



Conocimiento Abierto

Open Knowledge



La **Acreditación Europea de Manejo de Ordenador** (*European Computer Driving Licence -ECDL*) es la certificación internacional europea que otorga el reconocimiento de poseer una formación básica y completa en Informática a **nivel de usuario**.

Es una iniciativa de CEPIS (*Council of European Professional Informatics Societies*), recomendada por la Comisión Europea. En España, la implantación de la ECDL está supervisada y canalizada a través de ATI (**Asociación de Técnicos de Informática**).

[<http://ecdل.ati.es/>](http://ecdل.ati.es/)

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>

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>

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>, <flc@ati.es>

Arquitecturas
Jordi Tubella (DAC-UPC) <jordit@ac.upc.es>
Victor Vilhals Yuferra (Univ. de Zaragoza) <vyuferra@unizar.es>

Auditoría SITIC
Marina Touriño, Manuel Palao (ASIA)
<marinatourino@marinatourino.com>, <manuel@palao.com>

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

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

Isabel Davara Fernández de Marcos (Davara & Davara)
<isdavara@davara.com>

Enseñanza Universitaria de la Informática
Joaquín Ezpeleta Mateo (CPS-UIZAR) <ezpeleta@posta.unizar.es>
Cristóbal Pareja Flores (DSIP-UCM) <cpajef@sisip.ucm.es>

Informática y Filosofía
Josep Corco (UIC) <jcorco@unica.edu>
Esperanza Marcos (ESSET-URJC) <euca@esset.urjc.es>

Informática Gráfica
Roberto Vivo (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software
Javier Dolado Cosin (DLSI-UPV) <dolado@si.ehu.es>
Luis Fernández (PRIS-EL-UEM) <lufern@pris.esi.uem.es>

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

Interacción Persona-Computador
Julio Abascal González (PI-UPV) <julio@si.ehu.es>
Jesús Lorés Vidal (Univ. de Lleida) <jesus@eup.udl.es>

Internet
Alonso Álvarez García (TID) <alonso@ati.es>
Llorenç Pagès Casas (Indra) <lpages@ati.es>

Lengua e Informática
M. del Carmen Ugarte (IBM) <cugarte@ati.es>

Lenguajes Informáticos
Andrés Marín López (Univ. Carlos III) <amarin@it.uc3m.es>
J. Angel Velázquez (ESSET-URJC) <a.velazquez@esset.urjc.es>

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

Lingüística computacional
Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>
Manuel Palomar (Univ. de Alicante) <mpalomar@dlsi.ua.es>

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

Profesión informática
Rafael Fernández Calvo (ATI) <rfoalvo@ati.es>
Miquel Sarries Griñó (Ayto. de Barcelona) <msarries@ati.es>

Redes y servicios telemáticos
Luis Guijarro Coloma (DCOM-UPV) <lguijar@dcom.upv.es>
Josep Solé Pareta (DAC-UPC) <pareta@ac.upc.es>

Seguridad
Javier Areitio (Redes y Sistemas, Bilbao) <jareitio@orion.deusto.es>
Composicion, Edición y Redacción ATI Valencia

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

Software Libre
Jesús M. González Barahona, Pedro de las Heras Quirós
(CSYC-URJC) <jgb.pheras@gsyc.esctet.urjc.es>

Tecnología de Objetos
Jesus Garcia Molina (DIS-UM) <jmolina@correo.um.es>
Gustavo Rossi
(LIFIA-UNLP, Argentina) <gustavo@sol.info.unpl.edu.ar>

Tecnologías para la Educación
Josep Sales Ruffi (ESPIRAL) <jsales@pie.xtec.es>

Tecnologías y Empresa
Pablo Hernández Medrano (Bluemat) <pablohm@bluemat.biz>

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

TIC para la Sanidad
Valentín Masero Vargas (DI-UNEX) <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>

Composición, Edición y Redacción ATI Valencia
Reino de Valencia 23, 46005 Valencia
Tlf./fax. 963330392 <secreval@ati.es>

Administración y Redacción ATI Cataluña
Via Laietana 41, 1º, 08003 Barcelona
Tlf. 934125235; fax. 934127713 <secregen@ati.es>

Redacción ATI Andalucía
Isaac Newton, s/n, Ed. Sadiel, Isla Cartuja 41092 Sevilla
Tlf./fax. 954460779 <secreand@ati.es>

Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza
Tlf./fax. 976235181 <secreara@ati.es>

Redacción ATI Asturias-Cantabria <gp-astucant@ati.es>
Redacción ATI Castilla-La Mancha <gp-clmancha@ati.es>

Redacción ATI Galicia
Recinto Ferial s/n, 36540 Silleda (Pontevedra)
Tlf. 986581413; fax. 986580162 <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>

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 <i>Rafael Fernández Calvo</i>	2
Monografía: Conocimiento abierto / Open Knowledge (En colaboración con Upgrade) Editores invitados: <i>Philippe Aigrain</i> y <i>Jesús M. González Barahona</i>	
Presentación. Propiedad y uso de la información y del conocimiento: ¿privatización o procomún? <i>Philippe Aigrain, Jesús M. González-Barahona</i>	3
La Economía Política del procomún <i>Yochai Benkler</i>	6
El redescubrimiento del procomún <i>David Bollier</i>	10
La lengua en el medio digital: un reto político <i>José Antonio Millán</i>	13
Nota sobre las patentes de software <i>Pierre Haren</i>	16
Sobre la patentabilidad de las invenciones referentes a programas de ordenador <i>Alberto Bercovitz Rodríguez Cano</i>	17
Eligiendo la herramienta legal correcta para proteger el software <i>Roberto Di Cosmo</i>	21
Por favor, ¡pírateen mis canciones! <i>Ignacio Escobar</i>	24
La normativa europea y norteamericana sobre propiedad intelectual en el 2003: protección legal antipiratero y derechos digitales <i>Gwen Hinz</i>	26
'Informática de confianza' y política sobre competencia: temas a debate para profesionales informáticos <i>Ross Anderson</i>	30
Secciones Técnicas	
Lengua e Informática	
El software libre y las lenguas minoritarias: una oportunidad impagable <i>Jordi Mas i Hernández</i>	36
Lenguajes informáticos	
Evaluación parcial de programas y sus aplicaciones <i>Pascual Julián Iranzo</i>	40
COMPAS: un compilador para un lenguaje imperativo con aserciones embebidas <i>Joaquín Ezpeleta Mateo, Pedro Gascón Campos, Natividad Porta Royo</i>	47
Seguridad	
Ocultación de imágenes mediante Esteganografía <i>David Atauri Mezquida, Luis Fernández Sanz, Matías Alcojor, Ignacio Acero</i>	52
La confianza y la seguridad aspectos vitales para los servicios electrónicos <i>José A. Mañas Argemí</i>	58
Sistemas de Tiempo Real	
Sistemas Linux de tiempo real <i>Javier Miqueliez Álamos</i>	63
Referencias autorizadas	69
Sociedad de la Información	
Personal y transferible	
Locos por los ordenadores (II): Ada Byron y Charles Babbage, o la bella y la bestia <i>Rafael Fernández Calvo</i>	75
Asuntos Interiores	
Coordinación editorial / Programación de Novática	76
Normas de publicación para autores / Socios Institucionales	79
Monografía del próximo número: «Ingeniería del Software: estado de un arte»	

En resumen

Rafael Fernández Calvo
Coordinación Editorial de **Novática**

<rfcalvo@ati.es>

Apreciado lector / querida lectora:

Raro es el día en que los medios de comunicación, incluidos los no especializados en TIC, no publican noticias sobre el mundo del software libre (o de código fuente abierto --open source). Las penúltimas nos han descrito cómo numerosas grandes empresas y administraciones públicas están adoptando este tipo de programas (ver por ejemplo nuestra reciente monografía sobre e-Administración); las últimas nos hablan de la demanda judicial de la empresa SCO, propietaria al parecer de los derechos sobre el sistema operativo Unix y recién participada por Microsoft, contra IBM, argumentando que esta última empresa ha usado indebidamente código de dicho sistema para desarrollar sus productos basados en Linux.

Para quien no lo sepa, resulta oportuno recordar que nuestra revista ha venido dedicando una especial atención a este tema desde los tiempos en que era considerado una extravagancia subversiva de estudiantes izquierdistas (ver nuestra monografía de 1997 «Software Libre» en <<http://www.ati.es/novatica/1997/126/indice.html>>, que, aunque hoy parezca mentira, nos causó no pocos problemas con algunas empresas y organizaciones que no comprendían el fenómeno y la consideraron poco menos que como ¡apología de la piratería informática!). Más tarde, en la monografía del último número de 2001, ya de forma conjunta con *Upgrade*, tomábamos nota de que el software libre había recorrido con éxito la primera fase de su camino (ver «Software Libre / Fuente abierta: hacia la madurez» en <http://www.ati.es/novatica/2001/154/nv154_sum.html>).

Ahora esa idea de apertura y de libertad, tan necesaria en mi opinión en un mundo globalizado y hegemonizado por las grandes empresas transnacionales, especialmente en entorno informático, da un paso más y se extiende a la información y al conocimiento. Esta monografía parte por ello de la idea de que el carácter comunal de amplias porciones de ambos recursos --información y conocimiento-- es ya una parte importante de la sociedad de la información, y que tal carácter debería ser preservado y ampliado por su importancia para el futuro de la

El procomún del conocimiento

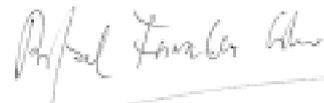
democracia y de las libertades individuales y colectivas. Así pues, derivandolo de *open source*, hemos elegido para la monografía el título «Conocimiento abierto / Open Knowledge» y hemos resucitado el viejo y hermoso término castellano **procomún** para traducir el inglés *commons*, el modelo de gestión colectiva de la información y el conocimiento que se está empezando a articular por la cada vez más amplia comunidad de personas y organizaciones (e incluso empresas) involucradas en el movimiento del software libre.

Esta novedosa monografía, editada de forma conjunta con *Upgrade*, ha tenido como editores invitados a *Philippe Agraïin*, responsable hasta hace pocos meses de software libre en la Dirección General de Sociedad de la Información de la Comisión Europea, y a nuestro compañero *Jesús M. González Barahona*, profesor en la Universidad Rey Juan Carlos de Madrid. A ambos les agradecemos con absoluta sinceridad su valiosa contribución, especialmente a Jesús, pionero y pilar de la comunidad de software libre en nuestro país y artífice, junto a *Pedro de las Heras Quirós*, de las ya citadas monografías de 1997 y 2001, sin olvidar la continua aportación de ambos a la sección técnica de software libre de nuestra revista.

El presente número se completa con artículos de las secciones «Lengua e Informática», «Lenguajes ifnormáticos», «Seguridad» y «Sistemas de Tiempo Real», y con una nueva entrega, en la sección «Personal y transferible», de la historia novelada de grandes figuras de la Informática, sin olvidar las novedades sobre las secciones técnicas de nuestra revista y sobre *Upgrade* que aparecen en «Coordinación editorial».

Espero y deseo que este número sea suficientemente atractivo como para, incluso, formar parte del fondo de lectura para las ya inminentes vacaciones de verano (o pido demasiado :-)

Un cordial saludo,



X CAEPIA-V TTIA' 2003

12/11/2003 - 14/11/2003

Palacio Miramar. Donostia-San Sebastián

<<http://polux.lcc.uma.es/CAEPIA2003/index.html>>

La Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA), y las Jornadas de Transferencia Tecnológica de Inteligencia Artificial (TTIA), son el foro bienal donde la comunidad que trabaja en temas relacionados con la Inteligencia Artificial presenta y discute los trabajos de investigación y desarrollo tecnológico.

ATI es entidad colaboradora de estos eventos

Conocimiento abierto / *Open Knowledge*

Philippe Aigrain¹, Jesús M. González-Barahona²

¹ *Society for Public Information Spaces*; ² *Universidad Rey Juan Carlos, Madrid*

<philippe.aigrain@wanadoo.fr>

<jgb@gsysc.esct.urjc.es>

1. Introducción

En las últimas décadas han emergido dos procesos diferentes y contradictorios en el campo de la creación y difusión de conocimiento e información. Por un lado, las nuevas tecnologías de la información (que actualmente están siendo embebidas en Internet) permiten la transmisión, virtualmente a coste cero, de grandes cantidades de información, y promueven a casi cualquier humano al papel de autor potencial, gracias a la espectacular reducción de los costes relacionados con la publicación. Por otro lado, hay una gran presión para aumentar el control sobre la difusión de conocimiento e información, mucho más allá de la situación actual, usando una mezcla de herramientas técnicas y legales.

Estas dos tendencias ya han colisionado en varios casos, y estos choques están dando forma al futuro. Sin embargo, la mayor parte de las discusiones y debates sobre cómo queremos que sea ese futuro están teniendo lugar lejos de los lugares habituales donde las sociedades deciden lo que quieren: sólo eslógenes y argumentos superficiales llegan a los medios de comunicación y a los foros políticos. Incluso los profesionales de las tecnologías de la información suelen desconocer las implicaciones de las tendencias que están luchando entre bastidores.

Este número de *Novática*, en colaboración con *Upgrade*, trata de mostrar una de estas tendencias (la menos conocida) y de arrojar algo de luz sobre el profundo debate que está en el núcleo de cómo nuestras sociedades tratarán con el conocimiento en los próximos años.

2. Bases y efectos de una revolución

Entre los años 1930 y mediados de los 1950 se crearon las bases científicas y técnicas de una revolución. Los fundamentos teóricos de la lógica simbólica y las invenciones técnicas de las que surgió el ordenador dieron a luz a la tecnología de la información tal y como la entendemos hoy día. En paralelo, la cibernética nos llevó a concebir y diseñar máquinas comuestas por centros de control de información conectados a sensores y actuadores. Esta visión se difundió también a nuestros modelos del mundo natural y pronto modelos y tecnologías basados en la información revolucionaron la genética y contribuyeron al nacimiento de la biotecnología en todas sus dimensiones.

Presentación

Propiedad y uso de la información y del conocimiento: ¿privatización o procomún?

Este artículo se publica bajo las licencias *Public Library of Science Open Access License* y *Creative Commons Attribution License*

Llevó mucho tiempo antes de que nos diéramos cuenta de las diversas implicaciones de esta revolución pues se desarrolló de una forma silenciosa y subterránea. Partes enormes del nuevo conocimiento fueron desarrolladas, por ejemplo, en el dominio público, como la algorítmica y la mayoría del arte de la programación.

Sólo en los años 1980 fue aparente que se habían puesto en marcha dos procesos potentes y contradictorios¹. El primer proceso es la creación de un nuevo ámbito de creación e intercambio de información libre, con costes de transacción extremadamente bajos, y con una enorme multiplicidad y diversidad de colaboradores. La primera revolución de Internet (basada en el correo electrónico, los foros de noticias, ftp, etc.) y el nacimiento del software libre pueden haber movilizado sólo a grupos especializados, pero a pesar de ello fueron paradigmas de un nuevo mundo de la información. Hoy este mundo es también el mundo de la Web, de los datos genómicos o astrofísicos públicos, de la publicación científica abierta, de

Editores invitados

Philippe Aigrain se halla actualmente dedicado a la creación de la *Society for Public Information Spaces*, una empresa de riesgo que se dedicará al desarrollo de herramientas y servicios para el debate público sobre temas técnicos complejos. Desde 1996 hasta abril de 2003 trabajó en la Comisión Europea para los programas de financiación de I+D, siendo responsable de software libre / código abierto, y de la interfaz entre el desarrollo de la sociedad de la información y la regulación de *copyright* y de patentes. Su formación es de carácter matemático e informático. Desde 1983 a 1996 lideró equipos de investigación en el campo del análisis e interacción de medios (video, música y fotografía). Es autor de más de 70 artículos técnicos y de muchos otros sobre la historia, sociología y economía de los intercambios de información.

Jesús M. González-Barahona Jesús M. González Barahona es profesor en la Universidad Rey Juan Carlos, Madrid. Investiga en las áreas de sistemas distribuidos y de computación entre iguales a gran escala. Se interesa por la ingeniería del software libre / código abierto. Comenzó a trabajar en la promoción del software libre en 1991. Actualmente colabora en varios proyectos de software libre (entre ellos Debian), colabora con asociaciones como Hispalinux y EuroLinux, escribe en varios medios sobre temas relacionados con software libre, y asesora a empresas en sus estrategias relacionadas con estos temas. Es socio de ATI y coordinador de la sección técnica de Software Libre de *Novática*.

las enciclopedias libres, de los nuevos medios de comunicación cooperativos, etc. Sin embargo, en paralelo, el segundo proceso vio cómo se reconstruían enormes industrias (farmacéuticas, agroalimentarias, medios de comunicación concentrados) o cómo nacían otras nuevas (software propietario empaquetado). Estas industrias han llegado a ser muy dependientes de su capacidad para conseguir la propiedad o el control del uso de la información y el conocimiento. Esto les ha llevado a pedir, y obtener, una gran extensión del ámbito, duración, intensidad, mecanismos y puesta en vigor de la propiedad de artefactos que consisten completamente, o en gran parte, de información. El sistema de propiedad intelectual se ha convertido en una enorme máquina, en gran parte fuera de control, y cada vez más agresiva porque no es capaz de mantener las aguas de la información dentro de los muros que trata de levantar.

