los ordenadores (II):**

**Ada Byron y Charles Babbage, o la bella y la bestia** **75**

*Rafael Fernández Calvo*

Asuntos Interiores

Coordinación editorial / Programación de Novática

Normas de publicación para autores / Socios Institucionales

**Monografía del próximo número:**

**«Ingeniería del Software: estado de un arte»**

Conocimiento abierto / *Open Knowledge*

Roberto Di Cosmo  
 Universidad París VII, Francia

<roberto@dicosmo.org>

Traducción: José María Cañas Plaza

**Resumen:** este artículo investiga los méritos relativos de las diferentes herramientas legales para proteger el software, en el justo momento en que la Unión Europea está considerando cambiar su política pública en estos temas. Ofrecemos algunos argumentos claros sobre el impacto que las diferentes herramientas tienen en el desarrollo y la innovación, y urgimos a los lectores a que formen su propia opinión.

**Palabras clave:** código abierto, derechos de autor, marcas registradas, patentes de software, política pública, propiedad intelectual, software libre.

## 1. Introducción

Si hoy día está usted en el negocio de desarrollo de software, entonces tiene a su disposición tres clases de herramientas legales con el potencial de proteger su trabajo de la competencia irrespetuosa: la marca registrada, los derechos de autor y las patentes. Es destacable que en los últimos años la ley de marcas registradas se ha aplicado uniformemente en el dominio de los programas sin ninguna voz en contra y que la ley de derechos de autor se ha usado durante décadas sin más discusión (hay algunas reticencias hacia los medios propuestos recientemente para proteger los derechos de autor, pero no contra los derechos de autor en sí mismos). Por el contrario la aplicación de la ley de patentes en los Estados Unidos, y su introducción en Europa en lo que refiere al software, se ha encontrado con una profunda reacción adversa de muchos integrantes del negocio del software y del campo de la investigación. Me gustaría ofrecer mi opinión desde el entorno académico europeo de la Ingeniería Informática<sup>1</sup>.

## 2. Herramientas legales de protección del software: ¿marcas registradas, derechos de autor o patentes?

Todos los que intervienen en este debate están de acuerdo en que deberíamos encontrar fórmulas para ofrecer a los inventores una justa recompensa por su esfuerzo, en la medida que tal esfuerzo produce un resultado beneficioso para la comunidad. El gran debate es sobre cuál—entre marca registrada, derechos de autor y patentes— es la herramienta apropiada que recompense justamente las invenciones a la vez que protege el interés general y la libre competencia (mucho de los actores consideran oficialmente equivalentes estos dos últimos).

Frecuentemente se oyen afirmaciones como «*Internet es el futuro, la riqueza del futuro es la Propiedad Intelectual y proteger la Propiedad Intelectual es necesario; de ahí que necesitemos las patentes de software*», que son bastante

## Eligiendo la herramienta legal correcta para proteger el software

Este artículo se publica bajo la licencia *Free Document Dissemination Licence*

falsas, dado que existen más formas de proteger las propiedades intelectuales que las patentes<sup>2</sup>. De hecho, se puede construir un mercado bastante considerable simplemente a través de marcas registradas, como hace la distribución de Linux Red Hat<sup>3</sup>, y se puede construir un monopolio de sistemas operativos durante décadas simplemente con los derechos de autor, como hizo Microsoft<sup>4</sup>. Por lo tanto uno puede preguntarse legítimamente qué hay realmente detrás de ese interés repentino por las patentes de software, en un clima de urgencia apresurada, después de décadas de innovación exitosa en los programas bajo la protección de la propiedad intelectual, que garantiza tanto la marca registrada como el derecho de autor.

## 3. Derechos de autor versus patentes

Antes que nada, se necesita comprender claramente que las bases legales de las patentes difieren *profundamente* de las del derecho de autor (para mayor claridad de ahora en adelante prescindiré de las marcas registradas). Además no entraré en los detalles técnicos de las diferencias entre la ley del derecho de autor en Estados Unidos y la ley francesa o europea, dado que son disparidades menores respecto de las diferencias con la ley de patentes.

### 3.1. Qué se protege y cuánto poder se otorga al autor

El derecho de autor protege un programa específico ya existente, del mismo modo que protege una novela existente y específica, como por ejemplo «El sabueso de Baskerville» de Arthur Conan Doyle, de ser copiada o reescrita sin autorización. Por ejemplo, usted no puede tomar el trabajo de Conan Doyle, cambiar unas cuantas líneas, reemplazar su nombre en la portada por el suyo e ir a vender el libro como suyo.

