

**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), 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**, **Ai2** y **ASTIC**.

**Consejo Editorial**

Antoni Carbonell Noguera, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte, Francisco López Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codiña, Fernando Peña Gómez (Presidente del Consejo), Ramon Puigjaner Trapal, Miguel Sarrías Grinó, Asunción Yturbe Herranz

**Coordinación Editorial**

Rafael Fernández Calvo < rfcvalvo@ati.es >

**Composición y autoedición**

Jorge Llacer Gil de Ramaltes

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**

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**

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

Jordi Tubella Margadas (DAC-UPC) < [jordi@ac.upc.es](mailto:jordi@ac.upc.es) >

**Auditoría ATIS**

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

**Derecho e tecnologías**

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

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

**Escuela Universitaria de la Informática**

José María Espinosa Mateo (CPS-UZAR) < [espinosa@posta.unizar.es](mailto:espinosa@posta.unizar.es) >

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

**Geografía del Conocimiento**

Juan Benet Solé (Cp Genmi Ernst & Young) < [joan.baiget@ati.es](mailto:joan.baiget@ati.es) >

**Informática y Filosofía**

Josep Corco Juvinyà (UIC) < [jjcorco@unicep.edu](mailto:jjcorco@unicep.edu) >

Esperanza Marcos Martínez (ESCET-URJC) < [cuca@escet.urjc.es](mailto:cuca@escet.urjc.es) >

**Informática Crítica**

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

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

**Ingeniería del Software**

Javier Dolado Cossin (DIA-UPV) < [dolado@si.ehu.es](mailto:dolado@si.ehu.es) >

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

**Inteligencia Artificial**

Federico Barber Sánchez, Vicente Botti Navarro (DSIC-UPV) < [fvbotti\\_fbarber@dsic.upv.es](mailto:fvbotti_fbarber@dsic.upv.es) >

**Interacción Persona-Computador**

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

Jesús Llorés Vidal (Univ. de Lleida) < [jesus@esp.upl.edu](mailto:jesus@esp.upl.edu) >

**Internet**

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

Lluisen Pages Casas (Andra) < [pages@ati.es](mailto:pages@ati.es) >

**Lengua e Informática**

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

**Lenguajes Informáticos**

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

J. Angel Velázquez Irujo (ESCET-URJC) < [a.velazquez@escet.urjc.es](mailto:a.velazquez@escet.urjc.es) >

**Lingüística computacional**

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

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

**Mundo estudiantil**

Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM) < [a.vazquez@ieee.org](mailto:a.vazquez@ieee.org) >

**Protección Informática**

Rafael Fernández Calvo (ATI) < [rfcalvo@ati.es](mailto:rfcalvo@ati.es) >

Miguel Sarrías Grinó (Ayto. de Barcelona) < [msarrias@ati.es](mailto:msarrias@ati.es) >

**Redes y servicios telemáticos**

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

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

**Seguridad**

Javier Arellano Bertolin (Univ. de Deusto) < [jarellano@eside.deusto.es](mailto:jarellano@eside.deusto.es) >

Javier López Muñoz (ETS Informática-UMA) < [jlmu@cc.uma.es](mailto:jlmu@cc.uma.es) >

**Sistemas de Tiempo Real**

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM) < [jaalonso@puente.com](mailto:jaalonso@puente.com) >

**Software Libre**

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

**Tecnología de Datos**

Jesús García Molina (DS-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**

Juan Manuel Doderó Beardo (UC3M) < [doderom@inf.uc3m.es](mailto:doderom@inf.uc3m.es) >

Julia Miragallón i Alfores (UOC) < [jmiragallon@uoc.edu](mailto:jmiragallon@uoc.edu) >

**Tecnologías y Empresa**

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

**TiE y Turismo**

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

**UPGRADE**

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  
 Tfn. 91 4029391; fax 91 3093685 < [novatica@ati.es](mailto:novatica@ati.es) >

**Composición, Edición y Redacción ATI Valencia**  
 Av. del Reino de Valencia 23, 46005 Valencia  
 Tfn./fax 963330392 < [secre@ati.es](mailto:secre@ati.es) >

**Administración y Redacción ATI Cataluña**  
 Ciudad de Granada 131, 08018 Barcelona  
 Tfn. 93 41 25 53; fax 93 41 27 713 < [secregen@ati.es](mailto:secregen@ati.es) >

**Redacción ATI Andalucía**  
 Isaac Newton, s/n, Ed. Sadiel,  
 Isla Cartuja 41092 Sevilla, Tfn./fax 95 44 60 779 < [secreand@ati.es](mailto:secreand@ati.es) >

**Redacción ATI Aragón**  
 Lagasca 9, 3-B, 50006 Zaragoza,  
 Tfn./fax 91 6235181 < [secreara@ati.es](mailto:secreara@ati.es) >

**Redacción ATI Asturias-Santander** < [gp\\_astucant@ati.es](mailto:gp_astucant@ati.es) >

**Redacción ATI Castilla-La Mancha** < [gp-clmcha@ati.es](mailto:gp-clmcha@ati.es) >

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

**Publicidad**  
 Padilla 66, 3º dcha., 28006 Madrid  
 Tfn. 91 4029391; fax 91 3093685 < [novatica.publicidad@ati.es](mailto:novatica.publicidad@ati.es) >

**Imprenta**  
 Derra S.A., Juan de Austria 66, 08005 Barcelona.

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

**Periodo**: Antonio Crespo Foix / © ATI 2006

**Diseño**: Fernando Agresta / © ATI 2006

**editorial** > 02  
**REICIS, nueva revista de ATI**  
**La digitalización completa de Novática**  
**El XL aniversario de ATI**

**en resumen** > 03  
**La Ingeniería del Software en la práctica diaria**  
*Rafael Fernández Calvo*

**noticias de IFIP** > 04  
**Reunión del Council de IFIP en Palma**

**monografía**

**Factores clave de éxito en Ingeniería del Software**  
 (En colaboración con UPGRADE)  
 Editores invitados: *Luis Fernández Sanz, Juan José Cuadrado Gallego y Maya Daneva*  
**Presentación. La Ingeniería del Software: más allá de una visión académica** > 05  
*Luis Fernández Sanz, Juan José Cuadrado Gallego, Maya Daneva*