En medio, muchos (la mayoría) de los agentes económicos están buscando desesperadamente un difícil camino. Para sobrevivir, tienen que luchar haciéndose camino en un juego cuyas reglas han sido concebidas para el beneficio de unos pocos grandes oligopolios, no pudiendo a la vez permitirse quedar apartados de los recursos del conocimiento libre.

3. ¿Qué contiene esta monografía?

La situación hoy día es conflictiva y de gran confusión. Como editores de este número especial hemos decidido presentar la visión del *procomún*² del software y de la información, de los intercambios científicos libres, de la creación abierta. Sabemos que la visión opuesta también merece ser expuesta, pero ya lo ha sido muy a menudo, y tiene todo el acceso necesario a los canales de comunicación para conseguir más atención. Por ello hemos ofrecido el espacio a un conjunto muy variado de colaboradores, unidos por el esfuerzo por entender y promover una infraestructura de información del *procomún*, y convencidos de que puede desarrollarse una economía más humana y próspera sobre estas bases. No es una visión compartida por todos y rogamos a los lectores que lo recuerden cuando estén leyendo. Pero para los que la comparten es la clave de un nuevo desarrollo de las habilidades humanas y de los intercambios sociales, de una revolución similar a la que representó para la humanidad el nacimiento de la lectura y la escritura.

En este contexto, hemos invitado a varios autores a escribir sobre tres temas generales:

- El papel de una infraestructura de información común en la sociedad de la información. «*La economía política del común*», de **Yochai Benkler**, abre esta sección, definiendo la estructura del *procomún* de la información, su sostenibilidad y su importancia para la democracia y la libertad individual. **David Bollier**, en «*El redescubrimiento del procomún*» explica cómo una gran parte de la sociedad de la información actual ya es comunal, jugando un papel vital en la producción económica y cultural. **José Antonio Millán**, en «*La lengua en el medio digital: un reto político*», analiza la situación de la mayoría de los idiomas en el dominio digital y cómo la promoción de una infraestructura disponible públicamente de software relacionado con el idioma serviría a las sociedades que usen esos idiomas.

- Por qué el software debería estar en el ámbito de la legislación sobre derechos de autor y no en el de la legislación sobre patentes. **Pierre Haren** abre esta sección con unos breves apuntes con su opinión sobre las patentes de software, «*Una nota sobre las patentes de software*». Más adelante, **Alberto Bercovitz Rodríguez-Cano** ofrece en «*Sobre la patentabilidad de las invenciones referentes a programas de ordenador*» una transcripción de su intervención en una audiencia organizada en el Parlamento Europeo, y **Roberto Di Cosmo** colabora con «*Eligiendo la herramienta legal correcta para proteger el software*», un artículo que estudia las diferentes herramientas legales de protección del software³.

- Los problemas que introducen las tendencias actuales sobre medidas legales y técnicas para limitar los derechos de los usuarios en la gestión de información. Por su carácter motivador, recomendamos la lectura de «*El derecho a leer*», de **Richard Stallman**, ya publicado en *Novática* hace unos años⁴, y que hoy ilustra, mejor si cabe, los problemas a los que nos referimos. En esta monografía, abrimos este apartado con «*Por favor, ¡pirateen mis canciones!*», de **Ignacio Escolar**, un músico que describe la situación de la industria de la música desde su propio punto de vista. Como complemento incluimos «*La normativa europea y norteamericana sobre propiedad intelectual en 2003: protección legal antipiratería y derechos digitales*», de **Gwen Hinze**, y «*'Computación fiable' y política sobre competencia: temas a debate para profesionales informáticos*», de **Ross Anderson**. Ambos muestran las tendencias actuales en la legislación y las medidas técnicas para tratar con la propiedad intelectual y los derechos de autor en el contexto de los medios electrónicos, y los problemas que presentan.

Esperamos que el lector disfrute leyendo estas colaboraciones tanto como nosotros hemos disfrutado mientras ayudábamos a hacer real este monográfico. Y que no olvide que, tenga la opinión que tenga sobre estos temas, nos estamos jugando, en estos años, nuestro futuro, y el de los que vendrán después.

Notas

¹ Algunos pensadores pioneros como Jacques Ellul, Ivan Illitch, Jacques Robin o René Passet describieron la revolución de la información de finales de los años 1960 y 1970. La relevancia de su análisis no fue ampliamente aceptada, en parte porque su crítica de la tecnología era tan fuerte que pocos vieron que también habría posibles caminos alternativos de desarrollo.

² **Procomún**, sustantivo masculino, derivado de «pro» (provecho) y «común», y que significa «utilidad pública» (DRAE). El editor de *Novática* lo ha utilizado aquí para traducir el término inglés *commons*.

³ Destacamos a este respecto la petición al Parlamento Europeo, disponible en <<http://www.upgrade-cepis.org/issues/2003/3/up4-3Petition.pdf>>, de un grupo de prestigiosos científicos e ingenieros informáticos europeos, relacionada con la propuesta de Directiva sobre patentes de software que se está debatiendo en la actualidad en el Parlamento Europeo. Asimismo EUROLINUX está promoviendo otra petición a la que se puede acceder (y firmar) en <http://petition.eurolinux.org/index_html?LANG=en>. Hasta ahora ha sido firmada por más de 150.000 personas y más de 400 empresas.

⁴ Disponible en español en <<http://www.ati.es/novatica/1997/130/if.html>>. La versión original en inglés, con una actualización de 2002, está disponible en <<http://www.upgrade-cepis.org/issues/2003/3/up4-3Stallman.pdf>>.

Referencias útiles sobre «Conocimiento abierto»

Ofrecemos una *lista no exhaustiva* de recursos sobre temas relacionados con lo que hemos denominado «Conocimiento Abierto», lista que, junto a las referencias incluidas en los artículos que componen la monografía, permitirán que los lectores que lo deseen puedan tener un conocimiento más amplio del asunto objeto de la misma.

Artículos

Philippe Aigrain:

- *Positive Intellectual Rights and Information Exchanges*, <<http://freesoftware.mit.edu/papers/aigrain.pdf>> (en inglés) or <http://www.freescape.eu.org/biblio/article.php3?id_article=133> (en francés).

- *11 Questions on Software Patentability Issues in the US and in Europe*, <<http://cip.umd.edu/Aigrain.htm>>.

- «The Individual and the Collective in Open Information Communities», *16th BLED Electronic Commerce Conference*, <http://www.bledconference.org/> (follow link to updated conference program and search for Aigrain). Accesible pronto en <<http://freesoftware.mit.edu/papers>>.

Yochai Benkler

- «Coase's Penguin, or Linux and the Nature of the Firm», *Yale Law Journal*, 112, Winter 2002-2003. <<http://www.benkler.org/CoasesPenguin.html>>

- *Property, Commons and the First Amendment: Towards a Core Common Infrastructure* (White paper for the Brennan Center for Justice) (March, 2001). <<http://www.law.nyu.edu/benkler/WhitePaper.pdf>>.

James Bessen Eric Maskin. *Sequential Innovation, Patents, and Imitation*, January 2000, MIT (Working Paper). <<http://www.researchoninnovation.org/patent.pdf>>.

James Boyle, «Cruel, Mean or Lavish?: Economic Analysis, Price Discrimination and Digital Intellectual Property», *536 Vanderbilt Law Review* 2007 (2000). <<http://www.vanderbilt.edu/Law/lawreview/vol536/boyle.pdf>>.

Rebecca Eisenberg (sobre patentes de biotecnología): <http://cgi2.www.law.umich.edu/_FacultyBioPage/facultybiopagenew.asp?unique=rse>.

Jesús M. González-Barahona and Carlo Daffara, eds., *Free Software / Open Source: Opportunities for Europe?*, Report of the European Working Group on Libre Software, <<http://eu.conecta.it/paper.pdf>>.

Legislación

· **Directiva 2001/29/CE** del Parlamento Europeo y del Consejo de 22 de mayo de 2001 relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines a los derechos de autor en la sociedad de la información. (Es conocida como EUCD). <http://europa.eu.int/information_society/topics/multi/digital_rights/doc/directive_copyright_es.pdf>.

· **Directiva 91/250/CEE** del Consejo, de 14 de mayo de 1991, relativa a la protección jurídica de programas de ordenador. <<http://europa.eu.int/scadplus/leg/es/lvb/l26027.htm>>.

· **Propuesta de Directiva del Parlamento Europeo y del Consejo** sobre la patentabilidad de las invenciones implementadas en ordenador. <<http://europa.eu.int/eur-lex/pri/es/oj/dat/2002/ce151/ce15120020625es01290131.pdf>>.

· **United States' Digital Millennium Copyright Act (DMCA)**. <<http://www.loc.gov/copyright/legislation/dmca.pdf>>.

· **Tratado de la Organización Mundial de la Propiedad Intelectual (IMPO -- WIPO en inglés)**, sobre Derechos de Autor, de 23 de diciembre de 1996. <<http://www.wipo.int/clea/docs/es/wo/wo033es.htm>>.

Libros

David Bollier. *Silent Theft: The Private Plundering of our Common Wealth*. Routledge, March 2002. ISBN:0415932645.

Manuel Castells. *The Rise of the Network Society*. Blackwell Publishers, Nov. 1996. ASIN:1557866171.

Lawrence Lessig. *The Future of Ideas: The Fate of the Commons in a Connected World*. Random House, 1st edition, Oct. 2001. ISBN:0375505784.

Lawrence Lessig. *Code and Other Laws of Cyberspace*. Basic Books, June 2000. ISBN:0465039138.

Jacques Ribin. *Changer d'ère*, Seuil, 1989.

Licencias y tendencias sobre propiedad intelectual

- Campaign for Digital Rights (UK), <<http://ukcdr.org/>>.
- Creative Commons: <<http://www.creativecommons.org/>>.
- Electronic Frontier Foundation: <<http://eff.org/>>.
- Public Library of Science: <<http://www.publiblibraryofscience.org/>>.

Varios

· Posición de ATI acerca de las patentes sobre las invenciones implementadas en ordenador: <<http://www.ati.es/DOCS/documentos/cepis-patentes-042003-esp.html>>.

· Recursos de la FFII (Foundation for a Free Information Infrastructure) sobre patentabilidad del software en Europa: <<http://www.ffii.org/index.en.html>>.

· Recursos de Jamie Love sobre propiedad intelectual: <<http://www.cptech.org/ip/>>.

Yochai Benkler

Facultad de Derecho, Universidad de Yale (EE.UU.)

<BENKLERY@juris.law.nyu.edu>

Traducción: Agustín Palomar (Grupo de Lengua e Informática de ATI)

Resumen: *este artículo define las características institucionales y normativas del procomún, y explica por qué son sostenibles bajo muchas circunstancias. Explica por qué mantener una infraestructura común básica para la producción y el intercambio de información en todo el ámbito informacional es importante tanto para la democracia como para la libertad individual. Se concluye apuntando una serie de acciones políticas prácticas necesarias para construir dicha infraestructura básica común.*

Palabras clave: *autonomía, democracia, características institucionales del procomún, política sobre información, procomún, sostenibilidad económica del procomún.*

1. ¿Por qué el procomún?

El procomún¹ son espacios institucionales en los que podemos practicar un tipo particular de libertad: libertad respecto a las restricciones que aceptamos normalmente como precondiciones necesarias para el funcionamiento de los mercados.

- Aunque a menudo pensamos en los «mercados libres» como espacios que permiten la libre elección, de hecho se trata de relaciones estructuradas que tienen la intención de resaltar un dato particular: la buena voluntad comparativa y la habilidad de los agentes para pagar dinero a cambio de recursos.
- Las restricciones más importantes a la que están sometidas los mercados son aquellas a las que usualmente llamamos propiedad. La propiedad es un conjunto de reglas de fondo que determinan:
 - qué recursos tiene cada uno de nosotros cuando establecemos relaciones con otros;
 - qué nos permite hacer la «posesión» o la «carencia» de un recurso en relación con los recursos implicados en la relación.

Estas reglas imponen restricciones sobre quién puede hacer qué en el dominio de acciones que requieren el acceso a recursos sujetos al derecho de propiedad.

- Si bien es una condición previa necesaria para los mercados, el derecho de propiedad significa que las mismas opciones que existen en los mercados no están libres de restricciones, sino que por el contrario están restringidas dentro de un determinado patrón.
- El procomún son espacios institucionales en los que los agentes humanos pueden actuar libres de las restricciones específicas requeridas por los mercados.

La Economía Política del procomún

© 2003 Yochai Benkler. Este artículo se publica bajo las licencias *Public Library of Science Open Access License* y *Creative Commons Attribution License*.

- Esto no significa que el procomún sea un espacio anárquico. La acción libre pura es ilusoria.
- Significa que los individuos y los grupos pueden usar recursos gobernados por tipos de restricciones diferentes de las impuestas por el derecho de propiedad. Estas restricciones pueden ser sociales, físicas o normativas. Pueden hacer que los individuos sean más o menos libres, en cierto sentido agregado, que lo que permiten las reglas de propiedad. Que el procomún potencie de hecho la libertad o la dañe depende pues de cómo esté estructurado ese procomún y de cómo se hayan estructurado los derechos de propiedad del recurso en la ausencia de procomún.

2. ¿Qué es el procomún?

El procomún es un tipo particular de ordenación institucional para gobernar el uso y la disposición de los recursos. Su característica prominente, que la define en contraposición a la propiedad, es que ninguna persona individual tiene un control exclusivo sobre el uso y la disposición de cualquier recurso particular. En cambio, los recursos gobernados por procomún pueden ser usados por, o estar a disposición de, cualquiera que forme parte de un cierto número de personas (más o menos bien definido), bajo unas reglas que pueden abarcar desde «todo vale» a reglas formales finamente articuladas y cuyo respeto se impone con efectividad.

El procomún puede dividirse en cuatro tipos basándose en dos parámetros:

- El primer parámetro es si está abierto a cualquiera o solamente

Autor

Yochai Benkler is profesor de la Facultad de Derecho de la Universidad de Yale (EE.UU.). Antes de enseñar en Yale fue profesor en la Facultad de Derecho de la Universidad de Nueva York (EE.UU.), donde fue director del Centro Engleberg de Derecho y Políticas de Innovación, y del Instituto de Derecho de la Información. Sus trabajos de investigación se centran en el efecto de la interacción del Derecho, la tecnología y las estructuras económicas sobre la organización de la producción y el intercambio de información, y sobre la distribución del control sobre los flujos de información, conocimiento y cultura en el entorno digital. En particular ha escrito sobre el papel y la sostenibilidad de enfoques no propietarios o basados en el procomún respecto a la producción e intercambio de información a través de las diversas capas del entorno digital, y sobre su papel en términos de democracia y libertad individual.

a un grupo definido. Los océanos, el aire y las redes de autopistas con ejemplos claros de procomún abierto. Diversas ordenaciones tradicionales de pastos o de zonas de riego son ejemplos clásicos, descritos por Eleanor Ostrom, de procomún de acceso limitado, en las que el acceso está limitado sólo a los miembros del pueblo o la asociación que «posee» de forma colectiva algunas tierras de pastoreo o sistemas de regadío definidos. Éstas pueden mejor considerarse expresiones de regímenes de propiedad común, en vez de procomún, ya que se comportan como propiedad de cara a todo el mundo excepto para los miembros del grupo que en conjunto los tiene en común.

- El segundo parámetro es si un sistema de procomún está regulado o desregulado. Prácticamente todos los regímenes de propiedad en procomún limitado estudiados están regulados por reglas más o menos elaboradas --algunas formales, algunas sociales o convencionales-- que gobiernan el uso de los recursos. Por otra parte, el procomún abierto varía mucho según los casos. Algunos tipos de procomún no están regulados en absoluto y se les llama procomún de acceso abierto. Cualquiera puede usar los recursos de este tipo de procomún a voluntad y sin pagar. El aire es de esta clase de recursos con respecto a la toma de aire (respiración, alimentación de una turbina). El aire es, sin embargo, un procomún regulado en lo que se refiere a la expulsión. Para los seres humanos individuales la respiración está ligeramente regulada por convenciones sociales: uno no se pone a respirar con mucha fuerza delante de la cara de otro ser humano a menos que se vea forzado a ello. El aire es un procomún mucho más regulado en la exhalación industrial --bajo la forma de controles de polución. Las áreas de procomún con más éxito y obviamente más reguladas en el panorama actual son las aceras, las calles, las carreteras, y las autopistas que cubren nuestra tierra y forman el fundamento de nuestra capacidad para desplazarnos de un lugar a otro. El recurso más importante que gobernamos como procomún abierto, sin el cual la humanidad no podría concebirse, es todo el conocimiento y la cultura previos al siglo XX, la mayoría del conocimiento científico de la primera mitad del siglo XX, y mucha de la ciencia y el aprendizaje académico contemporáneos.

3. ¿Es sostenible el procomún?