Pero esto es lo máximo que la propiedad intelectual puede hacer: Conan Doyle no tiene el derecho de evitar que otros

#### Autor

**Roberto Di Cosmo** es profesor titular en la Universidad París VII, Francia, donde es responsable de estudios de postgrado en Ingeniería Informática, después de haber impartido clases por varios años en la Ecole Normale Supérieure de París. Además es miembro investigador del INRIA. Sus intereses de investigación incluyen métodos formales, lógica matemática, programación funcional y programación paralela. En los últimos cinco años ha estado activamente involucrado en las reflexiones sobre el impacto de la informática en nuestra sociedad. <<http://www.dicosmo.org>>

escriban historias sobre cierto detective muy inteligente que investiga sobre casos intrincados y misteriosos. Afortunadamente, pues en caso contrario Poirot y tantos otros personajes apreciados por nosotros jamás hubieran visto la luz (por no mencionar que con una propiedad intelectual tan fuerte Edgar Allan Poe podría haber impedido que Conan Doyle escribiese sobre Sherlock Holmes).

En el escenario de la programación esto significa que no se puede robar el programa de otro (cosa que sin embargo ciertos vendedores de programas hacen de cuando en cuando), *pero usted es muy dueño de escribir un programa con funcionalidad similar*. Digámoslo otra vez: no se puede tomar el código del Office de Microsoft e incluirlo en un producto nuestro, pero está permitido bajo la ley de derechos de autor escribir un plataforma equivalente o mejor de aplicaciones de oficina, como el proyecto OpenOffice ha hecho exitosamente.

Esto suena bastante justo a primera vista: se protege a los autores de que su trabajo sea pirateado, pero al mismo tiempo se permite la libre competencia de méritos, haciendo a los autores competir en pos de la mejor novela de ficción o la mejor plataforma de aplicaciones de oficina, y la interacción cruzada de ideas que así es posible termina aumentando la riqueza global, para satisfacción general.

Por otra parte, la ley de patentes se introdujo sobre bases muy diferentes a la ley de propiedad intelectual: la idea era motivar a los inventores a difundir al público los detalles de sus invenciones, haciéndolas accesibles pronto para fomentar mayores innovaciones, en lugar de mantener en secreto esos detalles para proteger su negocio. En compensación por esa difusión los inventores recibieron un regalo precioso: un monopolio forzado y enteramente legal sobre la explotación comercial de cualquier posible materialización de su invento durante un periodo limitado (en la práctica librándoles de las molestias de la competencia del libre mercado durante tal periodo). Para que este fuera un trato justo, todo el mundo debería cumplir su parte del pacto: el inventor debe difundir una invención beneficiosa no trivial y demostrar que realmente funciona, que no es una quimera o una reinención de la rueda para obtener su patente. Esta es la razón por la cual, a diferencia de los derechos de autor que incluso protegen a los autores de las más estúpidas e inútiles obras de literatura, o trozos de código, las patentes no son automáticas, sino que deben ganarse en la oficina de patentes, que se supone que escrutan en profundidad la propuesta de patente antes de aceptarla.

¿Se adapta bien la ley de patentes a los programas? De acuerdo con el Artículo 52 del Convenio Europea sobre Patentes [1][2], en sus párrafos (2)(c) y (3), la respuesta es claramente no. Pero unas cuantas fuerzas poderosas están echando el resto para cambiar esta situación, argumentando que un *no* es la respuesta adecuada y presionando a la Unión Europea para votar una propuesta de directiva que introduciría las patentes de software en Europa [3].

Afortunadamente no tenemos que recurrir a experimentos mentales filosóficos para comprender lo que tal cambio en la directiva comunitaria significaría para el negocio del software: hace tiempo que un país transatlántico se embarcó en el camino de las patentes de software, así que podemos mirar para allá y ver lo que ha sucedido. Lo que encontramos cuando miramos a las patentes de software que han sido

concedidas es que ‘protegen’ un método abstracto de resolver un problema y no un trozo específico del código de un programa, que puede no haberse escrito aún siquiera para resolver el problema. A pesar de la desbordante cantidad de argumentos sofisticados y variados que se pueden encontrar sobre la materia hay un hecho sólido que hace tales patentes profundamente diferentes de los derechos de autor de un programador: si la ley de patentes se extendiera al software, entonces *usted no podría escribir un programa con funcionalidad similar a la de uno protegido por la patente de la competencia, sin importar cuanto mejor fuera el nuevo programa que usted escribiera, ni cuan mal escrito estuviera el del competidor, o aún peor no importaría si el competidor hubiera realmente escrito ya el programa ¡o si no tuviera intención de hacerlo nunca!* Para que usted escribiera su propio programa debería obtener una licencia del dueño de la patente, y esto significa que usted debería gastar algo de dinero en conseguirlo. Pero no todo es cuestión de dinero: el propietario de la patente podría denegar la licencia de una tecnología clave a todo aquel que no le gustara.