**Análisis de la Ingeniería del Software desde una perspectiva de Ingeniería** > 07  
*Alain Abran, Kenza Meridji*

**Utilizando UML™ 2.0 para resolver problemas de Ingeniería de Sistemas** > 21  
*Ian Barnard*

**Aplicando la medición de software orientada a servicios para obtener indicadores de calidad en componentes de código abierto** > 34

*René Braungarten, Ayaz Farooq, Martin Kunz, Andreas Schmietendorf, Reiner R. Dumke*  
**El repositorio de proyectos software ISBSG & ISO 9126: una oportunidad para medir la calidad** > 41

*Laila Cheikh, Alain Abran, Luigi Buglione*  
**El factor humano en la Ingeniería del Software** > 48

*Luis Fernández Sanz, María José García García*

**secciones técnicas**

**Lengua e Informática**  
**Phishing y pharming** > 55

*Grupo de Lengua e Informática de ATI, Lista Spanglish (Mª del Carmen Ugarte García, ed.)*

**Lenguajes Informáticos**  
**Procesamiento de páginas web con herramientas Java y XML** > 57

*Mireia Ribera Turró*

**Referencias autorizadas** > 62

**sociedad de la información**

**Personal y transferible**  
**Patentes de Software, situación tras el rechazo europeo** > 67

*Alberto Barrionuevo García*

**Programar es crear**  
**Estrellas diabólicas (CUPCAM 2005, problema F, enunciado)** > 73

*Cristóbal Pareja Flores*

**Programas equivalentes (CUPCAM 2005, problema E, solución)** > 74  
*Manuel Carro Liñares, Manuel Freire Morán*

**asuntos interiores**

**Coordinación editorial / Programación de Novática** > 76  
**Normas de publicación para autores / Socios Institucionales** > 77

Alberto Barrionuevo García  
Foundation for a Free Information  
Infrastructure (FFII), España

<abarrio@ffii.org>

# Patentes de Software, situación tras el rechazo europeo



Alberto Barrionuevo García, FFII, 2006. Este artículo está acogido a los términos de la licencia "Creative Commons Attribution-NonCommercial-NoDerivs 2.5 Spain", <<http://creativecommons.org/licenses/by-nc-sa/2.5/es/>>

## 1. Historia y situación legal

### 1.1 Historia de la protección intelectual del software

En el origen de la informática y la computación, la separación económica entre software y hardware no existía. El software era un complemento gratuito y sin valor económico alguno que se proveía junto con el hardware, que era donde de verdad residía el valor comercial. Se puede decir que en esos momentos todo el software era dominio público y de código abierto (no tenía sentido esconder algo sin valor).

Durante la década de los sesenta fue viéndose que esta situación no podía mantenerse más porque el software cada vez aportaba un mayor valor añadido a la solución final y, además, su coste de creación se estaba volviendo más y más significativo.

Ante este panorama, los actores de la industria y la legislación se reunieron a finales de la década para decidir cómo proteger ese nuevo activo que era el software. Las opciones eran tres, utilizar la propiedad intelectual (derecho de autor o *copyright*), utilizar la propiedad industrial (patente) o desarrollar una protección sui-generis para el software. Las dos últimas opciones fueron descartadas y se optó por usar el derecho de autor. La principal razón económica para descartar la patente fue el temor a que la multinacional preponderante de la época, IBM, monopolizara todo el mercado del software con sus patentes.

Así, el *copyright* es la figura que ha protegido todo el software (los programas informáticos) hasta nuestros días. Tanto el Convenio de Berna (1886-1979)<sup>1</sup>, como el TRIPS (*Trade-Related Aspects of Intellectual Property Rights*, 1994)<sup>2</sup>, como las vigentes directiva europea (1991)<sup>3</sup> y ley española (1996)<sup>4</sup>, rezan que los programas de ordenador deben ser protegidos como obras literarias, por ser desarrollos intelectuales.

Con el surgimiento mercantil de los programas de código cerrado, a esta figura del *copyright* se le unieron tres recursos legales más: el secreto industrial y comercial, la marca y, derivado del *copyright*, los acuerdos de licencia de usuario final (EULAs, *End-User License Agreement*). Queda la situación actual de forma que el software propietario o privativo se regula mediante

**Resumen:** todo el exitoso desarrollo de la ciencia y economía del software ha estado basado en cuatro pilares legales: derechos de autor (*copyright*), secreto industrial y comercial, y, ligado al derecho de autor, los acuerdos de licencia de usuario final (EULAs, *End User License Agreement*). En EE.UU., desde principios de los 80 y por medio de la jurisprudencia, se fue añadiendo paulatinamente una controvertida quinta figura legal, que es la patente (de software), y que nunca ha conseguido probar su interés económico para la sociedad y la innovación en software. Sin embargo, no ha sido hasta el estallido del "punto COM", cuando las patentes de software han jugado un papel significativo en la economía del software. En Europa, con legislación distinta a la estadounidense, las patentes en general se regulan por el Convenio de la Patente Europea de Munich del 1973 (CPE). En él se cita explícitamente a los programas de ordenador como un sujeto no patentable. Sin embargo, la Oficina Europea de Patentes (OEP) hace imaginativas interpretaciones favorables a la masiva patentabilidad de software, constituyendo ésta su práctica actual. Los pocos juicios de relevancia que se están produciendo en la Unión Europea están decantándose mayormente por una interpretación más restrictiva a la patentabilidad del software y más cercana a la postura contraria a la misma del Parlamento Europeo en la recientemente rechazada directiva de patentes de software. En España se puede concluir que existe una ley de las más restrictivas de Europa en cuanto a la patentabilidad del software, aunque esto no evita que también exista cierta controversia legal generada por la malabarística interpretación del CPE realizada por la OEP.