A finales de los años 60 del pasado siglo Garrett Hardin acuñó una metáfora inmensamente efectiva, «*la tragedia del procomún*». Aunque originalmente pretendía explicar por qué los incentivos privados llevarían a las empresas a contaminar su entorno incluso en contra de sus propios intereses a largo plazo y por lo tanto justificarían los controles de polución, la metáfora tomó vida propia. Vino a convertirse en la aseveración de que cualquier tipo de procomún es trágico y de que los derechos de propiedad son una condición previa necesaria para una gestión de recursos eficiente, o incluso sostenible. En los últimos veinte años hemos visto el desarrollo de una literatura que se cuestiona esta visión hasta ahora estándar del procomún. Este esfuerzo ha llegado a cristalizar de forma muy nítida en el trabajo de Eleanor Ostrom. Una revista y un ensayo bibliográfico recientes de Hess y Ostrom proporcionan una descripción general excelente de esta literatura. El quid de casi todo este trabajo es que hay ciertas

circunstancias bajo las cuales los regímenes de propiedad común son sostenibles y posiblemente bastante más eficientes que los regímenes de propiedad individual.

De forma más general, puede decirse que el procomún y la propiedad existen en un espectro de ordenaciones institucionales. Que en este espectro un sistema de gestión de recursos sea más o menos sostenible y eficiente depende de las características tecnológicas del recurso y de los patrones de su uso en cualquier momento histórico dado. Carol Rose identificó muy pronto que los recursos que tienen retornos crecientes en el lado de la demanda, como la externalización de redes, son candidatos especialmente buenos para ser procomún. Ella utilizó este concepto para sugerir por qué las carreteras y los canales, medios clásicos para el comercio, tendían a gravitar hacia modelos de procomún incluso si habían nacido como propiedad privada. Ellickson describió un fenómeno respecto a la tierra de labor, según el cual el tamaño del grupo de propietarios --de uno a muchos-- es una función del uso de la tierra y de los probables fallos a los que está expuesta su gestión. En lo que se refiere a los sistemas de producción y comunicación de información, he explicado como los recursos necesarios para dichos sistemas pueden gestionarse como procomún en modalidades que son sostenibles y deseables.

La información es un bien público en el sentido económico estricto y es también una entrada en su propio proceso productivo. Debido a estas características inusuales, pocos, si es que hay alguno, economistas argumentarían en contra de la proposición de que un área sustancial de procomún en bienes de información no solamente es sostenible, sino que realmente es necesario para sistemas de producción de información eficientes e innovadores.

Más allá de la característica de bien público de la información, el entorno de redes digitales está también permeado por recursos que, si bien no son bienes públicos en el sentido económico estricto, funciona bien sin embargo dentro de un modelo de procomún. Representan casos en los que la compartición de recursos en procomún tiende a reducir la escasez y se comporta mejor que los sistemas basados en propiedad. He escrito en detalle sobre por qué la capacidad de las comunicaciones inalámbricas tiene también esta característica y por qué la creatividad humana a gran escala, tal como la colaboración basada en Internet representada por el software libre y otros proyectos de producción entre iguales (*peer-to-peer*), comparten asimismo esta característica.

El núcleo común de estos dominios diferentes de recursos para la producción y comunicación de información es que hay algún aspecto de un determinado recurso --como la capacidad de comunicaciones inalámbricas, la creatividad humana, la capacidad de proceso distribuido, el almacenamiento distribuido-- que se evidencia en un mercado particularmente árido, caro e ineficiente. En estos casos, las comunicaciones de bajo coste y los procesadores baratos que forman una parte integral de la producción y el intercambio de información hacen que maduren las condiciones para una colaboración sostenible a gran escala y para una compartición de recursos basada en el procomún, en vez de para estructuras institucionales orientadas a la propiedad.

4. ¿Por qué deberíamos preocuparnos?

Hay muchas razones para preocuparse sobre la inclusión de áreas significativas de procomún en nuestro entorno informativo. Hoy se habla sobre todo de preocupaciones sobre políticas de innovación. Como Lessig ha explicado muy bien, las áreas de procomún en los entornos de redes son necesarias para permitir que la innovación progrese sin el permiso de personas interesadas que buscarían restringir la vía de la innovación de forma que ésta se ajustase a sus propios planes de negocio en lo que se refiere al futuro de la tecnología.

Pero el procomún de información, cultura y conocimiento no es sólo, ni es básicamente, una cuestión de innovación. El procomún tiene que ver con la libertad. El procomún son espacios institucionales en los que estamos libres de las restricciones impuestas por los requerimientos de los mercados. Cuando hablamos del entorno de la información, del espacio cultural y simbólico que ocupamos como individuos y ciudadanos, la diversificación de las restricciones bajo las que operamos, incluyendo la creación de espacios relativamente libres de las leyes que estructuran el mercado, va al mismísimo corazón de la libertad y la democracia.

El entorno de los medios de comunicación comerciales ha creado dos efectos de importancia central para la democracia. Uno puede llamarse el *efecto Berlusconi*: el poder político desproporcionado que la propiedad de los medios de comunicación da a sus dueños o a los que pueden pagarlos. El otro puede llamarse el *efecto Baywatch*: el desplazamiento sistemático del discurso público a favor de la distribución de productos de entretenimiento estandarizados. Estos mismos medios han creado también modelos sofisticados de marketing y publicidad que pretenden dar un determinado perfil a lo que cada uno de nosotros ve cuando miramos el mundo a través del cristal de los medios, de tal forma que nuestra mirada, nuestros deseos, nuestras acciones están enfocadas sobre aquellos comportamientos más fácilmente capaces de ser traducidos en consumo

Lo que hace posible el procomún es un entorno en el cual los individuos y los grupos pueden producir información y cultura en su propio interés. Permite el desarrollo de un papel sustancialmente más expansivo tanto para la producción no orientada al mercado como para la producción radicalmente descentralizada. Y estamos viendo organizaciones sin ánimo de lucro que utilizan la Web para proporcionar información o puntos de intercambio cultural con mucho mayor alcance y eficacia que fue posible nunca antes. No menos importancia es el surgimiento de la producción entre iguales (*peer-to-peer*) de información y cultura: fenómeno simbolizado por el software libre, pero que está expandiéndose para incluir noticias y comentarios, como ocurre en foros abiertos (*slashdot*), arte, ciencia, y también con instrumentos de indexación y búsqueda como el Proyecto Directorio Abierto (*Open Directory Project*)². Todos estos fenómenos --el crecimiento de la eficacia y el alcance de actores que están fuera del mercado y el surgimiento de la producción de información radicalmente descentralizada-- proporcionan un contrapunto enormemente importante a la economía industrial de información del siglo XX.

Pero los avances democráticos, la libertad individual y el crecimiento mediante la innovación posibilitados por el surgimiento de la producción sin mercado y descentralizada no emergerán de forma inexorable. Los gigantes industriales que dominaron la producción y el intercambio de información en el siglo XX no renunciarán fácilmente a su dominación. Dado que transitamos hacia una economía de información en red, cada punto de control sobre la producción y el flujo de la información y la cultura se convierte en un punto de conflicto entre el antiguo modelo industrial de producción y los nuevos modelos distribuidos. En la capa física, la propiedad sobre los cables y las licencias inalámbricas que son necesarias para la comunicación proporciona un punto de apoyo para el control. En la capa lógica, los estándares protocolos y software (como sistemas operativos) necesarios proporcionan un punto de control sobre el flujo, y por lo tanto las oportunidades de producción, de información y cultura. En la capa de contenidos, la propiedad intelectual y los modelos de negocio que dependen del estrecho control sobre la información y cultura existente (una entrada central en la nueva creación) amenazan con proporcionar a sus usuarios con la habilidad de controlar quién decide decir que a quién con los signos culturales clave de nuestro tiempo.

5. Una infraestructura básica común

Para conseguir los beneficios de libertad e innovación que la economía de la información en red hace posible, debemos construir una infraestructura básica común junto a la infraestructura propietaria. Tal infraestructura común se extenderá de la misma capa física del entorno de la información a sus capas lógica y de contenido. Debe extenderse de tal forma que cualquier persona tenga algún grupo de recursos de dichas capas que permita a esta persona producir y comunicar información, conocimiento, y cultura a cualquiera. No todos los instrumentos de producción de comunicaciones e información necesitan ser abiertos. Pero debe haber alguna porción de cada capa que cada uno pueda usar sin pedir permiso a nadie más. Esto es necesario para que haya siempre abierta una vía para que cualquier persona o grupo articule, codifique o transmita lo que él, ella, o ellos quieran comunicar, no importa lo marginal o invendible que pueda ser.

Las estrategias principales para construir el núcleo de la infraestructura común son:

- Una capa física abierta que debería construirse mediante la introducción de redes inalámbricas abiertas, o un procomún de espectros.
- Una capa lógica abierta que debería estar disponible mediante una política de preferencia sistemática hacia protocolos y estándares abiertos en lugar de hacia protocolos y estándares cerrados, y de apoyo a plataformas de software libre que ninguna persona o empresa pueda controlar unilateralmente. Más importantes son la reversión o el rechazo a adoptar medidas coactivas que den preferencia a los sistemas propietarios sobre los abiertos. Con esto nos referimos a las patentes sobre plataformas software, y al conjunto de normas que empiezan a surgir sobre mecanismos semejantes al copyright como la Ley de Derecho de Copia Digital del Milenio Digital de los Estados Unidos (*Digital Millennium Copyright Act*)³, que pretende preservar los modelos de negocio industrial de

Hollywood y las industrias discográficas mediante el cierre de la capa lógica de Internet.

- Una capa de contenido abierta. No todo el contenido debe ser abierto, pero los derechos de propiedad intelectual han quedado salvajemente fuera de control durante la pasada década, expandiéndose en alcance y fuerza como nunca antes. Hay una necesidad apremiante de abolir algunas de las reglas que pretenden dar soporte a los modelos de negocio del siglo XX. Estas leyes fueron aprobadas en respuesta al intenso cabildeo (*lobbying*) de las organizaciones interesadas e ignoró el enorme potencial que la producción fuera del mercado y la producción individual descentralizada tiene para convertir en centrales, en lugar de periféricos, los componentes de nuestro entorno de información.

- Reforma de las estructuras organizativas e institucionales que se resisten a los sistemas de producción ampliamente distribuida.

- El primer modelo con éxito a gran escala ha sido el software libre, con sus redes sociales informales ceñidas por el marco institucional formal de las licencias *copyleft* y código abierto
- En la ciencia estamos viendo los primeros signos de los esfuerzos por parte de los científicos para liberar ciencia del viejo modelo industrial de la publicación. La Biblioteca Pública de Ciencia (*Public Library of Science*)⁴ y la Iniciativa de Acceso Abierto de Budapest (*Budapest Open Access Initiative*)⁵ son los primeros esfuerzos primarios en ese sentido. Prometen proporcionar un marco de trabajo en el que los científicos --que ahora hacen el trabajo científico, revisan los artículos y editan las publicaciones más o menos gratuitamente-- puedan gestionar sus propios sistemas de publicación sin tener que depender de los grandes editores comerciales.
- Hablando de publicaciones de forma más general, el surgimiento de proyectos como *Creative Commons*⁶ es un marco de trabajo institucional de enorme importancia.
- En lo que se refiere a las comunicaciones personales informales, los diarios de usuarios de la Web (*weblogs*) están emergiendo como un espacio social interesante para la producción de información libre, independiente y ampliamente distribuida.
- En cada caso varían las características particulares del tipo de información, las barreras institucionales que levantan los organismos interesados y los patrones sociales de uso. En cada caso, las soluciones pueden ser un tantodiferentes. Pero en todos los casos estamos viendo surgir estructuras sociales e institucionales que permiten que individuos y grupos produzcan información libre de las restricciones impuestas por la necesidad de vender información como mercancía en un mercado basado en la propiedad.

Estamos en un momento que ofrecen una gran oportunidad y que suponen un reto a nuestra capacidad para hacer una política que ponga a los seres humanos en el centro de la sociedad de la información en red. Las redes digitales nos ofrecen la oportunidad de mejorar nuestra productividad y crecimiento mientras que simultáneamente se mejora nuestra democracia y se incrementa la libertad individual. Estos beneficios se producen sin embargo a expensas, de los organismos interesados que se han adaptado bien al modelo industrial de producción de información y que están encontrando dificultades para adaptarse a la economía de informa-

ción en red que lo sustituirá. Dichos organismos están presionando y sacando leyes, tecnología y mercados para dar forma al siglo actual a imagen del que ya pasó. Sería trágico que triunfasen.

La construcción de una infraestructura común es una condición previa necesaria para permitirnos la transición desde una sociedad de consumidores pasivos que compran lo que les vende un pequeño número de productores comerciales. Nos permitirá desarrollarnos en una sociedad en la que todos puedan hablar a todos, y en la que cualquiera pueda convertirse en un participante activo en el discurso político, social y cultural.

Notas

¹ **Procomún**, sustantivo masculino, derivado de «pro» (provecho) y «común», y que significa «utilidad pública» (DRAE). Aquí se utiliza para traducir el término inglés *commons*.

² <<http://dmoz.org/>>.

³ <<http://www.loc.gov/copyright/legislation/dmca.pdf>>, en inglés.

⁴ <<http://www.publiclibraryofscience.org/>>, en inglés.

⁵ <<http://www.soros.org/openaccess/>>, en inglés.

⁶ <<http://www.creativecommons.org/>>, en inglés.

Conocimiento abierto / *Open Knowledge*

David Bollier

Periodista, consultor, autor del libro Silent Thief (El robo silencioso)<bollier@essential.org>

Traducción: Alicia Díaz Migoyo

Resumen: *el discurso imperante al hablar de Internet es el del mercado. Pero las categorías económicas son demasiado estrechas de miras para nuestras necesidades como ciudadanos y como seres humanos en el ciberestado al que estamos abocados. Tampoco consiguen entender la cantidad de sitios web, de servidores de listas, de programas de software de código fuente abierto y de sistemas para compartir archivos entre iguales que funcionan como un procomún: sistema abierto y comunal para compartir y gestionar recursos. Resulta que esta producción entre iguales (peer-to-peer) muchas veces es una forma más eficiente y creativa para generar valor que el mercado, además de ser más humanista. El paradigma del procomún (commons) nos ayuda a comprender este hecho porque reconoce que la creación de valor no es una transacción económica esporádica --como mantiene la teoría del mercado-- sino un proceso continuo de vida social y cultura política. ¿Cuándo reconoceremos que el procomún juega un papel vital en la producción económica y cultural de nuestros días?*

Palabras clave: *ciencias económicas, creatividad, economía del don, procomún, teoría de mercado.*

Las categorías intelectuales de la doctrina del libre mercado están tan enraizadas en nuestro conocimiento que muchas veces resulta difícil ver el mundo como realmente es. Es algo que debe tener muy en cuenta quien quiera entender la evolución de Internet, porque muchos aspectos de la cultura digital no se ajustan a los principios económicos neoclásicos. En términos generales, los entornos de red tienden a funcionar más naturalmente como un procomún¹ que como un mercado. Y sin embargo, las categorías de mercado dominan por completo el diálogo público y las políticas que se adoptan, mientras que el procomún sigue siendo un concepto oscuro y mal entendido.

En esta tierra de nadie realmente carecemos de las herramientas conceptuales necesarias para comprender muchos tipos de comportamientos *on-line*. Nuestro discurso económico sólo ve un mercado lleno de consumidores potenciales y no un ciberestado que debería responder a unas necesidades más amplias que tenemos como ciudadanos y como seres humanos.

Uno de los problemas, creo yo, es que no conseguimos reconocer la dinámica que mueve al procomún: un modelo para gestionar recursos basado en la comunidad. Todos pueden acceder al procomún --es un derecho civil más-- y no

El redescubrimiento del procomún

Este artículo se publica bajo la licencia *Creative Commons Attribution-NoDerivs-NonCommercial License*

sólo los que pueden pagárselo. Es un sistema alternativo para fomentar la creatividad, la riqueza y la comunidad, todo a la vez.

El discurso imperante al hablar de Internet es el del mercado. La teoría del mercado da por hecho que los individuos son los principales actores de la vida económica y que esos individuos quieren maximizar sus propios intereses económicos comprando y vendiendo en un 'mercado libre'. Esto se considera la quintaesencia de la 'libertad'. Según la teoría de mercado, el bien público se maximiza al permitir a todos elegir libremente, sin interferencia alguna de los gobiernos. Esas elecciones individuales se consideran libres, mientras que las colectivas (normalmente realizadas por los gobiernos) se consideran coercitivas.

Este discurso es realmente muy estrecho de miras, aunque esté extendido en el mundo desarrollado. No admite que existe una importante dimensión de la sociedad que traspasa los límites de mercado y del estado. Esta dimensión --el procomún-- es una economía informal que, social y moralmente, nos pertenece al 'pueblo'. En la vida política, o en la norteamericana por lo menos, al 'pueblo' se le considera soberano y con más legitimidad que los gobiernos o los mercados. Es este sentido, el procomún *rodea* al mercado y al estado, y actúa como complemento necesario de ambos.