Creo que un ejemplo es bastante apropiado para clarificar lo que una patente puede hacer: muy recientemente una gran compañía de software puso disponibles a través de la Red parte de las especificaciones técnicas de un sistema de ficheros en red, obligada por una orden judicial, pero se las arregló para introducir disimuladamente en la versión definitiva unas cuantas líneas que prohíben a los desarrolladores de software libre escribir código compatible con tal especificación<sup>5</sup>.

¿Hubiéramos aceptado la idea de que al señor Conan Doyle se le hubiera otorgado el derecho legal de declarar que ninguna mujer estaría autorizada para escribir novelas ‘compatibles’ con la suya, es decir, que tratase sobre detectives que resuelven misterios gracias a sus habilidades deductivas? De ese modo se habría prohibido que Agatha Christie hubiera escrito sus trabajos.

### 3.2. Licencia para matar

Escribir programas significa, en la vida corriente, resolver docenas de diferentes problemas ‘nuevos’, aunque similares, combinando una serie de técnicas básicas que forman el dinámico ‘estado del arte’. Ordenar valores, disponerlos en estructuras eficientes de memoria, entregar interfaces de usuario claras, equilibrar la carga en un conjunto de procesadores, intercambiar datos entre ordenadores remotos, leer y escribir diferentes formatos de datos y muchas otras tareas son parte del trabajo diario de todo programador, y lo que hace su trabajo único es muy frecuentemente no la selección de algoritmos o la estructura de datos, sino la manera única en la que el programador convierte el algoritmo y la estructura de datos en código concreto y real.

Este hecho se confirma a través de nuestra experiencia docente diaria: cuando mandamos deberes a los estudiantes, normalmente ocurre que dado un problema concreto claramente especificado, sólo hay un reducido número de algoritmos o estructuras de datos que proporcionan la solución más eficiente, cuando no uno único. Aún así no hay dos programas que sean prácticamente iguales (a menos que alguien viole el derecho de autor, evidentemente).

Dicho esto, cualquier sistema de software mínimamente sofisticado puede necesitar código que utilice miles de métodos,

desde ideas sencillas a algoritmos muy complicados. Mientras que se puede estar de acuerdo en que los algoritmos complejos merecen reconocimiento, por cuanto que representan avances significativos en la ciencia, rara vez se da el caso de que tales avances se difundan a cambio de la patente (el protocolo de clave pública RSA es una sobresaliente excepción): la mayor parte de las veces tales avances se publican en revistas científicas, para que todo el mundo los lea, lo cual es bastante razonable, puesto que en la mayoría de las ocasiones tales resultados se obtienen gracias a la financiación pública.

Por otro lado una cantidad preocupantemente grande de patentes de software que se han concedido en Estados Unidos cubren ideas desesperantemente sencillas, como el heurístico “del viajante estresado” para encontrar una comida rápida (patente #5.249.290), o variantes de los borradores de trabajo de cierto comité de estandarización (la bien conocida patente de las Hojas de Estilo en Cascada —CSS— concedida a Microsoft), o incluso métodos triviales para la práctica de negocios, como la patente del *compra con un solo click* detentada por Amazon<sup>6</sup>.

### 3.3. Las patentes de software son especiales

El gran problema está en que frecuentemente no hay modo de zafarse de las ideas sencillas y por lo tanto no hay modo de asegurar la interoperabilidad con protocolos, formatos o estándares patentados, si el dueño de la patente se niega a conceder licencias sobre ellos.

Esto significa que una patente lo suficientemente amplia, difusa o trivial se convierte en una verdadera arma para matar a los competidores: consideremos el caso de una gran compañía enfrentándose a una joven empresa que nace con una idea maravillosa y que tiene el potencial de convertirse en un serio competidor de la primera dentro de unos años. La empresa joven puede desde luego patentar su idea. Sin embargo nunca llegará a convertirla en un producto: tan pronto como trate de implementar el código, será evidente inmediatamente que tal código implementa un amplio rango de funcionalidad accesoria, y es probable que al menos una de ellas esté cubierta por una patente trivial que tiene la compañía grande.

Por esto que la única solución viable será que la compañía grande compre a la pequeña, lo cual puede ser algunas veces conveniente para el capital de riesgo invertido, que de este modo puede conseguir un moderado beneficio, pero nunca será satisfactorio en términos de competencia y el mercado libre, e incluso menos en términos de innovación, que se ve menguada por tales prácticas.