**Palabras clave:** Asociación para una Infraestructura de Información Libre, Comisión Europea, FFII, invención asistida por computadora, invención implementada en ordenador, OEP, OEPM, Oficina Española de Patentes y Marcas, Oficina Europea de Patentes, Parlamento Europeo, patente, patente comunitaria, software, software libre.

esas cuatro figuras, y el software libre sólo mediante el *copyright*, por ejemplo en su acepción *copyleft*, y con las EULAs asociadas (GPL o *General Public License*, BSD o *Berkeley Software Distribution*, etc.)

### 1.2. Historia de la patente de software

#### 1.2.1. Estados Unidos de América

La práctica legal en EE.UU. cuenta con un concepto que no existe de igual forma y preponderancia en la Europa continental, aunque sí en Reino Unido: la jurisprudencia. En EE.UU., la interpretación de una ley, o la decisión a falta de ella, que tome el primer juez que juzga una nueva materia crea jurisprudencia y confiere un gran peso de referencia a su decisión para posteriores juicios sobre esa materia. En la Europa continental, sin embargo, es común que, ante las mismas circunstancias, varios jueces tomen decisiones distintas sin verse obligados unos por otros.

Por otro lado, en EE.UU., en materia de patentes, no existe el requisito de tecnicidad europeo, por lo que es tradición conceder patente a cualquier novedad que pueda ser útil. La oficina de patentes de EE.UU., al principio, no era partidaria de conceder patentes de software argumentando su falta de personal cualificado para examinarlas. Esa decisión fue cambiando con el tiempo y con

la presión de algunos de los grandes clientes de dicha oficina (IBM ha sido durante mucho tiempo su mejor cliente si se estudia en función del número de patentes concedidas).

Pero las patentes en cualquier materia no sirven para mucho si sus propietarios no pueden ejecutarlas y penar a los infractores de los monopolios de 20 años que conceden las mismas. Entonces, a falta de una legislación específica sobre la patente en software, los tribunales de EE.UU. fueron, principalmente durante los años 80, creando jurisprudencia a favor de las patentes de software. Así, dos sentencias clave son cuando el Tribunal Supremo en 1981 decide ratificar la patente a un proceso informático con aplicación práctica y en 1998 cuando decide que "es patentable todo lo que haga el hombre (*sic*) bajo el sol".

Sin embargo, no ha sido hasta el estallido del "punto COM", cuando las patentes de software han jugado un papel importante en la economía del software, al menos si ese papel lo juzgamos: en función del importe de las indemnizaciones por infracción que se están manejando en los juicios estadounidenses, con cuantías de cientos de millones de dólares; en función del número de solicitudes de patentes de software; y, en función de la

cantidad de juicios, demandas y avisos de infracción que se están produciendo. En un reciente estudio, el profesor Brian Kahin de la Universidad de Michigan estimaba en 100 los juicios anuales en EE.UU., en 2.500 las demandas y extrapolaba con el correspondiente factor 25 en 5 millones las cartas de aviso de infracción y exigencia de pago de licencia que no llegaban a demanda.

Actualmente la empresa con más patentes de software, tanto en EE.UU. como en Europa, es IBM y la empresa con más solicitudes es Microsoft.

## 1.2.2. Unión Europea

En la Europa continental se cuenta con una legislación y un concepto de jurisprudencia distintos a los norteamericanos. Las patentes en general se regulan en casi toda Europa por el Convenio de la Patente Europea (CPE) del 1973<sup>3</sup>.

En él se cita explícitamente en el artículo 52.a los programas de ordenador como un sujeto no patentable, aunque el punto 52.3 ha incitado a interpretaciones muy imaginativas a la OEP cuando menciona que los programas de ordenador quedan excluidos como tales.

La intención del legislador con ese punto 52.3 era expresar que aunque el software no fuera patentable, si lo pudieran ser invenciones que integraran software, siempre que dicho software no fuera el objeto de la patente (software como tal o software en sí mismo). Desgraciadamente la forma de expresar esto con un sencillo *as such* (como tal) no fue tajantemente categórica, y de aquí (así como, principalmente, de una incorrecta constitución de la Oficina Europea de Patentes) derivan la mayoría de los proble-

mas sobre patentes de software existentes en la Unión Europea.

La OEP, creada totalmente autónoma bajo el único amparo legal del CPE y ningún sistema judicial por encima, en un principio interpretó esos dos artículos según el ánimo del legislador, tal y como se puede comprobar en sus propias guías de examen del 1978 donde describe un programa de ordenador como: "*Un programa de ordenador puede tener varias formas, por ejemplo, un algoritmo, un diagrama de flujo, una serie de instrucciones codificadas, que pueden ser grabadas sobre cinta u otro medio de grabación legible por máquina, y puede ser considerado como un caso particular de un método matemático, o una representación de la información*".

Esta descripción de un programa de ordenador excluía absolutamente al software (inmaterial) de la patentabilidad mientras que possibilitaba patentar invenciones 'tradicionales' (materiales) que contaran con software en su interior siempre que no fuera el software en sí mismo (*as such*) el objeto de la patente.

Sin embargo, la OEP ha ido, de mutuo acuerdo con sus socios de "La Trilateral" (oficinas de EE.UU. y Japón), cambiando paulatinamente su propia interpretación de su carta fundacional y magna, y plegándose así a las demandas de sus principales clientes: grandes multinacionales de tecnología y software de, según orden decreciente, EE.UU., Japón y Europa.

La doctrina actual de la OEP consiste en considerar que "programas de ordenador" "como tales" son exclusivamente el código

fuelle y el código objeto cuando no se ejecutan en una computadora, y que toda idea que se pueda implementar en una computadora mediante software (programas de ordenador), sea de software, algoritmos, métodos de negocio, funcionalidades del software o meros procesos funcionales, puede ser patentada porque presenta la tecnicidad de el uso de una computadora (obsérvese, por ejemplo, que esa tecnicidad es equivalente a la que se encontraría en los pensamientos humanos o en la escritura en un papel). Esta doctrina, en la práctica, lleva al nihilismo y a la estulticia al artículo 52.2 del CPE. De ahí que se lo salte tan frecuentemente.