Internet ha potenciado las identidades sociales y los intereses no económicos de la gente, convirtiéndolos en una fuerza con mucha influencia en las redes electrónicas. La creciente

Autor

David Bollier es un estratega, periodista y consultor independiente que se ocupa de una amplia variedad de temas de interés público. Gran parte del trabajo más reciente de Bollier se ha centrado en la defensa del procomún como nuevo paradigma de la política, la economía y la cultura: un tema que ha examinado en su libro "*Silent theft: The Private Plunder of Our Common Wealth*" (Routledge, 2002). Desde 1984, Bollier ha colaborado con el guionista/productor televisivo Norman Lear en numerosos proyectos, es miembro del Norman Lear Center del USC Annenberg Center for Communication. Bollier es también co-fundador de *Public Knowledge*, una organización de defensa del interés público y que representa los derechos del público en temas de propiedad intelectual, tecnología e Internet. Los escritos de Bollier se pueden consultar en <<http://www.bollier.org>>. Vive en Amherst, Massachusetts, EE UU.

popularidad del sistema operativo GNU/Linux y del software de fuente abierta (*open source*) confirman rotundamente el poder del procomún on-line. Hay otros muchos, como los sitios web de colaboración, los servidores de listas por grupos de afinidades, las redes inalámbricas, los archivos on-line para eruditos, y los archivos compartidos entre iguales (*peer-to-peer*). Todas estas modalidades del procomún son nuevas formas de colaboración humana que resultan extraordinariamente productivas.

Pero a la teoría del mercado --tan centrada en el individuo y en lo que se puede medir y vender-- le cuesta aceptar este hecho. No consigue entender cómo unas comunidades estructuradas sobre la confianza, el trabajo voluntario y la colaboración pueden ser más eficientes y flexibles que los mercados convencionales del 'mundo real'. Y es que no consigue valorar en sus justos términos el potencial en creación de valor de la 'producción entre iguales'. Quizá sea porque en el mundo de los negocios se busca el máximo rendimiento en un plazo corto, mientras que esta producción entre iguales es sobre todo un proceso social continuo que gira alrededor de valores compartidos. En los negocios se buscan recursos que sea fácil convertir en bienes de consumo y vender, mientras que el resultado del trabajo en estas relaciones entre iguales tiende a considerarse propiedad inalienable de toda la comunidad.

De hecho, ésa fue la razón principal para crear la Licencia Pública General (*General Public License*, GPL en sus siglas inglesas) para software libre: que las comunidades que desarrollan software puedan seguir controlando su producción colectiva. La GPL permite el acceso libre y por lo tanto fomenta el uso del código del software y la introducción de mejoras en el mismo. Pero también impide --y esto es muy importante-- que alguien 'privatice' el código fuente y quiera convertirse en su propietario para controlarlo. Lo más importante de GNU/Linux es que la GPL permite asegurar que los frutos del procomún se mantendrán en el procomún, otorgándole unas importantes ventajas estructurales sobre el desarrollo de software promovido por empresas.

La teoría económica convencional tiene problemas para entender cómo funciona la 'economía del don' (*gift economy*) del procomún. Es filosóficamente incapaz de explicar cómo puede darse un software creado on-line por un colectivo de voluntarios. ¿O es que la ley de propiedad intelectual no insiste en que la gente no trabaja a menos que su 'propiedad' tenga una fuerte protección legal y que se les remunere económicamente por su trabajo? Pero resulta que aquí tenemos a miles de buenos programadores repartidos por todo el mundo que trabajan gratis, sin el respaldo de aparato empresarial alguno e incluso sin mercado.

Todos estos integrantes del procomún ¿serán excepciones, o incluso aberraciones, de las que las ciencias económicas y los legisladores pueden hacer caso omiso?

Ésta ha sido una tentación en la que llevan décadas cayendo los teóricos de la economía. La estrategia continuamente repetida es agrupar todo lo que no sigue las leyes del mercado y rechazarlo calificándolo de irrelevante.

En la legislación sobre propiedad intelectual, por ejemplo, el dominio público es como una chatarrería donde se acumulan todo tipo de libros, piezas musicales e ilustraciones absolutamente carentes de valor y no protegidas por dicha ley. Las obras valiosas son propiedad del que se ha preocupado de protegerlas, según la opinión más generalizada. El dominio público no pasa de ser «*la estrella oscura en la constelación de la propiedad intelectual*», en palabras del catedrático David Lange.

Igualmente, los economistas consideran la contaminación y las rupturas sociales causadas por el mercado como meras 'externalidades': efectos secundarios que carecen de importancia comparados con el núcleo central de la teoría de mercado, el acto de comprar y vender. La economía de mercado incluso ha construido su propio modelo de comportamiento humano: alaba los comportamientos 'racionales', los que 'maximizan la utilidad' y los que 'buscan el interés personal', pero no valora otros rasgos humanos como la moralidad, las emociones, la identidad social, tachándolos de fuerzas irracionales sin consecuencias.

Hablar del procomún es recuperar importantes aspectos del comportamiento humano, y también de su cultura y su naturaleza, que el discurso de mercado ha desechado. El procomún establece una nueva vara de medir el 'valor'. 'Valor' no es sólo cuestión de precio, es algo que está enraizado en las comunidades y en sus relaciones sociales.

Hablar de procomún es decir que el dinero ya no es el único valor importante: pertenecer a una comunidad con la que se comparten valores morales y objetivos sociales puede ser una potente fuerza creativa por derecho propio. Resulta que la libertad significa algo más que maximizar la utilidad económica propia.

Internet no es el único campo en el que se están desbancando las ficciones del mercado y reconociendo el valor del procomún. Los economistas estudiosos de los comportamientos --largo tiempo frustrados por los frágiles modelos formales de la actividad económica-- están desarrollando nuevos modelos empíricos más rigurosos para describir cómo se comportan los mercados en la vida real.

En vez de dar por sentado, por ejemplo, que todo el mundo tiene cantidades ilimitadas de racionalidad y una información perfecta están documentando cómo se integran en el mercado las emociones y las normas sociales. Los teóricos de la complejidad también están haciendo patentes las serias limitaciones que tienen los modelos económicos rígidos y cuantitativos, y las ficciones teóricas como el 'equilibrio de mercado'. Argumentan que resultaría más convincente examinar los caminos evolutivos propios del desarrollo económico y los principios del cambio auto-organizativo y no lineal.

Estamos asistiendo al surgimiento de una nueva visión mundial y de la economía postmercado. Se está viendo que algunas de las limitaciones inherentes de la ley de la propiedad privada del siglo XVIII y su filosofía económica no resultan adecuadas para el siglo XXI. Lo que todavía no se ha conseguido es articular un nuevo modelo que describa la reintegración de la actividad económica y su contexto social y humano.

El paradigma del procomún, sin embargo, parece resultar bastante prometedor. Ofrece nuevas formas de explicar fenómenos que la economía convencional y los teóricos de la propiedad no saben explicar. El catedrático Yochai Benkler, uno de los principales teóricos sobre los aspectos legales del procomún, ha señalado que la producción entre iguales muchas veces es sencillamente más productiva e innovadora que la basada en la propiedad. Opina que los incentivos del mercado quizá no puedan competir con la producción entre iguales que se puede hacer en pequeñas unidades modulares, para después integrarla en un todo mayor (ejemplos pueden ser Linux, los proyectos compartidos para corrección de pruebas o los mapas de avistamientos de aves).

En la actualidad, la Comisión Federal de Comunicaciones de EE.UU. está estudiando la idea de que un procomún puede ser más eficiente y más equitativo para gestionar el espectro electromagnético que un régimen de asignación de derechos de propiedad. En lugar de que el Gobierno conceda (o subaste) los derechos exclusivos sobre el espectro, la gente podría explotar las nuevas tecnologías para permitir que todos lo compartan, igual que todos comparten la infraestructura de Internet. Además, al permitir que más voces utilicen un recurso público, un modelo de procomún reconocería que el espectro pertenece a todos y no sólo a las compañías que tienen la licencia.

Hay razones poderosas para afirmar que el procomún es un tema económico. Pero no ir más allá es desperdiciar la oportunidad de ampliar los límites del debate. Lo que el procomún nos promete es la posibilidad de volver a integrar lo económico y lo moral, lo individual y lo colectivo, en un marco nuevo y más humanista.

Un reordenamiento conceptual basado en el procomún nos permite hablar de roles, de comportamientos y de relaciones que la teoría del mercado no es capaz de captar adecuadamente. El léxico del procomún va más allá del 'lenguaje del mercado', para el que todos tenemos que ser o productores o consumidores. Y también va más allá del 'lenguaje de la propiedad', para el que todo tiene que ser propiedad de alguna empresa o alguna persona. Nos permite ir más allá de ese pensamiento a corto plazo que sólo quiere aumentar los beneficios y pensar en objetivos más amplios y a más largo plazo que quizá no generen muchos beneficios para los inversores actuales, pero sí son útiles y socialmente constructivos.

En resumen, el procomún *resitúa* lo que entendemos por producción creativa, que pasa de un contexto de mercado a otro más amplio, el de nuestra vida social y nuestra cultura política. En lugar de constreñirnos con la lógica del derecho de propiedad, de los contratos y de las impersonales transacciones de mercado, el procomún inaugura un debate más amplio, más vibrante y más humanista. Se pueden renovar las conexiones entre nuestras vidas sociales y los valores democráticos, por un lado, y por otro entre el rendimiento económico y la innovación. Ganan una nueva legitimidad teórica temas que de otra forma se habrían dejado de lado, como las virtudes de la transparencia, el acceso universal, la diversidad de los participantes, o una cierta equidad social.

Es indudable que el procomún juega un papel vital en la producción económica y social de nuestros días. Cuándo se aceptará plenamente ese papel, o cómo afectará a nuestras futuras actuaciones, es algo que debemos dilucidar.

Nota

¹ **Procomún**, sustantivo masculino, derivado de «pro» (provecho) y «común», y que significa «utilidad pública» (DRAE). Aquí se utiliza para traducir el término inglés *commons*.

Conocimiento abierto / *Open Knowledge*

José Antonio Millán
Lingüista y editor digital

<jam@jamillan.com>

La lengua en el medio digital: un reto político

Este artículo se publica bajo la Licencia 20 minutos

... y entre todos me la hicieron,
habla por habla,
soñando, sueña que sueña,
canta que canta.
Delante la tengo ahora,
toda tan ancha,
delante de mí ofrecida,
sin guardar nada,
onda tras onda rompiendo,
en mí -su playa-,
mar que llevo a todas partes,
mar castellana.

(Pedro Salinas, «Verbo» en «Todo más claro y otros poemas», 1949)

«La cuestión --dijo Humpty Dumpty-- es saber quién es el que manda..., eso es todo».

(Lewis Carroll, «A través del espejo»)

Resumen: el bien colectivo que constituye una lengua se convierte en una mercancía en cuanto pasamos al medio digital. Sin embargo, no hay ninguna razón por la que los hablantes de lenguas que no son el inglés, o los hablantes de lenguas minoritarias, o de variantes minoritarias de lenguas extendidas, no puedan beneficiarse de las muchas promesas de la lengua en el medio digital: interfaces orales, ayudas a la traducción, etc. Este artículo propone medios de política lingüística para que las investigaciones existentes (muchas de ellas financiadas con dinero público) puedan desembocar en programas que sirvan a las necesidades de la sociedad, y no aumenten su dependencia tecnológica.

Palabras clave: interfaces orales, lenguas minoritarias, licencia GPL, política lingüística, traducción.

En una sociedad ya muy mediada digitalmente, hay una tendencia creciente a incorporar el lenguaje natural para la comunicación automatizada entre sistemas y personas, y entre personas de distintas lenguas. Y ello por una razón muy clara: porque la lengua es un sistema de comunicación que la sociedad distribuye a todos, y que todos manejamos habitualmente. La lengua no sólo es la interfaz más común, sino que además es la más fina: no hay menú de elecciones o mapa clicable que pueda aportar todas las posibilidades que da una simple frase. Y si pensamos en la lengua hablada, incluso analfabetos o personas que dudarían ante un ratón o un teclado son capaces de contarle a un sistema automático (bien diseñado) lo que quieren.

Pero al llegar al terreno digital la lengua, un bien común de creación colectiva («entre todos me la hicieron»), gratuito («delante de mí ofrecida») y de uso ilimitado («sin guardar nada») se convierte en una mercancía. Para que las máquinas nos entiendan y nos hablen tienen que servirse de programas que tienen un desarrollo lento y caro, y que exigen conjuntos de datos estructurados (*corpus*, diccionarios) para su desarrollo. Incluso aunque existan estos programas y queramos (y podamos) pagarlos, es muy posible que no tengan en cuenta el conjunto de necesidades de nuestras sociedades. En este artículo voy a exponer los principios que deberían guiar futuras políticas públicas al respecto [1].

Voy a comenzar por una descripción impresionista de un horizonte de uso del software lingüístico, que formulé en otro contexto [2]: «¿Qué tipo de sistemas van a utilizar la lengua como interfaz? De todos los tipos: los de entrada de datos en general (de la agenda personal a sistemas profesionales), los de comercio electrónico (sistemas que busquen productos con determinadas características y vuelvan con descripciones y comparaciones), de ocio (localización de espectáculos, restaurantes, información turística...), educación y formación (sistemas de tutoría y evaluación automáticas), o investigación (localización de materiales, acceso 'inteligente' a bases de datos)».

Usaremos (a veces sin saberlo) estos programas cada vez más. Tendrán capacidades multilingües y serán capaces de emitir hipótesis sobre el interés que tendrá para nosotros un cierto material, traducirlo (con distintos grados de fiabilidad) y resumírnoslo. Serán nuestras herramientas de trabajo intelectual y profesional.

El entorno de estos sistemas configurará un sector económico de importancia. Pero en el caso del español el resultado será que aumentará la dependencia tecnológica de los países hispanohablantes, y se inclinará aún más su balanza de pagos [3]. Sí: los hablantes de muchas lenguas, incluso extendidas,

Autor

José Antonio Millán, lingüista y editor digital, dirigió la edición en CD-ROM del Diccionario de la Real Academia (1995) y creó el Centro Virtual Cervantes en Internet (1997). Ha realizado consultorías y desarrollos para numerosas instituciones, y ha trabajado en política lingüística en el marco de «Los tres espacios lingüísticos», 2001-2002. Es autor de los libros «Internet y el español» (2001) y «De redes y saberes. Cultura y educación en las nuevas tecnologías» (1998).

tendremos que pagar por usarlas en las redes, y los hablantes de lenguas o variantes minoritarias ni siquiera tendrán esa posibilidad: no existirán programas para ellos. Pagaremos por un diccionario informático de sinónimos del español de España o del francés de Francia (aunque sea como parte de un procesador de textos), pero uno del francés de Senegal o del español de Bolivia no lo conseguiremos ni pagando...

¿Por qué habría de ser así, existiendo, como existen para muchas lenguas, investigaciones y recursos --casi siempre financiados con dinero público-- que podrían ser la base del desarrollo de softwares lingüísticos a muy distintos niveles (generales y locales)? ¿Por qué este sector industrial, importante y estratégico, va a estar casi completamente colonizado para lenguas como el español, el francés o el portugués? ¿Por qué va a tardar en existir (si es que llega a hacerlo) para lenguas europeas con menos hablantes?

Para algunas lenguas, la razón puede ser que --por motivos de historia, y de recursos y dotación de sus universidades y otras instituciones-- no ha existido suficiente investigación de base. Pero para el español, para el portugués, para el francés o para el italiano, la verdadera razón es que sus respectivos gobiernos carecen, realmente, de una política lingüística digital. Este es un terreno especialmente resbaladizo, porque en él se juntan dos áreas en las que por lo general (y hablo del caso de España) los gobernantes carecen de conocimientos... y de ganas de solicitarlos: la política lingüística, y la política digital. Respecto a la primera, normalmente no saben ni que existe (salvo en las comunidades autónomas con lengua propia, donde se convierte en un instrumento político), y el alcance social de la cuestión digital tampoco acaba de ser comprendida (ahí está para España una legislación inadecuada, la vergonzosa gestión del dominio .es, la perpetuación de los monopolios de comunicaciones, etc.)...

¿Qué metas debería plantearse una política lingüística digital?:

- Garantizar que los recursos (como *corpus* y programas de desarrollo) y los conjuntos de datos estructurados que alimentan a los sistemas automáticos (como diccionarios) estén disponibles para nuevos desarrollos.
- Aumentar el número de agentes que desarrollan software lingüístico, para que crezca la calidad y la cantidad de opciones.
- Facilitar la incorporación al software lingüístico de las lenguas minoritarias o las variantes locales de lenguas extendidas.