Esta es probablemente la razón por la cual apenas se ve un limitado número de casos en los tribunales: o ambas compañías contendientes son grandes, y entonces llegan a un acuerdo para intercambiarse gratuitamente licencias, o una de las dos es un pez pequeño, que es devorado generalmente sin mucho alboroto.

## 4. Actuar ahora

En resumen, un gran problema de las patentes de software es que en el negocio del software conseguir una idea patentable es normalmente fácil, mientras que desarrollar programas que usan una idea auténticamente no trivial puede ser extremada-

mente costoso y complejo, y normalmente implica usar muchas ideas triviales, que frecuentemente suelen estar patentadas por algún tipo listo, que tiene de este modo el dedo puesto en el gatillo de un arma que apunta directamente a nuestro propio negocio. Y en tal situación, este tipo listo ni siquiera necesita ser una empresa de software<sup>7</sup>.

La próxima vez que se ponga a pensar si la ley de derechos de autor o la de patentes es la más apropiada para los programas, recuerde que la ley de derechos de autor permite proteger costosos desarrollos, y lleva haciéndolo durante mucho tiempo, mientras que de la ley de patentes se puede abusar para anular tales costosos desarrollos por medio de alegaciones de violación sobre alguna oscura trivialidad que equivale en programación a la reinención de la rueda.

Entonces, pregúntese si es aceptable conceder una licencia para matar a un puñado de grandes compañías y permitirles de este modo medrar a sus anchas en medio de la vasta comunidad de desarrolladores de software que están trabajando en la pequeña y mediana empresa y que son el mejor activo para construir la sociedad y el mercado del futuro. En el caso de que considere que podría ser un problema, debería tomarse unos minutos para hacer que sus representantes en el Parlamento Europeo sean conscientes de su opinión, mientras aún se está a tiempo.

## Referencias

- [1] **Convenio Europeo sobre Patentes**, Artículo 52, 1978. <<http://www3.european-patent-office.org/legal/epc/e/ar52.html>>.
- [2] **Directiva del Consejo sobre la Protección Legal de los Programas de Ordenador (91/250/EEC)**, 14 Mayo 1991. <<http://europa.eu.int/eur-lex/en/index.html>>.
- [3] **Propuesta para una Directiva del Parlamento Europeo y del Consejo de Patentabilidad de invenciones materializadas en computador**. Boletín Oficial C 151 E, páginas 129-131, 25 Junio 2002.
- [4] **J. Ullman**. *Ordinary skills in the art*. Conferencia en la entrega del Premio Knuth <<http://www-db.stanford.edu/ullman/pub/focs00.tml>>.

## Agradecimientos

El autor quiere agradecer a Juliusz Chroboczek sus interesantes comentarios sobre la primera versión de este artículo.

## Notas

- <sup>1</sup> También invito al lector a leer la significativa opinión del profesor Ullman sobre la materia [4].
- <sup>2</sup> Intencionadamente he omitido argumentos como «*las patentes se necesitan para conseguir inversiones de capital de riesgo*», o el ridículo pero real «*una gran compañía del otro lado del charco envió un experto para explicarme por qué necesito patentes de software para ser competitivo frente a ellos*».
- <sup>3</sup> Para aquellos que todavía no lo saben, se puede descargar una copia íntegra del CD de Red Hat desde la Red, sin coste alguno. Su licencia no sólo le permite a usted hacerlo, sino que además se le anima a ello.
- <sup>4</sup> Microsoft ha entrado recientemente en el juego de las patentes de software.
- <sup>5</sup> Véase <[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/Finalcifs\\_LicenseAgrmnt\\_032802.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/Finalcifs_LicenseAgrmnt_032802.asp)>
- <sup>6</sup> Permítame ser claro sobre esto: Amazon hizo un trabajo estupendo en la venta en línea de libros, y construyó un maravilloso sistema para respaldar su creciente negocio, incluyendo una implementación bastante sofisticada de un sistema seguro que permite comprar con un solo click a sus muchos clientes. Pero mientras la implementación es sofisticada y debe estar protegida, como de hecho lo está por los derechos de autor, la idea de comprar con un solo click es una trivialidad.
- <sup>7</sup> Los Estados Unidos han visto recientemente el nacimiento de una peculiar compañía que no se dedica al software, cuyo negocio consiste precisamente en acumular patentes y entonces perseguir a compañías de programadores que han tenido moderado éxito para forzarles a ‘pagarles por sus derechos’, amenazando con demandarles.