Eso se refleja en su guía de examen del 2001 cuando dice que será patentable "*un efecto técnico adicional que va más allá de las interacciones normales entre el programa y el ordenador*", dejando sin definir "adicional" a qué, qué es "técnico", cuánto o qué es el "más allá" y qué es "normal". También es notoria la guía de examen del 2001 donde, importado de la Conferencia de la Trilateral del 2000, introduce el tan manido eufemismo "*computer implemented invention*" (invención implementada mediante ordenador) para poder patentar con él cualquier idea que se lleve a cabo informáticamente, sea o no técnica (como los métodos de negocio). Obviamente, lo único que puede ser implementado en una computadora es el software, los programas, por lo que, en definitiva, este término sólo es otra forma eufemística de llamar al software para evitar nombrarlo y que por tanto caiga en la materia excluida.

Sin embargo el problema para la OEP es que las patentes de software que ha concedido, más de 30.000, <<http://gauss.ffii.org>>, son difícilmente ejecutables en las legislaciones nacionales de los países que forman parte de la OEP, donde las leyes, en general, son más cercanas a la original intención del CPE que de la actual doctrina de la OEP.

En general y no sin controversias, los juicios que se están produciendo en la Unión Europea sobre el tema, más bien pocos, están decan-tándose principalmente por una interpretación restrictiva acerca de la patentabilidad del software y más cercana a las posturas contrarias a las patentes de software tomada por el Parlamento Europeo en su Primera Lectura en la finalmente casi unánimemente rechazada directiva de patentes de software.

De aquí vienen los dos intentos serios y frustrados de la OEP de cambiar la legislación vigente: el primero en el 2.000 dirigido a su carta magna, el CPE; y el segundo a partir del 2002 con la rechazada directiva de patentes de software de la Unión Europea que pretendía "armonizar" todas las leyes



Figura 1. Tienda web patentada.



Las patentes de software no aportan mucho a la innovación en las PYMES ni a la sociedad

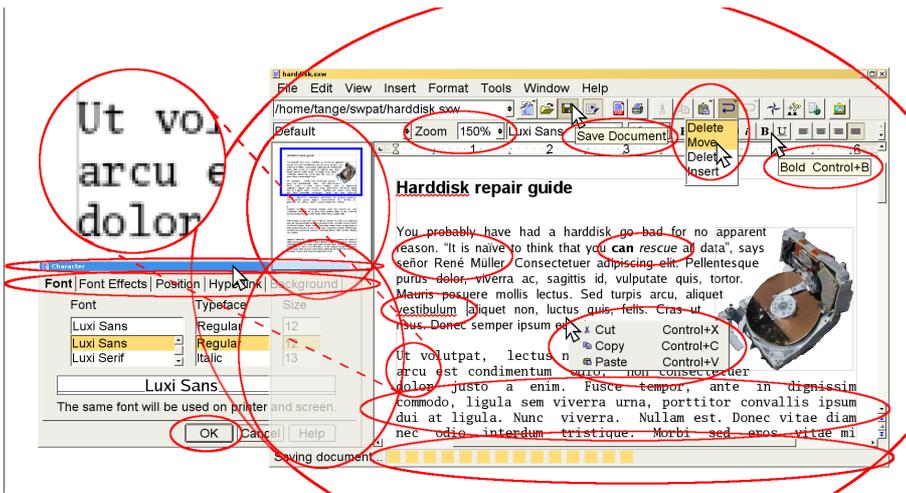


Figura 2. Procesador de texto patentado.

nacionales con la doctrina de la OEP y con la práctica estadounidense (en la práctica casi iguales), y que tanta crispación social generó. El resultado de esta descoordinación entre OEP y las legislaciones nacionales son patentes de software europeas sin reconocimiento legal como el ejemplo sobre una tienda electrónica que se muestra en la **figura 1** (ver [1] en el anexo). O las patentes sobre los procesadores de texto de la **figura 2** (ver [2] en el anexo).

La distribución de patentes de software concedidas por la OEP, por países, es la que se observa en la gráfica de la **figura 3**.

### 1.2.3. España

La ley de patentes española, una de las más restrictivas respecto al software en Europa, viene a decir que será inválida toda reivindicación de patente de invención que "comprenda" (contenga) programas de ordenador (Ley 11/1986)<sup>6</sup> siempre que dicha invalidez no contradiga el CPE (Real Decreto 2424/1986). Por tanto los tribunales españoles están sujetos en última instancia a interpretar que no se contradiga el texto del CPE, pero no así a seguir la malabarística doctrina interpretativa de la OEP.

La ley española de patentes también dice que las patentes deben servir para "elevar el nivel de competitividad de nuestra industria" y "proteger eficazmente los resultados de nuestra investigación". Teniendo en cuenta que España no reúne más del 0,20% de las patentes de software concedidas por la OEP, y que entre EE.UU. y Japón tienen más del 70%, no es muy de extrañar, políticamente, el voto español contrario al texto

de directiva en el polémico y manipulado Consejo de la Unión Europea del 17 mayo del 2004 bajo presidencia irlandesa.

### 1.3. Actualidad

#### 1.3.3. Estados Unidos de América

Las patentes de software, y en menor medida, las patentes en general, se encuentran cada vez más cuestionadas en EE.UU. por cuanto a que estén frenando la competitividad y la innovación en beneficio de unas pocas empresas dominantes que, además, tampoco se vean incentivadas a innovar ni a competir. Es significativo el informe emitido en 2003 por la *US Federal Trade Commission* donde pone en cuestión la idoneidad de la protección por patente como tal. En todo caso, los políticos estadounidenses, en materia de patentes de software, tienen sus ojos más puestos en el otro lado del Atlántico que en sus propios fueros.

#### 1.3.2. Unión Europea

Tras el rotundo rechazo a la Directiva de Patentes de Software por parte del único órgano representativo de la Unión Europea, el Parlamento, los grupos pro-patentes de software vuelven a presionar para intentar legalizarlas mediante la recién "resucitada" Directiva de la Patente Comunitaria.

Esta directiva ha sido hasta ahora realmente conflictiva y nunca ha conseguido poner de acuerdo a los distintos países para su aprobación. El principal punto de desencuentro han sido los idiomas en los que legalmente se podrán realizar los trámites de patentes y más concretamente si las traducciones han de ser vinculantes o meramente informativas.