En realidad estos tres puntos se resuelven en uno sólo: abrir al uso los recursos y los datos, mediante una licencia de utilización que garantice que los productos derivados sean igualmente *abiertos* y *reutilizables*. Estos son puntos básicos: en el caso del español (y quizás en otros), existen recursos en instituciones públicas o históricas que luego encuentran extraños problemas cuando se quieren usar para desarrollos. Algunos de recursos presumen de haberse «abierto» a Internet, pero eso sólo significa que les puede consultar: uno puede dar una palabra para que un *corpus* le devuelva sus ocurrencias, o para que se la analice un sistema morfológico. Pero este uso no es apto para desarrollos. *Abrir*, para que quede claro, es entregar un DVD o cualquier otro sistema de almacenamiento, con la totalidad del recurso, a quien lo pida. Al final de este artículo repasaré posibles objeciones a esta

forma de obrar. En cuanto al segundo punto: la *reutilización* se puede asegurar dando el recurso vía una licencia tipo GPL (*General Public License*) [4] o *Creative Commons* [5].

La situación actual (por lo menos en el caso del español) es que los recursos lingüísticos de los centros de investigación públicos no llegan de forma transparente a todas las empresas que podrían utilizarlos, sino a un reducidísimo número, que son los que desarrollan los programas de usuario final. Una política eficaz aspiraría a que los recursos de desarrollo de herramientas lingüísticas en poder de cualquier tipo de instituciones (tanto públicas como privadas) alcanzara a cualquiera que quisiera desarrollar software lingüístico. La visión dominante es que ésta es sólo una tarea de grandes empresas (y por añadidura, norteamericanas), pero la verdad es que tanto a nivel de datos, como de programas de desarrollo, o de pequeños softwares de usuario final, uno puede concebir todo tipo de desarrollos, muchos de ellos a la medida: por ejemplo, diccionarios de especialidad, orales y escritos, que complementarían los léxicos con los que ya contarán conversores habla/texto y sistemas de conversación [6].

Existen distintas vías que podrían lograr este fin. Una de ellas podría ser la creación de un Fondo de Recursos Lingüísticos, gratuito, abierto para cualquier entidad o individuo que quiera hacer desarrollos (bajo licencias como las mencionadas), para garantizar que los resultados del uso de estos recursos fueran igualmente abiertos y reutilizables.

Para constituir este Fondo, lo más realista sería comprar la licencia de uso de los recursos y datos lingüísticos a aquellas instituciones (universidades, o empresas) que las poseen. Puede parecer paradójico que frutos de la investigación pagada con dinero público deban ser comprados de nuevo para que alcancen al bien público, y lo es: pero parece la solución más práctica, comparada con otros caminos posibles... En paralelo, y dado que la titularidad del Fondo sería pública y su destino el bien común, se podría iniciar una campaña para que las instituciones cedieran gratuitamente sus recursos al Fondo (en vez de vendérselos).

¿Qué ámbito deberían tener este tipo de Fondos? Deberían constituirse por lenguas, más que por Estados. Muchas lenguas europeas están distribuidas por un gran número de países, en diferentes continentes (caso del francés, el portugués o el español), y sería absurdo limitarse a la variante europea: un Fondo de la Lengua Española, por ejemplo, debería reunir recursos de cuantas más variantes del español, mejor.

En una sociedad interconectada y multilingüe como la nuestra, podríamos intentar además ampliar el ámbito de los beneficios de una acción como esta, haciéndola extensible a diferentes lenguas. Para ello, cabría incentivar también recursos que saquen partido de la proximidad que exista entre ellas (por ejemplo, las lenguas descendientes del latín: español, francés, catalán, italiano, ...) para generar núcleos de morfología, sintaxis, lexicografía, etc. comunes a todas.

Los modelos de funcionamiento de estos Fondos podrían basarse en los que están vigentes en el *Linguistic Data Consortium* [7] o en la *European Language Resources Association* [8].

En otro lugar me he adelantado a las posibles objeciones que una acción así podría despertar: *«El copiar un corpus o un diccionario morfológico no los merma en nada, en ningún aspecto. Si todos los agentes que quieran hacer desarrollos lingüísticos consiguen libremente los resultados de esta imprescindible investigación de base lo más que puede pasar es que tengamos al poco tiempo una proliferación de programas que reconocen palabras, analizan frases, etc. Muchos de ellos no serán directamente utilizables por los usuarios finales, pero podrán formar parte de sistemas automáticos más elaborados, y el resultado final es que habrá más sistemas que usen nuestra lengua, de más tipos, y más baratos»*[9].

Una propuesta tan simple, barata y con unos beneficios tan claros para la sociedad; una solución que fomenta las capacidades de empresas y grupos de usuarios propios en detrimento sólo de los oligopolios; una propuesta que permitiría controlar un sector estratégico y de tránsito obligado para nuestras instituciones y ciudadanos, debería ser fácilmente asumida por las instancias gubernamentales que están para estas cosas.

La retórica vacua de alabanza de una lengua que nos hemos encontrado ya hecha y extendida podría dar paso a acciones que la defiendan realmente, en vez de perpetuar su situación de colonización tecnológica, con mal servicio a la sociedad que la usa.

Referencias

- [1] Este artículo se beneficia de las discusiones que tuvieron lugar en los encuentros entre expertos lusófonos, francófonos e hispanohablantes en el marco de los Tres espacios lingüísticos (2001/2002), convocados por la Organisation Internationale de la Francophonie, Organización de Estados Iberoamericanos, Comunidades dos Países de Língua Portuguesa, Unión Latina y Secretaría de Cooperación Iberoamericana (<http://www.jamillan.com/trespa.htm>). Quiero agradecer sus aportaciones a Daniel Pimienta e Isabel Trancoso y su constante apoyo a Daniel Prado.
- [2] **José Antonio Millán**, «El español en la sociedad digital: una propuesta», intervención en el II Congreso Internacional de la Lengua Española, Valladolid, 16 a 19 de octubre del 2001 <http://cvc.cervantes.es/obref/congresos/valladolid/mesas_redondas/millan_j.htm>
- [3] **José Antonio Millán**, «La lengua que era un tesoro», 28 de marzo del 2001 <<http://www.jamillan.com/tesoro.htm>> y su versión abreviada en inglés «How much is a language worth. A Quantification of the Digital Industry for the Spanish Language» <<http://www.jamillan.com/worth.htm>>.
- [4] <<http://www.gnu.org/copyleft/gpl.html>>.
- [5] <<http://creativecommons.org/>>.
- [6] Este tipo de acciones, mediante las cuales, por ejemplo, la jerga de la neurobiología en portugués o el léxico de la ingeniería mexicana se incorporara a sistemas preexistentes exige, por una parte que estos sean abiertos y admitan ampliaciones, y en segundo lugar un trabajo colaborativo de creación de un corpus. Éste puede ser tan simple como llamadas telefónicas de voluntarios a un sistema automático que va almacenando muestras.
- [7] <<http://www ldc.upenn.edu/>>.
- [8] <<http://www.icp.grenet.fr/ELRA/home.html>>.
- [9] «El español en la sociedad digital: una propuesta», cit.

Pierre Haren
Ilog

<pharen at ilog.fr>

Traducción: José María Cañas Plaza

Nota sobre las patentes de software

Resumen: el autor explica en este breve artículo sus opiniones sobre las patentes de software.

Palabras clave: derechos de autor, patentes de software, PYMEs.

1. El software es un tipo especial de objeto

El software está más próximo a las matemáticas y los conceptos que a cualquier otro objeto vendible inventado por la humanidad. Las matemáticas no pueden patentarse o venderse. La música está protegida por los derechos de autor (ley de derechos de autor y derechos similares) pero no puede ser patentada.

Conclusión: los derechos de autor para el software deberían protegerse mejor y las copias piratas ser más perseguidas. ¿Por qué producir nuevas leyes cuando las ya existentes no se materializan?

2. Los estudios sobre el impacto económico de las patentes del software en Estados Unidos no encuentran ningún efecto positivo, excepto en las facturas de abogados

¿Por qué las prisas por adoptar un sistema cuyo funcionamiento en Estados Unidos ha demostrado limitaciones, y cuyo único efecto garantizado es la transferencia de los recursos financieros dedicados a la investigación en software hacia gastos de representación legal?

Conclusión: los efectos negativos del sistema se observan a diario en Estados Unidos. ¿Por qué no reconocerlo?

3. La cantidad de patentes triviales concedidas en los Estados Unidos ilustra cuan difícil es establecer la originalidad del software

Además de saturar el sistema legal, favorece a las compañías que pueden permitirse largos pleitos y litigios legales, o aquellas que siendo dueñas de alguna patente la usan para amenazar con represalias a otras compañías acusándolas de violación de esa patente. Si la Oficina Europea de Patentes validara estas decenas de miles de patentes sin un estricto examen sería un desastre.

Conclusión: las patentes del software en Europa crearían una situación nociva, debido al establecimiento anterior de patentes triviales en los Estados Unidos. La eliminación de esas patentes triviales serviría de refuerzo importante de la competitividad y liberaría recursos de la Oficina Europea de Patentes.

4. La situación actual en Europa abre un espacio de libertad, que además es favorable a las pequeñas y medianas empresas europeas del software

La situación actual no perjudica a las multinacionales que venden su software en Estados Unidos y lo protegen allí mediante patentes. Si la autoridad legislativa hiciera patentable en Europa el software estadounidense se daría un regalo gratuito a las compañías americanas de software, cuando muy pocas divisiones de compañías multinacionales han pedido esa patentabilidad en Europa.

Conclusión: ¿por qué tal regalo?

Author

Pierre Haren fundó ILOG, S.A. en Abril de 1987 (<<http://www.ilog.fr>>). Previamente trabajó durante cuatro años en el centro francés de investigación en Ciencias de la Computación, INRIA, como responsable del proyecto sobre el sistema experto SMECI. Anteriormente fue gestor durante tres años de los fondos de investigación del Ministerio francés del Mar. Pierre Haren es titulado de la Ecole Polytechnique (1973) y de la Ecole Nationale des Ponts et Chaussées (1978), y doctorado en ingeniería civil por el MIT (1980). Es miembro ejecutivo de IPSOS, de ENPC, de la Fundación de la Ecole Polytechnique, miembro de la Academia de las Tecnologías y Caballero de la Legión de Honor.

Conocimiento abierto / *Open Knowledge*

Alberto Bercovitz Rodríguez Cano
Catedrático de Derecho Mercantil. Abogado

<albercovitz@wanadoo.es>

1. La propuesta de Directiva Europea sobre la patentabilidad de las invenciones implementadas en ordenador

Existe en estos momentos un importante debate en el ámbito de la Unión Europea para decidir si, además de la protección de los programas de ordenador por medio del derecho de autor que ya estableció la Directiva 91/250/CEE del Consejo, de 14 de mayo de 1991¹, debe permitirse el otorgamiento de patentes para invenciones en las que estén implicados programas de ordenador. Se discute especialmente la propuesta de Directiva del Parlamento Europeo y del Consejo «sobre la patentabilidad de las invenciones implementadas en ordenador» (Bruselas 20.02.2002, COM(2002)92 final)².

La propuesta de Directiva sobre patentabilidad es bastante breve, puesto que incluye once artículos, de los cuales solamente los seis primeros establecen normas de regulación sustantiva. De esas normas cabe destacar lo siguiente:

1. Los Estados miembro han de garantizar «que las invenciones implementadas en ordenador sean patentables a condición de que sean nuevas, supongan una actividad inventiva y sean susceptibles de aplicación industrial» (art. 4.1), y para juzgar si esas invenciones implican una actividad inventiva, las invenciones implementadas en ordenador deben aportar una contribución técnica (art. 4.2).

2. A estos efectos se entiende por «invenciones implementada en ordenador, toda invención para cuya ejecución se requiera la utilización de un ordenador, una red informática u otro aparato programable y que tenga una o más características nuevas prima facie que se realicen total o parcialmente mediante un programa o programas de ordenador». Y como contribución técnica se considera «una contribución al estado de la técnica en un campo tecnológico que no sea evidente para un experto en la materia» (art. 2).

3. Se dispone que «la contribución técnica deberá evaluarse considerando la diferencia entre el estado de la técnica y el ámbito de la reivindicación de la patente considerada en su conjunto, que puede incluir tanto características técnicas como no técnicas» (art. 4.3).

4. Por lo que se refiere a las reivindicaciones de las invenciones implementadas en ordenador, éstas pueden reivindicarse «como producto, es decir, como ordenador programado, red informática programada u otro aparato programado, o como procedimiento realizado por un ordenador, red informática o aparato mediante la ejecución de un programa» (art. 5).

Sobre la patentabilidad de las invenciones referentes a programas de ordenador

Este artículo reproduce sustancialmente la intervención del autor en la audiencia ante el Parlamento Europeo celebrada el 7 de noviembre de 2002 sobre la patentabilidad de las invenciones ejecutadas por medio de ordenador.

5. Y en cuanto a la relación con la Directiva 91/250/CEE, el artículo 6 dispone que «los actos permitidos en el marco de la Directiva 91/250/CEE sobre la protección jurídica de programas de ordenador mediante derechos de autor, y en particular sus preceptos relativos a la descompilación y la interoperabilidad o las disposiciones relativas a la topografía de los productos semiconductores o las marcas comerciales, no se verán afectados por la protección que las patentes otorgan a las invenciones pertenecientes al ámbito de aplicación de la presente Directiva».

2. Cuestiones a tratar

A la vista de la problemática planteada por la propuesta de directiva a que acaba de hacerse referencia, esta breve exposición se va a centrar en torno a tres cuestiones fundamentales: primera, el carácter industrial de las invenciones patentables; segunda, las patentes y el derecho de autor como medios para la protección de los programas de ordenador, y, tercera, el previsible impacto que tendría la nueva directiva que se propone.

3. El carácter industrial de las invenciones patentables

El problema básico que plantea la protección por medio de patentes de las invenciones implementadas por ordenador no es un nuevo sino que ha sido una de las cuestiones fundamentales desde siempre para la aplicación del derecho de patentes; consiste en que las patentes solamente pueden concederse para invenciones industriales.

Ya en mi tesis doctoral sobre «Los requisitos positivos de patentabilidad en el Derecho alemán», publicada en el año 1969, me ocupaba de exponer las conclusiones de la doctrina y jurisprudencia alemanas sobre esta materia.

Autor

Alberto Bercovitz Rodríguez Cano pertenece al despacho de abogados Estudio Jurídico y es Catedrático de Derecho Mercantil (especializado en temas de derechos de autor, marcas y patentes), campo en el que lleva ejerciendo como letrado desde hace 25 años. Se le conoce especialmente por sus intervenciones en litigios sobre patentes y su labor es altamente considerada por sus colegas.

Según esa doctrina estaba claro que no eran patentables todas las invenciones, sino solamente las invenciones industriales, «*las que permiten una utilización industrial*» (*die eine gewerbliche Verwertung gestatten* en el texto de la Ley alemana de 2 de enero de 1968). En la sentencia del Reichsgericht de 21 de enero de 1933 se declaraba ya que «*la invención debe ser industrial según su naturaleza, esto es, que ha de trabajar con los medios de las fuerzas naturales para conseguir un resultado que pertenezca al campo del actuar humano delimitado de esa misma manera*». Por eso se declaraba que no era patentable, por carecer de carácter industrial, todo aquello que se puede considerar como «*pura indicación al espíritu humano*», que no deba traducirse en una actuación sobre las fuerzas de la naturaleza.

El carácter industrial de la invención exigía, por tanto, no sólo que los medios utilizados en su ejecución fueran predominantemente industriales, sino también que el resultado tuviera carácter industrial, es decir que no contuviera una indicación a la actividad intelectual humana. Así se declaraba que no producían resultado industrial y no eran por tanto patentables los planos de situación y las guías, los planos de construcción, las tablas de calcular, las tablas de logaritmos o la sustitución por nuevos símbolos de ciertas indicaciones puestas en un aparato.

Esta exigencia de que la invención patentable tenga carácter industrial se expresaba habitualmente estableciendo que sólo las invenciones técnicas (*Technische Erfindungen*) son patentables, entendiéndose en este contexto el término ‘técnica’ como equivalente a ‘técnica industrial’; puesto que no cabe ignorar que existen otras técnicas distintas aplicables a otros ámbitos de actuación del ser humano, tales como las técnicas de enseñanza, las técnicas de organización o las técnicas comerciales.

En el Convenio Europeo de Patentes de 1973³ se mantiene esta exigencia, al exigir que las invenciones patentables «*sean susceptibles de aplicación industrial*», lo cual equivale en la doctrina tradicional a exigir que las invenciones patentables sean invenciones técnicas, entendida la técnica como técnica industrial. No es correcta la equiparación que hace el artículo 27 del convenio ADPIC⁴, en su nota explicativa, donde se establece que «*susceptibles de aplicación industrial*» es sinónimo de ‘útiles’. Útiles, en el sentido de que satisfacen necesidades humanas, son todas las invenciones, sean o no sean industriales. Y esta distinción entre utilidad y susceptibilidad de aplicación industrial tiene importancia porque en la legislación norteamericana no se contiene expresamente la exigencia de que las invenciones patentables sean industriales, aunque sí se exige que sean útiles. Por ello los planteamientos sobre la patentabilidad aplicados en los Estados Unidos no pueden ser trasladados pura y simplemente al Derecho europeo.

Hay que destacar que la exigencia de que las invenciones patentables sean industriales tiene consecuencias fundamentales en el derecho de patentes. Esa exigencia significa que solamente pueden ser patentables invenciones que se apliquen para la producción de bienes o servicios que inciden sobre objetos materiales destinados al mercado. Ese dato de

que el resultado de la invención pueda ser comercializado con autonomía frente a la empresa que lo produce es fundamental, porque significa que, aunque el titular de la patente tiene un derecho de exclusividad para la explotación de esa patente, tiene que producir los resultados de la misma para comercializarlos y de esa manera el derecho de exclusividad beneficia al titular, pero beneficia también al conjunto de la sociedad, puesto que el titular tiene que tener suficientemente abastecido al mercado. No se olvide que el propio Convenio de la Unión de París para la protección de la propiedad intelectual (1973)⁵ permite el otorgamiento de licencias obligatorias cuando el titular de la patente no suministra adecuadamente al mercado. Y, además, el titular sólo podrá beneficiarse económicamente de la invención si comercializa los productos obtenidos con ella. Por ello el derecho de patentes es inadecuado para proteger, por ejemplo, los métodos de organización o los métodos comerciales, porque éstos pueden explotarse sin necesidad de comercializar ningún producto o servicio concreto obtenido directamente por la invención, de manera que la sociedad en su conjunto no puede beneficiarse de la explotación de la invención. Y por lo demás, basta pensar en la rémora que significaría para el progreso el hecho de que una empresa que inventara un método de comercialización verdaderamente novedoso, como puede ser en estos momentos un sistema informático para confirmar los pedidos de los clientes o para facilitar esos pedidos, o como por ejemplo pudo ser en su momento el de las ventas a plazos o el *leasing*, pudiera impedir nada menos que durante veinte años que otras empresas hicieran lo mismo. Puede ser significativo que ya en el Estatuto sobre Propiedad Industrial español de 1929 se permitiera patentar los procedimientos económico-comerciales, sin que llegara a otorgarse ninguna patente de ese tipo.

Si aplicamos las consideraciones precedentes a la posible patentabilidad de las invenciones implementadas en ordenador se llega claramente a la conclusión de que son patentables las invenciones que incorporen programas de ordenador a máquinas o aparatos que producen resultados industriales, esto es, resultados que tengan una manifestación física susceptible de ser comercializada autónomamente, o que produzcan efectos técnicos en el funcionamiento del ordenador.

Pero lo que hay que destacar es que la exigencia que ya establece el convenio europeo de patentes de que las invenciones sean susceptibles de aplicación industrial es lo mismo que exigir que la invención aporte una contribución técnica. Es necesario establecer esa exigencia adicional de la aportación técnica en el derecho norteamericano, porque no la establece la ley; pero en el convenio europeo de patentes no tiene sentido exigir por un lado la susceptibilidad de aplicación industrial y por otro lado la aportación de una contribución técnica. Es duplicar el mismo requisito. Y por ello si esa exigencia suplementaria se implanta en la futura directiva, creará una distorsión innecesaria en el derecho de patentes europeo en general, pues permitirá interpretar que la aportación de una contribución técnica es sólo una exigencia para las invenciones implementadas en ordenador, pero no para todas las invenciones industriales.

Por otra parte, de la doctrina tradicional antes expuesta resulta también con carácter evidente que no deben poder patentarse

las invenciones puramente comerciales o de organización, aunque utilicen un ordenador.

Cabría decir, en síntesis, que el hecho de que una invención incluya un programa de ordenador, ni hace por sí solo que la invención sea patentable, ni la hace que no sea patentable.

4. Las patentes y el derecho de autor como medios para proteger los programas de ordenador

Como es bien sabido, la protección jurídica de los programas de ordenador, establecida en la Directiva 91/250/CEE, de 14 de mayo de 1991 fue impulsada por quienes pretendían la protección de esos programas, por considerar que ofrecía ventajas importantes frente a la protección por medio de patentes.

Como tales ventajas hay que considerar el hecho de que la protección se obtiene de manera automática a nivel prácticamente universal por el simple hecho de la creación, sin necesidad de registros ni de pago de tasas; la no exigencia de actividad inventiva, la no exigencia de una descripción que deba hacerse pública, y la no obligación de tener abastecido el mercado, aparte de un plazo de protección mucho más largo que el de las patentes.

Pero hay que recordar como aquella directiva hubo de adaptar el derecho de autor, modificando algunas de sus premisas fundamentales, tal como una definición específica de la exigencia de originalidad, la atribución de la titularidad según reglas distintas a las del derecho de autor, y especialmente una prohibición, totalmente ajena al derecho de autor, como son las restricciones a la descompilación.

Obsérvese pues que se aplica un régimen jurídico, el de derecho de autor, porque se considera el más favorable, pero modificándolo expresamente para evitar consecuencias inherentes con carácter general a ese régimen jurídico. Ahora se pretende dar un paso más, se pretende llegar a poder proteger los programas de ordenador por medio de patentes, pero una vez más se trata de conseguir una protección 'a la carta', que altera sustancialmente los principios básicos de esa rama del derecho.

Si se pretende proteger por medio de patentes las invenciones implementadas en ordenador habrá que exigir rigurosamente que tanto los medios como el resultado sean industriales; habrá que exigir que la novedad y la actividad inventiva necesarias impidan que el simple hecho de aplicar un programa de ordenador al alcance de cualquier técnico en la materia permita apropiarse de creaciones que ya están en el dominio público; habrá que exigir que la descripción que debe hacerse pública incluya el código fuente y, por supuesto, deberá permitirse plenamente la investigación sobre el objeto de la patente, y por tanto su descompilación, según se establece con carácter general para las invenciones patentables.

No tiene sentido mantener para las patentes la limitación a la descompilación establecida en la directiva sobre protección jurídica de programas de ordenador; pero si que habrá que permitir el uso de las interfaces.

Y, por supuesto de acuerdo con lo establecido anteriormente,

debe mantenerse la exigencia de que el titular de la patente tenga adecuadamente abastecido al mercado con el resultado de la invención patentada.

Lo que no parece admisible en ningún caso es establecer un régimen jurídico en el que se toman del derecho de autor y del derecho de patentes las partes que se consideran beneficiosas para los titulares de derechos y se excluyen aquellas otras normas que se consideran perjudiciales, pero que son las que garantizan el equilibrio entre el interés de los titulares y el interés de la sociedad.

Por otra parte, la acumulación que puede darse entre la protección por patente y la protección por el derecho de autor debería regularse cuidadosamente.

La patente protegerá no sólo la forma de expresión del programa, a diferencia del derecho de autor, sino las ideas caracterizadoras del mismo y sus equivalentes. Ello significa que cuando se extinga la patente cualquier tercero podrá explotar un programa basado en esas ideas y que se exprese de manera diversa a la del programa del titular de derecho de autor que continua vigente.

Pero la existencia del doble régimen jurídico tiene otra implicación fundamental y es que si una invención patentable no es patentada, no existe ningún derecho de exclusividad sobre las ideas caracterizadoras de la invención. Por tanto, si una invención ejecutada por ordenador no se patenta, siendo patentable, sus ideas caracterizadoras podrán ser investigadas y utilizadas por cualquiera, de manera que el titular del programa de ordenador sólo podría impedir, en base al derecho de autor, que se copie la forma de expresión del programa. Por ello, si prospera la idea que incorpora la propuesta de directiva, debería plantearse la conveniencia de integrar en una única directiva la ya existente sobre protección por derecho de autor y la protección por patente.

5. El previsible impacto que tendría la nueva Directiva que se propone

No cabe duda de que en la situación actual existe el riesgo de que los derechos nacionales, especialmente en su aplicación jurisprudencial, evolucionen de manera divergente en relación con la patentabilidad de las invenciones implementadas en ordenador; igual que existe el riesgo cierto de que la Oficina Europea de Patentes, <<http://www.european-patent-office.org/>>, de manera autónoma y sin control ninguno, extienda la patentabilidad de esas invenciones.

Desde ese punto de vista puede tener sentido llegar a dictar una directiva sobre esa materia, siempre que se complemente adecuadamente de manera que se armonicen también las reglas que ha de aplicar la Oficina Europea de Patentes.

Pero esa directiva debe ser realmente clarificadora lo cual exige:

- a) que la protección por patente respete los principios fundamentales del derecho de patentes;
- b) que se clarifique de manera pormenorizada como se produce la acumulación, en su caso de la protección por patente y por derecho de autor;

c) que se evite de manera clara la tendencia a ampliar la materia patentable a invenciones no industriales por el simple hecho de incorporar un programa informático.

Pero no cabe ignorar que la extensión de la patentabilidad a invenciones implementadas en ordenador suscita en la actualidad otra clase de riesgos.

En efecto, se considera que tanto la Oficina Europea de Patentes (OEP) como la Oficina Europea de Marcas, Dibujos y Modelos (OAMI, <<http://oami.eu.int/>>) deben autofinanciarse, lo cual significa que deben conseguir el mayor número de solicitudes de protección posibles, para lo cual tratan de atender a las pretensiones de lo que denominan su 'clientela', que son en definitiva y fundamentalmente las empresas más importantes. Pero esto tiene consecuencias gravísimas que pueden apreciarse ya perfectamente en la práctica y es que para conseguir esa autofinanciación y satisfacer los requerimientos de su 'clientela' se tiende a extender cada vez más el ámbito de lo patentable y se reduce extraordinariamente la exigencia de nivel inventivo.

Así ocurre que proliferan miles de patentes con un ínfimo nivel inventivo, si es que lo tienen, y ello produce efectos devastadores en el sistema de patentes. En primer término se producen retrasos intolerables de muchos años en la concesión de las patentes y además se cambia sustancialmente la funcionalidad del sistema de patentes, porque se pasa a beneficiar a aquellas empresas con mayor poder económico que por su capacidad financiera pueden solicitar mayor número de patentes, aunque apenas tengan nivel inventivo, y tienen capacidad económica para pleitear prácticamente sin limitaciones.

Es obvio que esta situación, que ya existe, perjudica especial para solicitar cientos de patentes y defenderlas o defenderse en un número grande de procedimientos judiciales. Pero sobre todo se perjudica al propio sistema de patentes que está pensado para promocionar las auténticas invenciones, esto es, las que tienen un nivel inventivo indudable, y no está pensado para beneficiar a aquéllos que tienen mayor potencia económica.

Pues bien, si se amplía lo patentable en el ámbito de las invenciones implementadas en ordenador, las consecuencias previsibles son evidentes a la vista de la situación actual que acaba de describirse: al aumentar el número de solicitudes de patentes aumentarían los retrasos en la concesión de las mismas y, dado el mínimo nivel inventivo que hoy día se está exigiendo, proliferarían las patentes y las PYMES no podrían enfrentarse a la actuación de empresas muy potentes económicamente, que podrían solicitar cientos o miles de patentes e iniciar pleitos indiscriminadamente, muchas veces con efectos puramente intimidatorios. Por tanto, la primera prioridad que debería establecerse en estos momentos sería la de volver a dar al sistema de patentes en su conjunto su verdadera función exigiendo un nivel inventivo alto y limitando claramente la protección a las invenciones industriales.

No se olvide que la concesión de patentes debería hacerse en todo caso con criterio restrictivo, puesto que es evidente que uno de los principios fundamentales en que reposa la Unión Europea es el principio de libertad de empresa.

Notas

- ¹ Disponible en <<http://www.davara.com/documentos/relacionados/propiedad/Dir91250.pdf>>, en español.
- ² Disponible en <http://www.belt.es/legislacion/reciente/pdf/propuesta_direc_patentab_25_jun_02.PDF>, en español.
- ³ Disponible en <http://www.oepm.es/internet/legisla/dcho_eur/22cpe.htm>, en español.
- ⁴ Disponible en <http://www.wto.org/spanish/docs_s/legal_s/27-trips_01_s.htm>, en español.
- ⁵ Disponible en <http://www.oepm.es/internet/legisla/dcho_int/11conpar.htm>, en español.

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 Informática¹.

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 patentes². De hecho, se puede construir un mercado bastante considerable simplemente a través de marcas registradas, como hace la distribución de Linux Red Hat³, y se puede construir un monopolio de sistemas operativos durante décadas simplemente con los derechos de autor, como hizo Microsoft⁴. 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 especificación⁵.

¿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 Amazon⁶.

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 software⁷.

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

- ¹ También invito al lector a leer la significativa opinión del profesor Ullman sobre la materia [4].
- ² 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*».
- ³ 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.
- ⁴ Microsoft ha entrado recientemente en el juego de las patentes de software.
- ⁵ Véase <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnkerb/html/Finalcifs_LicenseAgrmnt_032802.asp>
- ⁶ 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.
- ⁷ 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.

Conocimiento abierto / *Open Knowledge*

Ignacio Escolar
Músico

<nacho@meteosat.org>

Resumen: un músico explica cómo prefiere que las canciones de su grupo se difundan lo más posible, de cualquier forma, que controlar estrictamente que nadie haga copias ilegales de ellas.

Soy un músico con suerte. Mi grupo ha vendido, por los pelos, más de 15.000 copias de su primer LP. En un mundo en el que Enrique Iglesias coloca seis millones de CDs cantando así¹, esta modesta cifra tampoco es para tirar cohetes. Pero si me aplicase tanto como futbolista, jugaría en primera división y, si me dedicase a la medicina con tanto éxito, sería neurocirujano.

Durante un par de semanas del mes de abril de 2000, uno de nuestros *singles* se coló en el número diecisiete de las listas de ventas en España; el número tres, si se contaba únicamente a los artistas nacionales. Cada año salen 32.000 discos nuevos al mercado en todo el mundo y sólo 250 convencen a más de 10.000 compradores. Apenas el 0,7% de los músicos que han presentado disco el año pasado (la gran mayoría no llega siquiera a grabar) es más afortunado que yo.²

Se pensarán que nado en dinero. O que, por lo menos, vivo dignamente de mis habilidades musicales. ¿Cuánto cobra el 0,7% con más suerte de su profesión? No les aburriré con cifras pero, tras tres años de esfuerzos hasta conseguir ver mi LP en las tiendas, sólo he ganado poco más de 3.000 euros por venta de discos y derechos de autor. Apenas 85 euros al mes es lo que me ha rentado mi afortunada carrera musical.

Mi parte alícuota del local de ensayo -la garantía de que mis vecinos no me echarán de casa por ruidoso- me sale por 40 euros al mes. Estas navidades quemé la mitad de mis beneficios en un teclado nuevo³, un capricho. Si tuviera un gerente con facultad para vetar mis presupuestos, seguiría tocando con el *casiotone*⁴ que me regalaron los Reyes Magos en 1986.

No culpo a la piratería de mi bancarrota. No a la de «sexo, drogas y rock and roll» que aparece en el anuncio de pésimo gusto con el que la SGAE (Sociedad General de Autores y Editores)⁵ intentó concienciar a los melómanos de la necesidad de pasar por su caja. Como la gran mayoría de los chiflados que malgastamos nuestro tiempo en locales de ensayo y nuestro dinero en instrumentos y amplificadores, prefiero la satisfacción personal de saber que alguien se molesta en escuchar mi música a las treinta pesetas que me tocan por cada copia vendida (la cuarta parte si el disco está de oferta o es comprado durante una campaña de televisión).

Por favor, ¡pirateen mis canciones!

Este artículo se publicó por primera vez el 17 de enero de 2001 en Baquia <<http://www.baquia.com/>> y fue galardonado con el I Premio de Periodismo Digital José Manuel Porquet <<http://www.congresoperiodismo.com/premio.htm>>. Meteosat ha dejado de existir y algunos enlaces y datos han sido actualizados a partir de la versión original del texto, disponible en <<http://www.baquia.com/com/20010118/art00001.html>>. Se permite la reproducción total o parcial de este artículo siempre que se enlace a la fuente, <<http://www.escolar.net/>>, y se cite al autor.

Si mi gerente, ese imaginario del que les hablaba antes, fuese listo, estaría de acuerdo conmigo. Por cada concierto que doy, gano, dependiendo del aforo y la generosidad del promotor, entre 100 y 400 euros limpios. Prometo que si acuden a alguno de ellos, no les pediré una fotocopia del código de barras del CD para entrar. Como todos los músicos que hayan hecho las cuentas, sé que son más rentables 100.000 fans piratas que llenen mis conciertos a 10.000 originales.

El MP3, Kazaa⁶ o Gnutella⁷ tampoco van a acabar con la música. Ni con la mía ni con la de nadie. Les aseguro que, afortunadamente, puedo prescindir de los 85 euros mensuales que generan mis derechos de autor y mis royalties. A Metallica, y a cualquier grupo superventas, la regla, aunque sus cifras sean mayores, le vale igual.

Dan mucho más dinero los conciertos, las camisetas y los anuncios que un grupo de su fama puede grabar, que el *royalty* (entre el 8 y el 15% del precio de venta a mayorista) que pagan las multinacionales por disco vendido. Es cierto que las compañías discográficas costean la grabación y la promoción de los músicos, pero ¿conocen algún otro negocio en el que el reparto entre los que aportan la idea y la mano de obra y los que ponen el dinero sea tan desigual?