La forma de legalizar las patentes de software mediante esta directiva se basaría en las siguientes premisas tal y como están incluidas en el actual borrador emitido por el comisario McCreevy de Mercado Interno de la Comisión Europea (el mismo que presentó y defendió la de patentes de software):

- 1) las patentes comunitarias serían concedidas y gestionadas por la Oficina Europea de Patentes;
- 2) la Directiva de Patente Comunitaria incluiría como ley la actual práctica administrativa de la OEP. Esta práctica incluye, en materia de patentes de software, prácticamente una copia de lo que el Parlamento Europeo rechazó y enmendó en dos sucesivas lecturas ya. Por sí misma, esta inclusión podría ser suficiente para legalizar las patentes de software en la Unión;
- 3) se constituiría un tribunal supremo europeo dedicado en exclusiva a patentes que estaría compuesto, gestionado, controlado o muy influenciado por la OEP. Este tribunal permitiría ratificar en última instancia, y mediante jurisprudencia, la legalidad de las patentes de software en la Unión.

Se ha de recordar la no pertenencia a la Unión Europea de la OEP, por lo que el hecho de ceder tal grado de soberanía a esta oficina externa compuesta por funcionarios podría llegar a suponer incluso un incumplimiento de los tratados constitucionales de la propia Unión. También son notables en los órganos judiciales tradicionales las voces de rechazo a que se constituya un tribunal especial que sea dependiente de la OEP y que no esté bajo la tutela del legítimo cuerpo judicial independiente.

Se comprueba que, en caso de cumplirse lo propuesto por esta directiva en su actual redacción, no sólo se legalizarían las patentes de software, sino que también se ratificaría que la OEP, además de ser un órgano administrativo, externo, autónomo, independiente y sin control alguno por encima, ostentaría el poder ejecutivo, el legislativo y el judicial en materia de patentes, y, por ende, el control de la innovación y competitividad tecnológica en la Unión Europea. Probablemente esto la pondría a la altura de todos los ministerios de industria en cuando a poder controlar el tejido productivo y la economía europeas.

Actualmente está por decidirse de qué forma se tramitará esta directiva: si por el procedi-

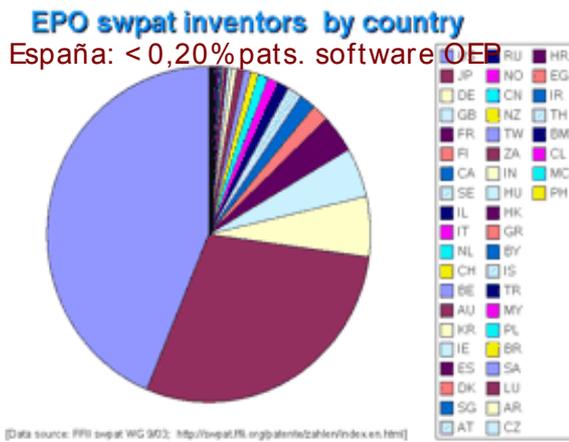


Figura 3. Distribución de patentes de software concedidas por la OEP por países.

miento de codecisión entre el Consejo y el Parlamento Europeo, o si por el hasta ahora fallido ahora procedimiento de unanimidad entre los miembros del Consejo de la Unión.

Dependiendo del camino que elija la Comisión, la tramitación será muy rápida (algunos meses) o tan lenta como la de la difunta Directiva de Patentes de Software (varios años). En todo caso, es de rigor aclarar que la Directiva de Patente Comunitaria no es ni mala ni buena intrínsecamente, sino que, según su desarrollo legislativo, podría significar tanto la legalización definitiva de las patentes de software en la Unión Europea (y en el mundo), como la abolición de las mismas y el encauzamiento de la OEP u oficina alternativa, hacia unos visos democráticos regidos por los legítimos poderes representativos de la Unión.

**2. Economía**

**2.1. PYMES de software**

La mayor parte de la economía e innovación informáticas en Europa está constituida por PYMES, siendo las grandes multinacionales del sector software casi todas norteamericanas (exceptuando prácticamente sólo a SAP AG). El precio de una patente de software europea, contando todos los gastos necesarios, viene a estar entre los 30.000 y los 50.000 Euros. Su tramitación tarda entre unos 3 y 4 años. Un juicio por infracción de patente ampliamente supera casi siempre el millón de Euros por cada una de las partes.

En materia de patentes de software, en EE.UU. donde son legales, se pueden distinguir económicamente cuatro tipos de PYMES: las nuevas empresas, las que han fracasado, las parásitas y las orientadas al cliente. Las que han fracasado son aquellas empresas a las que no les ha funcionado su modelo de negocio. Estas empresas, se puede concluir que no aportan innovación, por lo menos sostenidamente, a una sociedad que ha decidido, con sus reglas de mercado, que no las necesita.

Las empresas parásitos de patentes son empresas de abogados de patentes que se dedican a adquirir carteras de patentes de software. Obviamente, la innovación e incluso la mínima creación de software es algo lejano a su plan de negocio.

Las nuevas empresas difícilmente tienen recursos para dedicarse a registrar patentes y, si lo hacen, seguramente, cuando les sean concedidas y las puedan explotar, ya habrán dejado de ser nuevas empresas y sólo conservarán las patentes si han tenido éxito con su modelo de negocio no basado en patentes.

De todas formas, no ha habido ningún estudio serio que evidencie que las carteras de patentes de software, o más bien de solicitudes de patentes de éstas, hayan propiciado obtenciones de financiación. Generalmente la incertidumbre sobre la valoración de esas carteras es demasiado grande en una nueva empresa y no les aportan mucho valor añadido. Las orientadas al cliente son aquellas empresas informáticas que denominaríamos normales: su modelo de negocio ha tenido éxito y prestan un servicio a sus clientes ya sea en forma de servicios o de productos software (y hardware). Lo que denominaríamos una empresa informática normal y asentada.