Les confieso que no entiendo las razones que movieron a Metallica y compañía a poner la cara por sus patrones en el juicio contra Napster. Todo, para que sus fans se la partan⁸, pacte Dios con el Demonio y Napster pase de pirata a corsario⁹. A mí se me habría puesto cara de tonto.

La distribución gratuita de las canciones por Internet no terminará con la creación musical, pero espero que sí lo haga con los abusivos tratos que impone la industria discográfica. Y eso que los 'juntantotas', con el tiempo, hemos mejorado bastante. Si los pobres músicos de *blues*¹⁰ de los años

Autor

Ignacio Escolar es periodista. Edita Spanishpop.net <<http://www.spanishpop.net/>>, colabora en <<http://www.gsmbox.es/>>, en el mensual GEO y se ocupa de "El Navegante", la sección dedicada a Internet de Informativos Telecinco 1:30. Su trabajo remunerado permitía que <iescolar@informativost5.com> pudiese pagar los teclados y el bajo con los que tocaba en el quinteto Meteosat, un grupo de Universal Music <<http://www.universalmusic.es/>>, la compañía de Metallica. Ambos correos electrónicos los responde la misma persona y su *nick* en Napster era *redskai*.

cuarenta -esos a los que el sello RCA (hoy, propiedad de Bertelsmann, el socio de Napster) pagaba seis dólares y una botella de bourbon por grabar sus canciones- oyesen los lamentos del batería de Metallica¹¹, Lars Ulrich...

No puedo alegar que no sabía dónde me metía cuando hace un año y medio firmé mi contrato con Universal Music¹². En aquella reunión, un alto directivo de la compañía me resumió en una sola frase los nueve folios del acuerdo: «Las discográficas somos un mal necesario». No lo voy a negar. Sin ellas, mi grupo jamás habría vendido 15.000 discos. Aunque estoy seguro de que sí hubiese podido regalarlos.

Notas

¹ <<http://www.angelfire.com/fm/calimonche/enrique.ram>>

² Los porcentajes de ventas de discos, entre otras muchas cosas de este artículo, están sacados del polémico artículo de Courtney Love acerca de los desmanes de la industria del disco, disponible en <<http://www.salon.com/tech/feature/2000/06/14/love/index.html>>.

³ <<http://www.audiomusica.com/imgs/analisis/ms2000.htm>>

⁴ <http://www.sonicstate.com/synth/_inc/picview.cfm?synthid=234>

⁵ <<http://www.sgae.es/>>

⁶ <<http://www.kazaa.com/>>

⁷ <<http://www.gnutella.co.uk/>>

⁸ <<http://www.metallicasucks.com/>>

⁹ <<http://www.baquia.com/com/20001108/art00001.html>>

¹⁰ <<http://www.thebluehighway.com/>>

¹¹ <http://www.metallica.com/napsterforum/lars_statement.html>

¹² <<http://www.universalmusic.es/>>

Conocimiento abierto / *Open Knowledge*

Gwen Hinze

Abogado, *Electronic Frontier Foundation*

<gwen@eff.org>

Traducción: José María Cañas Plaza

Resumen: *la Directiva Europea sobre Propiedad Intelectual requiere a cada estado miembro la elaboración de leyes con objeto de proteger las medidas tecnológicas usadas por los propietarios intelectuales para controlar el acceso a sus trabajos protegidos. Muchos estados miembro están elaborando borradores de esas leyes. Este artículo revisa la experiencia norteamericana con el Decreto sobre la Propiedad Intelectual Milenio Digital y argumenta que la implementación de la legislación de los estados miembro en esta materia debería incluir excepciones, permitiendo sortear esas medidas para usos legítimos y actividades socialmente provechosas. Además analiza el nuevo régimen de protección tecnológica contemplado en el borrador del 2003 de la Directiva Europea para la Aplicación de los Derechos de Propiedad Intelectual.*

Resumen: *control antipiratería, derechos digitales, DMCA, EUCD, normativa legal, propiedad intelectual.*

1. Introducción

En los dos años desde que se aprobó la Directiva Europea sobre Armonización de Ciertos Aspectos de la Propiedad Intelectual y Derechos Asociados en la Información de la Sociedad (Directiva 2001/29/EC, en adelante EUCD)¹, sólo tres estados miembro de la Unión Europea han aprobado legislación que proteja las medidas tecnológicas para controlar el acceso a trabajos registrados. La protección legal para estas medidas tecnológicas ha tenido por lo tanto un impacto mucho más limitado en los consumidores y tecnólogos de Europa que el que ha tenido en Estados Unidos la Ley sobre la Propiedad Intelectual Milenio Digital (*Digital Millennium Copyright Act*, en adelante DMCA)².

Sin embargo, parece que esto va a cambiar en los próximos meses. Primeramente, varios estados miembro están considerando borradores de leyes para materializar el régimen antipiratería de la directiva EUCD. En segundo lugar, el parlamento europeo ha elaborado recientemente el borrador de una Directiva para Aplicar los Derechos de Propiedad Intelectual, la cual, caso de aprobarse, podría restringir aún más la posibilidad de que los consumidores y tecnólogos hagan uso de trabajos digitales. La experiencia en Estados Unidos con la ley DMCA demuestra los peligros de una sobreprohibición de mecanismos para saltarse los controles, y recalca la necesidad de excepciones efectivas para usos legítimos en la legislación concreta de cada país miembro para materializar la directiva EUCD.

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

Este artículo se publica bajo la licencia *Creative Commons-Attribution-NoDerivs-NonCommercial 1.0*

2. Protección legal para controles tecnológicos: la Directiva Europea

El Parlamento Europeo y el Consejo aprobaron la directiva EUCD en mayo del 2001. Entre otras cosas, la directiva EUCD establece el marco que obliga a cada estado miembro a elaborar leyes que protejan los controles tecnológicos en trabajos registrados³. El artículo 6 de la directiva EUCD incorpora las obligaciones de dos tratados de 1996 de la Organización Mundial de la Propiedad Intelectual (WIPO): el artículo 11 del Tratado de la WIPO sobre la Propiedad Intelectual⁴ y el artículo 18 del Tratado de la WIPO sobre Actuaciones y Fonogramas⁵.

Esas disposiciones obligan a los países firmantes a «*proporcionar protección legal adecuada y remedios efectivos contra la transgresión de los controles tecnológicos que son usados por los autores en conexión con el ejercicio de sus derechos ... y que restringen los actos ... que no están autorizados por los autores respectivos o que no están permitidos por la ley*». Estas disposiciones fueron incorporadas en los tratados de la WIPO por orden de los detentores de la propiedad intelectual de música y películas, los cuales, precisamente antes de poner sus contenidos accesibles a los consumidores en formato digital, solicitaron apoyo legal para las medidas tecnológicas que usaban para proteger sus contenidos de la reproducción digital no autorizada.

El artículo 6 de la directiva EUCD ha sido criticado por inclinar la balanza en muchos aspectos hacia el lado de los dueños de la propiedad intelectual, a costa del interés general. Primero, el artículo 6 crea un marco que ofrece a los detentores de los derechos un nivel de protección más elevado que el que

Autor

Gwen Hinze es abogado en la Electronic Frontier Foundation <<http://www.eff.org>> y está especializado en temas de legislación sobre propiedad intelectual y de regulación de Internet. Fundada en 1990, la Electronic Frontier Foundation es una organización sin ánimo de lucro que vive de las aportaciones de sus miembros y que está dedicada a la protección de los derechos digitales individuales y de la libertad de expresión en línea. Con sede en San Francisco (EE.UU.), la EFF ha estado involucrada en todos los principales casos judiciales relativos a la interpretación de las normas restrictivas de la ley norteamericana sobre la Propiedad Intelectual Milenio Digital (DMCA) de 1998 y ha testificado ante la Oficina de Copyright de los EE.UU. sobre la necesidad de excepciones a esas normas.

proporcionan los dos tratados de la WIPO o incluso la ley de Propiedad Intelectual. El artículo 6 efectivamente concede a los detentores de derechos (propiedad intelectual, otros asociados y derechos sui generis sobre bases de datos) la capacidad de controlar el *acceso* a sus trabajos, no sólo la protección contra la violación de la propiedad intelectual y actos no autorizados por los propietarios. Segundo, su prohibición de las herramientas que permiten saltarse esos controles tecnológicos de protección parece más amplia de lo necesario. Por ejemplo, al contrario que la ley DMCA (la respuesta de los Estados Unidos a los tratados de la WIPO), el artículo 6.3 prohíbe la «*posesión con propósito comercial*», además de la creación y distribución de herramientas para sortear esos controles.

Tercero, como hecho más importante, el artículo 6 no contempla excepciones a la prohibición de herramientas que superan esos controles para usos legales atendiendo a la ley nacional del derecho de copia (como la copia personal), y actividades legítimas, como las pruebas de seguridad de los sistemas, la investigación sobre criptografía o la ingeniería inversa. El artículo 6.4 contempla que ciertas excepciones y limitaciones a la propiedad intelectual según la ley nacional (en el artículo 5) se acomodarán dentro del régimen de protección tecnológica a través de acuerdos voluntarios entre el público y los detentores de los derechos.

En ausencia de esos acuerdos voluntarios los estados miembro deben requerir a los detentores de los derechos a que hagan accesible a los beneficiarios los medios para disfrutar de las excepciones o limitaciones a la propiedad intelectual. Sin embargo, no está claro a que obliga esto y tiene limitada utilidad práctica porque los detentores de la propiedad intelectual pueden optar por no seguir ese requerimiento para trabajos distribuidos en la red y sujetos a disposiciones contractuales⁶.

Sin embargo el impacto real del régimen de protección legal en los consumidores y los tecnólogos no se puede afirmar analizando el texto de la directiva EUCD. En gran parte, la directiva EUCD determina lo que los países miembros de la Unión Europea pueden y no pueden hacer al legislar sobre protección para los controles tecnológicos, y el alcance incierto de la directiva se debe en parte a la fórmula compleja de armonización para los recortes al derecho de copia, que las legislaciones nacionales pueden decidir activar bajo el artículo 5 de la directiva. Mientras que la directiva establece límites a las leyes nacionales en este área, el impacto real en los consumidores y tecnólogos dependerá del texto de las legislaciones nacionales, y en particular, del alcance de las excepciones que permiten trasgredir los controles de protección.

3. Estado actual de la materialización de la Directiva europea EUCD

La fecha tope para que los estados miembro materializasen la directiva EUCD era el 31 de Diciembre de 2002. Italia aprobó su legislación respectiva en marzo del 2003. Un borrador homólogo fue aprobado por la Cámara Baja del Parlamento alemán el 11 de abril y se espera que sea aprobado por la Cámara Alta en mayo de este mismo año. En Francia también

se ha elaborado el texto de un borrador. En Austria se presentó un proyecto de ley en abril y se espera su aprobación en julio. Portugal, Reino Unido y Holanda están considerando comentarios públicos sobre sus respectivos borradores. En Finlandia un proyecto de ley fue rechazado por su parlamento en Enero del 2003, pero puede ser modificado y relanzado a finales de este año⁷.

Acelerar la aprobación de leyes antipiratero conlleva dos riesgos elevados. Primero, existe el riesgo de crear una sobreprohibición que impida (o dificulte en exceso) posibles excepciones que permitan saltarse las protecciones electrónicas con la finalidad de ejercer los derechos existentes, como la copia privada (permitida por las leyes de propiedad intelectual en algunos estados, como Alemania), u otras actividades legítimas. Segundo, existe el riesgo de efecto repetidor, según el cual una prohibición demasiado amplia sobre el piratero y las herramientas que sortean las protecciones electrónicas, y un lenguaje limitado sobre las excepciones, sean adoptadas como 'plantilla tipo' para las legislaciones nacionales.

4. Cuatro años del Decreto DMCA: la experiencia norteamericana

La experiencia en Estados Unidos con la ley DMCA claramente demuestra la necesidad de que los estados europeos incluyan excepciones que permitan a los consumidores y tecnólogos sortear los mecanismos electrónicos de protección, bien para ejercer sus derechos reconocidos, bien para permitir actividades socialmente beneficiosas como la investigación en criptografía, el análisis de seguridad de los sistemas, el fomento de innovación tecnológica y el desarrollo de tecnología interoperable.

Como el artículo 6 de la Directiva europea EUCD, la sección 1201 del Estatuto sobre Propiedad Intelectual en Estados Unidos incluye dos prohibiciones. Primero, se prohíbe el hecho de sortear una protección tecnológica que controla el acceso a material propietario, y segundo, prohíben ciertas 'herramientas' (tecnología, dispositivos, productos y servicios) que pueden ser usadas para burlar las medidas de control de acceso y copia. Aunque la ley DMCA incluye excepciones a la prohibición antipiratero para siete actividades, en la práctica estas excepciones se han revelado como demasiado estrechas para servir a su propósito. Como resultado, las disposiciones antipiratero del decreto, demasiado amplias, han provocado una seria amenaza para al menos tres tipos de actividades que no violan las propiedad intelectual.

4.1. Freno para la investigación científica y la libre expresión

Primeramente, la sección 1201 ha sido utilizada por los propietarios de derechos para frenar la libre expresión y la investigación científica lícita. Dos incidentes que claramente ilustran esta tendencia son las amenazas de juicio al profesor Felten de la Universidad de Princeton, y el arresto del programador ruso Dmtry Sklyarov.

El profesor Edward Felten y un equipo de investigadores de Xerox y las Universidades de Princeton y Rice fueron amenazados con emprender acciones legales contra ellos por violar

la ley DMCA cuando respondieron con éxito al reto público planteado por el grupo de la industria musical, *Secure Digital Music Initiative* (en adelante SDMI), acerca de borrar marcas de agua digitales en música digital, y trataron de publicar un artículo describiendo su metodología. Por el hecho de que el artículo describía vulnerabilidades en la tecnología de marcas de agua digital que el SDMI usaba para proteger la música digital, el SDMI consideró que la publicación del artículo violaría la prohibición expresa en la ley DMCA de distribuir «herramientas que sortean controles de protección».

La ley DMCA contiene una excepción para investigación en criptografía, en la sección 1201(g), pero no permitía la publicación de esta clase de trabajos. Primero, esto obliga a los investigadores a esforzarse para obtener una autorización expresa de los poseedores de los derechos de autor antes de hacer sus investigaciones. Segundo, incluso si consigue ese permiso para investigar, no les estaría permitida la publicación de sus resultados de investigación. Después de que el profesor Felten y su equipo entablaran un pleito, la amenaza fue abandonada y a los investigadores se les permitió presentar una parte de su investigación en una conferencia posterior⁸.

El programador ruso Dmitry Sklyarov estuvo detenido en los Estados Unidos durante cinco meses en el 2001, después de presentar un artículo en una conferencia describiendo las debilidades en la protección de los libros electrónicos de Adobe (e-Book). En Rusia, Sklyarov había trabajado en un programa que permitía a los dueños de libros electrónicos de Adobe convertirlos al formato PDF, también de Adobe, borrando de este modo las restricciones de uso embebidas en el propio libro. El propietario de la empresa rusa en la que trabajaba Sklyarov ofreció este programa a la venta a través de Internet. Aunque el programa podía usarse para hacer usos lícitos de libros electrónicos comprados (como imprimirse una copia), también podía ser usado para copiar un libro electrónico sin el permiso del editor, y el Departamento de Justicia de los Estados Unidos acusó a Sklyarov del delito de traficar con una herramienta que sortea los controles de protección. Finalmente se permitió a Sklyarov volver a casa, pero el Gobierno continuó su batalla contra ElcomSoft, su empresa rusa. En diciembre de 2002, 18 meses después del arresto de Sklyarov, un jurado absolvió a la compañía de todos los cargos.

El freno de estos casos a la investigación científica y la publicación de información sobre vulnerabilidades en los sistemas de control de acceso ha sido profundo. En el 2001, un prominente criptógrafo y analista de seguridad de sistemas danés, Niels Ferguson, declinó publicar los detalles de una brecha de seguridad que había descubierto en el estándar de encriptación de vídeo para la protección del contenido digital de gran ancho de banda (High Bandwidth Digital Content Protection).

A comienzos de año, el editor estadounidense del libro de Andrew «Bunnie» Huang en el que describía los fallos de seguridad en la consola de videojuegos X-Box de Microsoft, renunció al libro argumentando riesgos relativos al Decreto DMCA. Y en octubre del 2002, el anterior consejero jefe de seguridad cibernética de la Casa Blanca, Richard Clarke, advirtió que la ley DMCA había sido usada para ralentizar la investigación legítima en seguridad y solicitó una reforma del decreto.

Hay una creciente preocupación en los Estados Unidos por el impacto de una sobreprohibición de herramientas que sortean controles de protección en la lícita investigación científica. El congreso norteamericano está actualmente considerando dos diferentes estatutos preliminares que corregirían las disposiciones antipiratería presentes en el Decreto DMCA, para permitir esa transgresión y el uso de esas herramientas en la investigación científica⁹.