Bien, pues las patentes de software pueden ser beneficiosas para los dos tipos de empresas primeras citadas, porque ambas podrían, sin riesgo de infringir patente ajena alguna, hacer valer su cartera de patentes exigiendo licencias a todos aquellos del mercado que las infrinjan o vendiéndoselas al mejor postor. Los parásitos de patentes son especialmente agresivos en exigir pagos importantes a las empresa informáticas productivas. De hecho suelen tener mucho negocio en empresas pequeñas a las que les cuesta más la mera consulta a un abogado de patentes que pagar la licencia exigida.

El problema para la sociedad es que quienes pagarían esas licencias (o extorsiones, según como se vea) serían las empresas realmente productivas e innovadoras: las empresas orientadas a sus clientes e útiles a la sociedad.

Para estas empresas, el posible beneficio que puedan obtener por cualquier patente de software que posean se ve totalmente sobrepasado por el coste de pagar las licencias (o extorsiones) de todas las muchas patentes que con total seguridad infringirá el software que produzcan, aún sin siquiera ser conscientes de que existan las mismas. Por otra parte, el posible beneficio que obtendrían con la supuesta difusión de la innovación (patente significa originalmente "hacer público") de las innovaciones ajenas en el sistema de patentes es mucho más que discutible. ¿Conoce a algún profesional informático que consulte la base de datos de 300.000 patentes de software estadounidenses, la mayor del mundo, a la hora de diseñar sus soluciones software? Más bien ninguno.

Se deduce fácilmente pues, que las patentes de software no aportan mucho a la innovación en las PYMES ni a la sociedad y sí a hacer florecer el negocio de la extorsión contra los verdaderos productores e innovadores en software.

**2.2. Grandes multinacionales software**

Las grandes multinacionales de software y de tecnología (casi todas utilizan software), cuentan con un recurso que no poseen las PYMES: la posibilidad de llegar a los frecuentes macro-acuerdos de intercambio de bolsas de patentes. Esta posibilidad, obviamente, es exclusiva del selecto club de las grandes: no es muy factible que IBM estuviera muy interesado en intercambiar sus miles de patentes de software a cambio de las pocas de una PYME.

En este grupo de empresas se podrían distinguir dos tipos: aquellas que tienen un monopolio de facto en un nicho más o menos amplio de mercado y aquellas que no.

Para las que no tienen muy amplias cuotas de mercado en sus nichos, lo que ganan con sus patentes y lo que pierden por sus infracciones de patentes no licenciadas podría llegar a quedar compensado, aunque siempre cuentan con la incertidumbre de que su principal activo, el software, infrinja patentes imposibles de eludir y por las que tengan que pagar cientos o decenas de millones de dólares no presupuestados: casos Kodak vs. Sun, e-Plus vs. Ariba, NTP vs. RIM (BlackBerry), etc.

Para las que cuentan con monopolios o cuotas de mercado muy altas en sus nichos (Microsoft, Oracle, SAP, Cisco, Adobe, etc.), las patentes de software pueden serles una

forma de perpetuar su monopolio impidiendo que su competencia tenga acceso a recursos del software que en algunos casos son incluso imprescindibles. Muestra de esto son las amenazas públicas de Microsoft a Linux por infracción de sus patentes de software o las exclusiones del código abierto en sus licenciamientos. Frente a este beneficio de perpetuar o incluso incrementar su cuota de mercado ya monopolísticas, el pagar varios cientos de millones de vez en cuando es un riesgo perfectamente asumible: casos Eolas vs. Microsoft, e-Plus vs. SAP, etc. Por otra parte, para ninguna de las grandes empresas suponen un coste significativo las tarifas de obtención y litigación de patentes de software, por lo que para todas ellas es una tremenda ventaja competitiva con respecto a sus PYMES de software competidoras.

## 2.3. Soluciones software de empresas no informáticas

Este tipo de software también estaría obligado a pagar licencias o a no infringir las patentes de software existentes y venideras. Estas empresas se encuentran ante una grave problemática, porque es muy común que sus desarrollos sean conocidos por muchos desarrolladores que no son de su propia plantilla o que algún día dejarán de serlo (20 años dan para mucho) y que un día pueden estar trabajando en una consultora y otro día en cualquier otra empresa.

Es por ello bastante posible que al final los propietarios de patentes acaben conociendo la existencia de esos desarrollos y acaben pidiendo cuantiosas sumas por las patentes infringidas. También es posible que incluso ex-empleados intenten extorsionarlas amenazando con desvelar infracciones.

Este riesgo es aún mayor para las empresas que prestan servicios masivamente, tipo telefonía o ASP/ISP (*Application Service Provider/Internet Service Provider*), porque las sumas por indemnizar irían en función del número de usuarios finales.

## 2.4. Software libre, Shareware, Freeware y otros modelos de distribución gratuita del software.

### 2.4.1. Pago por copia

Este tipo de productores o de modelos de distribución gratuita de software por Internet están en una grave desventaja competitiva con respecto a otros modelos de distribución. Para éstos es imposible conocer el número de copias de un paquete de software determinado que hay en el mercado. Es por tanto imposible cuantificar las indemnizaciones o el número de licencias por copia que habrían de pagar en el nada raro caso de infringir alguna patente.

### 2.4.2. Interoperabilidad y estándares

Por otra parte, el hecho de la existencia de patentes trae consigo el licenciado de los

estándares usados en la ineludible interoperabilidad del software. Prácticamente todo el software está obligado a interoperar entre sí en mayor o menor medida.

Existen tres políticas que se pueden exigir por ley, o por ejemplo, por un consorcio de estandarización, sobre las formas de licenciar y permitir la interoperabilidad por parte de los propietarios de patentes de software: 1) sin restricciones en cuanto a la licencia u obligaciones, 2) licencias RAND (del inglés "Razonable y no discriminatoria"), y 3) licencias libres, abiertas y gratuitas.

El primer tipo permite al propietario de la patente del interfaz licenciarla o no, y por el precio y condiciones que determine para cada cliente. Se concluye que esto es una forma de excluir selectivamente a su competencia o penalizarla severamente. Este licenciamiento es imposible de aplicar a cualquier objeto que se quiera llamar mínimamente estándar. El segundo tipo, RAND, en cierta medida compromete al propietario a licenciarlo por un precio y unas condiciones públicas, comunes y, en teoría, justas. Sin embargo, este segundo tipo es, contra lo que indica su propio nombre (otro eufemismo), discriminatorio, pues puede fácilmente impedir que modelos de distribución libres o que publican el código y por tanto desvelan los detalles de la interoperación, puedan interoperar con el interfaz patentado (caso Comisión Europea contra Microsoft).

Para algunos estándares internacionales se usan licencias de este tipo, no sin generar un amplio rechazo en las comunidades discriminadas: generalmente software libre, código abierto y, a veces, los restantes modelos de distribución gratuita de software. El consorcio W3C (*World Wide Web Consortium*), encargado de estandarizar la *web*, tuvo que dar marcha atrás a su decisión de utilizar este modelo de licenciamiento en sus estándares para no discriminar con él al software libre ni al gratuito. Es significativo que durante la tramitación de la rechazada directiva de patentes de software, hubiera un amplio debate, sobre todo entre los partidarios de la patentabilidad del software, acerca de permitir la excepción de interoperabilidad en las infracciones.

El tercer caso es la excepción de interoperabilidad conducida hasta anular el monopolio de la patente en el caso de que se trate de estándares. Esto es algo obligado por algunos consorcios de estandarización como el citado W3C, OASIS en OpenDocument o incluso por leyes. Este modelo es el único que permite, en caso de que en el mercado existan legalmente las patentes de software, la supervivencia de los estándares de derecho libres y abiertos, esto es, los verdaderamente no discriminatorios hacia ningún tipo de software ni colectivo.

## 2.4.3. Código abierto y desarrollo altruista

Sobre el software de código abierto, ya sea libre o propietario, planea un riesgo mayor de ser encausados por infracción de patente. A los poseedores de las patentes les es mucho más fácil comprobar o encontrar los desarrollos que están infringiendo sus patentes pues tienen todo el código fuente a su disposición. El otro problema endógeno del software libre con las patentes es que una parte significativa de su desarrollo está hecho por programadores voluntarios que no pueden o no están dispuestos a pagar miles de Euros para obtener patentes. En última instancia, su modelo de trabajo está basado en la compartición y colaboración, y, por tanto, en absoluta contradicción con tener patentes para impedir a otros usar su software. Es por ello por lo que se considera que los proyectos de software libre están en desventaja competitiva porque no pueden patentar sus innovaciones, pero, sin embargo, sí tienen que cumplir con todas las patentes ajenas.

Por contra ha habido algunas grandes multinacionales que han liberado partes de sus carteras de patentes para que sean libremente usadas por el software libre en su totalidad o, a veces, por un subconjunto muy restringido (IBM, Sun, CA, Nokia, etc.).

A finales del 2005, algunas de estas empresas favorables a las patentes de software pero interesadas en el software libre, constituyeron una empresa para agrupar sus patentes licenciadas al software libre, principalmente Linux. No se espera sin embargo que esto sea la solución, ni que merme mucho la amenaza existente en los mercados donde las patentes de software son legales, pues cualquier proyecto podrá seguir siendo llevado a juicio por infringir el resto de patentes que no puede licenciar.

Así, no es nada raro por otro lado que proyectos de software libre reciban cartas de propietarios de patentes que imponen paros a sus desarrollos. Son conocidas las amenazas recibidas por los proyectos de reproducción multimedia Videolan (VLC) y mPlayer, ambos radicados en Europa. Pero en definitiva quienes más tienen que temer económicamente a esos pleitos, en los mercados donde son legales las patentes de software, son las grandes entidades usuarias de software de código abierto.

## 3. La esencia del software

La diferencia primordial entre el software y las materias patentables es su naturaleza intelectual o inmaterial. Las patentes siempre han sido aplicadas a las implementaciones físicas de las ideas según son descritas en los formularios de patentes. De esa forma se divulga el conocimiento de esa innovación a cambio de un monopolio tem-

poral. Lo que se patenta en estos casos no es la descripción, sino la combinación concreta de componentes físicos que constituyen ese aparato según figura en la descripción y en las reivindicaciones de la patente. Esa combinación de aparatos, en el caso del software siempre es lo mismo: una computadora o varias computadoras. El problema de fondo con patentar software es que estamos tratando algo inmaterial e intelectual cuya implementación coincide con la descripción si esta descripción es suficientemente precisa. ¿Qué mejor y más precisa descripción hay de una idea sobre software que escribir su código fuente? Por tanto, al patentar la descripción, aunque sea imprecisa o de muy alto nivel, lo que estamos patentando y monopolizando es la propia descripción de la idea, esto es la idea misma, y no la implementación de la misma como le es exigible a cualquier patente.

De esta contradicción viene que se lleguen a incongruencias como que las patentes de software, en la práctica, puedan usurpar la propiedad, o cuando menos ser contradictorias con la protección del software mediante el derecho de autor. ¿A quién pertenece un desarrollo de software, al autor del mismo, o a los propietarios de las patentes de software que infringe?

Aparte de esto, existen dislates como que para desarrollar software de forma segura para no ser demandado por infracciones, sea condición sine-qua-non consultar toda la base de patentes de software y solicitudes existente para asegurarse de no infringir ninguna.

Prácticamente cualquier programa de computadora actual mínimamente útil infringe patentes de software, y generalmente muchas. De hecho, es aún más grave que nunca quede la certeza de saber si se incumplen o no muchas de las patentes concedidas debido al lenguaje críptico y enrevesado interesadamente utilizado en la jerga 'patentil' del software. A esto hay que añadir los casos en que no hay camino alternativo posible para desarrollar algo.

Otra incongruencia más palpable incluso es la duración de 20 años de la patente que viene a corresponder con varias eras informáticas en el calendario de la acelerada evolución del software. ¿Tendremos que esperar 20 años para que de verdad se promocióne y popularice la IPv6 (*Internet Protocol versión 6*) para así no sufrir demandas masivas por infracción de patentes de software en nuestras redes? Curiosa forma de fomentar la innovación.

#### 4. Conclusión

Las patentes de software, aparte de contar con un amplio rechazo económico, democrático y social en la Unión Europea, se

encuentran, como mínimo, en un estado de alegalidad en dicho territorio. Están principalmente originadas como un subproducto derivado de la incorrecta constitución de la Oficina Europea de Patentes, órgano supranacional y en la práctica independiente de cualquier control democrático, que cuenta, en materia de patentes y al menos parcialmente, con los tres poderes: ejecutivo, legislativo y judicial.

El principal problema es que las patentes de software, llevadas a la práctica, dejan el mercado de la creación e innovación del software en manos de unas pocas grandes multinacionales, la mayor parte de las cuales no europeas, sin por contra beneficiar a dicha innovación en general.

En regiones como Europa, donde el mercado de creación de software está casi totalmente dominado por PYMES y además está retrasado en general con respecto a EE.UU., esto plantea un serio problema de competitividad y de creación o incluso mantenimiento de tejido industrial local.

Así, los movimientos de la OEP en favor de armonizar con EE.UU. (y Japón) la legislación de patentes referente al software y materias abstractas o intelectuales, serían un movimiento para favorecer a sus mejores clientes, pero que perjudica seriamente al resto de la sociedad a la que supuestamente debe servir dicha oficina.

Es por tanto necesario, para preservar la salud de la innovación y los innovadores en Europa, proceder a una severa reforma de la OEP de forma que quede bajo el amparo de los poderes políticos democráticos de la Unión Europea y no vuelva a producir incertidumbre económica en el mercado europeo del software.

Y por último, el mayor peligro: si se legalizaran las patentes de software, con las primeras patentes sobre materias intelectuales se habría abierto una peligrosísima brecha legislativa. ¿Qué áreas del saber humano se monopolizarían después? ¿Las ciencias, las artes, la genética humana, el derecho, la religión, ...?

#### Anexo: Patentes referenciadas

- [1] Tienda en Internet
- 1) EP803105 y EP738446
- 2) EP1090494
- 3) EP807891 y EP784279
- 4) EP689133
- 5) EP537100
- 6) EP933892
- 7) EP633694
- 8) EP287578
- 9) EP820620 y EP779587
- 10) EP927945
- 11) EP715740
- 12) EP798657
- 13) EP986016

- 14) EP217308
- 15) EP673135
- 16) EP670652
- 17) EP266049
- 18) EP628919
- 19) EP370847
- 20) EP195098

[2] Procesador de textos  
EP214547 & EP772144 & EP693740; EP665489; EP767419 & EP349459; EP689133; EP349459; EP348031; EP743605; EP537100; EP621528; EP249293; EP449322; EP573262; EP240909; EP281742; EP394160; EP205082; EP262804; EP660248 y relacionada (Patente de Nueva Zelanda 525484 por Microsoft: "Broad XML/Word-processing").

#### Bibliografía

La documentación sobre las patentes de software es escasa en formato tradicional en libro editado, pero muy amplia en formato digital en Internet.  
**FFII, CCIA y miembros del PPE**, 2005, *Software Patents Conference (European Parliament)*, <<http://swpat.ffii.de/events/2005/bx10601/>>.  
**Kahin, Brian**, 1990-, *Brian Kahin: Selected Publications*, <<http://www.si.umich.edu/~kahin/kahinpubs.html>>.  
**Smets, Jean-Paul**, 2000, correo electrónico *Important documents on software patents – Economy and Law*, <[http://europa.eu.int/comm/internal\\_market/en/indprop/comp/snets.pdf](http://europa.eu.int/comm/internal_market/en/indprop/comp/snets.pdf)>.  
**US Federal Trade Commission**, 2003, *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy*, <<http://www.ftc.gov/os/2003/10/innovationrpt.pdf>>.  
**OMPI, 1886-1979**, *Convenio de Berna para la Protección de Obras Literarias y Artísticas*, <[http://www.wipo.int/treaties/es/ip/berne/trtdocs\\_wo001.html](http://www.wipo.int/treaties/es/ip/berne/trtdocs_wo001.html)>.  
**36 países europeos**, 1973, *European Patent Convention*: <[http://db1.european-patent-office.org/www3/dwld/epc/epc\\_2002\\_v1\\_bm.pdf](http://db1.european-patent-office.org/www3/dwld/epc/epc_2002_v1_bm.pdf)>.  
**FFII, índice de documentos**, varias fechas, *Papers on Software Patents*, <<http://swpat.ffii.org/vreji/papri/index.en.html>>.

#### Notas

- <sup>1</sup> El Convenio de Berna para la Protección de las Obras Literarias y Artísticas, de 9 de septiembre de 1886, con sus sucesivas actualizaciones, está disponible en <[http://www.wipo.int/clea/docs\\_new/es/wo/wo001es.html](http://www.wipo.int/clea/docs_new/es/wo/wo001es.html)>
- <sup>2</sup> Aspectos de los derechos de propiedad intelectual relacionados con el comercio, disponible en <[http://www.wto.org/spanish/docs\\_s/legal\\_s/27-trips\\_01\\_s.htm](http://www.wto.org/spanish/docs_s/legal_s/27-trips_01_s.htm)>.
- <sup>3</sup> Directiva 91/250/CEE del Consejo de 14 de mayo de 1991 sobre la protección jurídica de programas de ordenador, disponible en <<http://europa.eu.int/ISPO/legal/es/propint/software/software.html>>.
- <sup>4</sup> Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las Disposiciones Legales vigentes sobre la materia, disponible en <<http://inicia.es/de/tip/LPI.doc>>.
- <sup>5</sup> Disponible en <[http://www.oepm.es/internet/legisla/dcho\\_eur/22cpe.htm](http://www.oepm.es/internet/legisla/dcho_eur/22cpe.htm)>.
- <sup>6</sup> Ley 11/1986 de 20 de Marzo de 1986, por la que se aprueba la Ley de Patentes de invención y modelos de utilidad, disponible en <<http://www.oepm.es/internet/legisla/patentes/iii1a1pat.htm>>.