4.2. El impacto sobre uso legítimo y no infractor

En segundo lugar, las disposiciones de la ley DMCA han perjudicado la capacidad de los consumidores de hacer usos legales y no infractores de los trabajos digitales. Empleando medidas tecnológicas de protección para controlar el acceso y uso de los trabajos sujetos a la propiedad intelectual y haciendo ilegal burlar esos controles, los propietarios de los derechos han reemplazado en la práctica la ley de propiedad intelectual por la protección tecnológica. Esa protección tecnológica puede bloquear acciones que no implican la vulneración de la propiedad intelectual. La prohibición por la ley DMCA de la elaboración y distribución de herramientas que podrían ser usadas para permitir la transgresión de esos controles en usos lícitos ha redefinido radicalmente los derechos de los consumidores en Estados Unidos.

Por ejemplo, varias personas y la revista 2600 Magazine <<http://www.2600.com>> han sido demandados por albergar (y en algunos casos meramente por poner enlaces a sitios que lo contenían) el programa DeCSS. Este programa puede ser usado para descifrar las protección CSS en los DVDs, tanto para usos legítimos como para los ilegítimos. Un tribunal estadounidense está considerando actualmente la legalidad de un software similar vendido por 321 Studios <<http://www.321studios.com>>, el cual puede usarse para hacer una copia personal de seguridad de un DVD comprado. Y recientemente se ha solicitado a la Oficina Norteamericana de Propiedad Intelectual que conceda una exención a la prohibición de burlar los controles de protección para permitir a los consumidores avanzar rápido la reproducción en los anuncios 'insaltables' que aparecen en sus DVDs¹⁰.

4.3. Amenaza a la competición, a la ingeniería inversa y a la innovación tecnológica

En tercer lugar, la ley DMCA ha sido usada para dificultar los esfuerzos de competidores legítimos en crear productos interoperables. Aunque la ley DMCA incluye una excepción para permitir la ingeniería inversa con objeto de favorecer la interoperabilidad (sección 1201(f)), se ha hecho patente que es demasiado limitada para ayudar a quienes quiera usarla. Porejemplo, Lexmark, el segundo mayor distribuidor de impresoras en los Estados Unidos, ha usado la ley DMCA para bloquear el desarrollo de un mercado con los cartuchos de impresoras Lexmark reciclados. Ciertos cartuchos de impresoras Lexmark incluyen un chip especial que contiene una rutina de autenticación que interopera con el *driver* de las impresoras Lexmark, para evitar el rellenado de tinta en esos cartuchos. La empresa Static Control Components hizo ingeniería inversa del chip de los cartuchos de Lexmark y vendieron su propio chip a recicladores de cartuchos, para permitir que cartuchos rellenos funciona-

ran bien en las impresoras de Lexmark. Lexmark ha usado la ley DMCA para obtener un mandato judicial que prohíbe a Static Control vender su chip.

En un caso similar el grupo Chamberlain, un fabricante de dispositivos electrónicos para abrir puertas de garaje, ha usado la ley DMCA para demandar a Skylik, el fabricante competidor de otro dispositivo de apertura, que es capaz de interoperar con el mecanismo de las puertas de garaje hecho por Chamberlain. Chamberlain argumenta que el dispositivo interoperable de apertura de Skylink ha violado la ley DMCA saltándose un régimen de autenticación que controla el acceso al motor de la puerta de garaje, montado en la unidad de Chamberlain.

Está claro que el Congreso de los Estados Unidos no buscaba que la ley DMCA fuera usada para poner obstáculos infranqueables al desarrollo de mercados legítimos de productos de consumo interoperables, no sometidos a la propiedad intelectual. Sin embargo los amplios términos de sobreprohibición antipiratero del Decreto DMCA, la estrecha excepción a favor de la ingeniería inversa, y la ausencia de excepciones para la transgresión de los mecanismos de control con fines legales, han dejado abierto el camino a la posibilidad de que la ley DMCA sea usado de este modo. La directiva europea EUCD ciertamente limita cómo los estados miembro pueden materializar la protección legal de las medidas tecnológicas antipiratero, pero la experiencia estadounidense subraya la importancia de redactar una prohibición antipiratero convenientemente estrecha y de contemplar excepciones efectivas en la legislación de cada estado miembro.

5. ¿Un nuevo régimen de protección tecnológica más amplio?

Posiblemente más problemático que el alcance incierto del artículo 6 de la directiva EUCD es una disposición en el artículo 21 del borrador de la Directiva sobre Medidas y Procedimientos para Garantizar el Cumplimiento de los Derechos de la Propiedad Intelectual, hecho público por la Comisión Europea en enero del 2003, el cual parece crear un nuevo superrégimen de protección tecnológica.

El artículo 21 está expresamente diseñado para complementar al artículo 6 de la directiva EUCD y proporcionaría protección a una lista de elementos mucho más extensa que el artículo 6, o la sección 1201 del Decreto sobre Propiedad Intelectual en Estados Unidos. Caso de aprobarse, el artículo 21 haría dos cosas:

1. Primeramente, crearía protección legal para cualquier tipo de trabajo que incluya o incorpore una característica o marca «manifiestamente identificable». Incorporando dicha marca, cualquiera que lo deseara podría ejercer derechos sobre trabajos no protegibles, sobre bases de datos, o trabajos de dominio público.

El artículo 21 no especifica qué tipo de características o marcas garantizarían dicha protección. En contraste con los «controles antipiratero» en el artículo 6, las marcas de autenticidad no necesitarían cifrar, encriptar o proteger tecnológicamente el acceso al trabajo para darle protección legal. Una mera marca de identificación, como un holograma, logo o posiblemente una marca de agua digital, sería suficiente.

El artículo 21 usa la frase «*incorporación de elementos que sean manifiestamente identificables por clientes y consumidores*». «Manifiestamente identificables» no queda definido y no está claro si las marcas de autenticidad se limitarían a marcas físicas, o si por ejemplo, una etiqueta en el soporte digital que indicara al consumidor la existencia de una marca no material (como la incorporada en una marca de agua digital) sería suficiente para activar la protección legal.

2. En segundo lugar, el artículo 21 prohibiría potencialmente una colección de dispositivos transgresores más extensa que la prohibida bajo el artículo 6 de la Directiva EUCD o la ley DMCA estadounidense. Prohibiría «*cualquier dispositivo técnico el cual esté diseñado para sortear otro dispositivo técnico, lo que permite la fabricación de productos que infringen los derechos de propiedad industrial y que incorporan elementos manifiestamente identificables descritos en el punto (a)*». Si la definición de «dispositivo técnico» es suficientemente amplia como para incluir marcas no materiales incorporadas, como las marcas digitales de agua, entonces esta disposición prohibiría el uso de cualquier tecnología o dispositivo diseñado para borrarlas. El artículo 21 no contiene ninguna resolución que contemple las excepciones, por lo tanto la prohibición parecería aplicarse incluso si la razón del consumidor para saltarse los controles fuera legal.

Como el artículo 6 de la Directiva EUCD y la sección 1201 del estatuto estadounidense sobre propiedad intelectual, el régimen contemplado por el artículo 21 tiene el potencial de reducir significativamente los derechos existentes de los tecnólogos y consumidores de medios digitales.

Notas

¹ Disponible en <http://europa.eu.int/eur-lex/en/consleg/pdf/2001L0029_do_001.pdf>, en inglés.

² Disponible en <<http://www.loc.gov/copyright/legislation/dmca.pdf>>, en inglés.

³ La Directiva EUCD no se aplica a programas de ordenador. Sin embargo, la Directiva 91/250/EEC proporciona un régimen de protección tecnológica similar a los programas de ordenador.

⁴ Disponible en <<http://www.copyright.gov/wipo/treaty1.html>>, en inglés.

⁵ Disponible en <<http://www.copyright.gov/wipo/treaty2.html>>, en inglés.

⁶ *Technological Measures vs. the Copyright Limitations*, de Kamiel J. Koelman, Junio de 2001, disponible en el Institute for Information Law, Universidad de Amsterdam, y en la dirección <<http://www.ivir.nl/publications/koelman/alaiNY.html>>; *Why the Copyright Directive is Unimportant, and Possibly Invalid* de Bernt Hugenholtz, en EIPR 11, páginas 501-502, 2000, disponible en <<http://www.ivir.nl/publications/hughholtz/opinion-EIPR.html>>; y en *The Evolution of Article 6.4 of the European Information Society Copyright Directive* de Alvisse Maria Casellati, en VLAJ.L. & Arts 369, 24 Colum., 2001.

⁷ Véase <<http://wiki.ael.be/index.php/EUCD-Status>>.

⁸ Para más detalles sobre esto y otros incidentes descritos, véase el artículo de la Electronic Frontier Foundation *Unintended Consequences: Four Years under the DMCA*, disponible en <http://www.eff.org/IP/DMCA/20030102_dmca_unintended_consequences.html>.

⁹ La ley norteamericana sobre Derechos de los Consumidores de Información Digital (*Digital Media Consumer Rights Act*, H.R. 107, Cong. 108), presentada por los congresistas Boucher y Doolittle; la ley B.A.L.A.N.C.E. (H.R. 1066, Cong. 108) presentada por el congresista Lofgren

¹⁰ Véase la petición, por parte de Electronic Frontier Foundation, de cuatro excepciones a la prohibición de herramientas para saltarse controles de protección en el Decreto DMCA, disponible en <www.eff.org/IP/DMCA/20021218_EFFPKcomments.pdf>.

Conocimiento abierto / *Open Knowledge*

Ross Anderson
Universidad de Cambridge

<Ross.Anderson@cl.cam.ac.uk>

Traducción: Alicia Díaz Migoyo

Resumen: *el desarrollo estratégico más significativo que se ha producido en el último año ha sido 'informática de confianza' --en inglés Trusted Computing (o TC). En este artículo se describe a grandes rasgos qué es TC y se esbozan los efectos que pudiese tener en el sector informático y en las personas que trabajan en él.*

Palabras clave: *antimonopolio, atado de producto, control de accesorios, copyright, DMCA, EUCD, Intel, Microsoft, monopolio, Palladium, política sobre competencia, TCPA, TCG, Trusted Computing.*

1. Introducción

Uno de los problemas más complejos que tienen los profesionales informáticos es hacer frente a las estrategias de precios que los proveedores utilizan para sacarle hasta el último céntimo a su base de clientes. Los proveedores dominantes --como Microsoft hoy e IBM en la generación anterior-- intentan atar a los clientes a sus arquitecturas para ir extendiendo el control de un producto a otro. Muchos productos funcionan en ciclos que van «de ganga a timo»: una vez que has comprometido a tu organización con una determinada tarjeta inteligente, o paquete contable, los precios suben misteriosamente. Otra estrategia es la de atar productos, y un buen ejemplo son los cartuchos de tinta para impresoras. Son los cartuchos los que hacen rentables a estas impresoras: esta fórmula permite a los fabricantes utilizar el mismo producto para los grandes usuarios de las empresas y para los usuarios domésticos que miran más el precio. A esta subvención cruzada le ponían un cierto límite los cartuchos reciclados y los de otras marcas. Por eso ahora muchos cartuchos vienen con chips para que las impresoras los autentifiquen, una práctica que empezó en 1996 con la Xerox N24 (ver en [5] la historia de los chips de cartucho). En un sistema cualquiera, si la impresora detecta un cartucho de otra marca o un cartucho reciclado, puede bajar el rendimiento de 1.200 p.p.p. a 300 p.p.p., o incluso dejar de funcionar.

Más reciente es la introducción de la fecha de caducidad: los cartuchos de la HP BusinessJet 2200C caducan cuando están en la impresora más de 30 meses, o a los 4,5 años de la fecha de fabricación - provocando la indignación de los consumidores, por supuesto.

Esta práctica con los cartuchos de impresora nos lleva a un conflicto comercial entre EE.UU. y la Unión Europea. En EE.UU., el fabricante de impresoras Lexmark ha conseguido un mandamiento judicial que impide la venta de cartuchos de terceros con chips que permitan su utilización en

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

Este artículo se publica bajo la licencia *GNU Free Documentation License* es una versión abreviada, especial para *Novática y Upgrade*, de un artículo titulado «*Cryptography and Competition Policy - Issues with 'Trusted Computing'*» que se puede encontrar en <<http://www.ross-anderson.com/>>.

impresoras Lexmark. Al mismo tiempo, el Parlamento Europeo ha aprobado una «Directiva sobre material eléctrico y electrónico de desecho» que en 2006 forzará a los Estados miembro a impedir que las compañías incumplan la normativa sobre reciclado de la UE mediante la fabricación de productos con chips que impiden su reciclado [8].

El control postventa y el atado de un producto son prácticas que están creciendo con rapidez y utilizando todo tipo de mecanismos técnicos. Los fabricantes de teléfonos móviles, por ejemplo, suelen ganar más con las baterías de los teléfonos que con los teléfonos en sí, así que han introducido chips de autenticación que dificultan la utilización de baterías de la competencia [10]. Los fabricantes de coches están utilizando el bloqueo del formato de datos para impedir que mecánicos independientes reparen los vehículos de sus clientes [12]. Y las compañías de juegos de ordenador llevan años cobrando *royalties* a los desarrolladores de software, para subvencionar con esos ingresos la venta de consolas [11].

¿Esto es bueno o malo para el mercado? Según los economistas, «depende». Hal Varian argumenta que, desde el punto de vista de la política de ventas, unir las impresoras a los cartuchos puede no ser demasiado cuestionable, porque en el mercado de las impresoras todavía hay competencia y esa práctica hace que los fabricantes de impresoras compitan más para vender más unidades y, por lo tanto, bajen los precios [9].

Pero cuando esta técnica se utiliza para unir dos mercados en los que la competencia es relativamente escasa, el mercado de sistemas operativos o de servidores de Internet, por ejemplo puede limitar las opciones de los clientes y hacer subir los precios. Ésta fue una de las objeciones que se le pusieron a la iniciativa Passport de Microsoft en términos de política de competencia: los comerciantes que querían utilizar Passport estaban obligados a utilizar también servidores de Microsoft.

Autor

Ross Anderson dirige el grupo de seguridad del laboratorio de Informática de la Universidad de Cambridge (Reino Unido). Pertenece a la Institution of Electrical Engineers y al Institute of Mathematics and its Applications. Ha publicado artículos bien conocidos sobre temas de políticas de seguridad, desde privacidad médica a sistemas bancarios, así como sobre tecnologías subyacentes como criptografía y protección contra falsificaciones. Es autor del libro «*Security Engineering - A Guide to Building Dependable Distributed Systems*». Preside la Foundation for Information Policy Research.

Los fabricantes tendrán más facilidades aún para implementar una compleja política de precios y un mayor control postventa gracias a la introducción de *Trusted Computing* (TC), o «informática de confianza» [2].

2. Informática de confianza

En junio 2002, Microsoft anunció el lanzamiento para el 2004 de Palladium, una versión de Windows para implementar la 'Informática de confianza'. En este contexto, «de confianza» significa que terceros pueden confiar en el software de un PC con el que están en comunicación y que el propietario de la máquina no ha modificado ese software. Los programas también se podrán comunicar unos con otros --y con sus autores-- con total confianza. Esto abre toda una serie de nuevas e interesantes posibilidades.

La aplicación más obvia sería la gestión de los derechos digitales (*Digital Rights Management* o DRM, en inglés): Disney podría vendernos DVD que se describirían y funcionarían en una plataforma Palladium, pero que no se podrían copiar. La industria discográfica podría vendernos descargas de música que no se podrían intercambiar, o CDs que sólo se podrían escuchar tres veces o el día de nuestro cumpleaños. Esta aplicación será polémica, mientras que otras quizá no lo serán tanto. Un ejemplo: las plataformas con TC pueden albergar juegos en los que sea más difícil hacer trampas.

Palladium surgió gracias al trabajo de la Alianza para una Plataforma con Informática de Confianza (*Trusted Computing Platform Alliance* o TCPA, en inglés), fundada por Microsoft, Intel, IBM y HP. A ella se ha unido ahora ADM y se ha rebautizado como el Grupo de Informática de Confianza (*Trusted Computing Group*, o TCG en inglés) [13]. El TCG propone rediseñar el hardware de PC para darle a la unidad central de proceso (CPU) más privilegios (que permite a los procesos acceder a zonas de memoria inaccesibles incluso para los superusuarios normales) y un componente de seguridad del hardware --el chip Fritz-- que comprueba el software y el hardware que está utilizando una máquina. Los chips Fritz de diferentes máquinas se pueden comunicar entre sí. El papel de Fritz en el entorno 'de confianza' es asegurar a terceros que nuestra máquina es realmente la que nosotros decimos que es y que realmente utiliza el software que decimos que utiliza.

No todo el mundo está de acuerdo con el calificativo 'de confianza' para describir a esta tecnología. Microsoft prefiere hablar de 'Informática fiable' (*Trustworthy Computing* en inglés), porque confiar en un sistema no significa necesariamente que el sistema sea fiable. Si vemos a un empleado de la Agencia de Seguridad Nacional (NSA) norteamericana en los servicios de un aeropuerto internacional vendiéndole material clave a un diplomático chino, y siempre que esa operación no estuviese autorizada, diríamos que ese empleado es 'de confianza pero no fiable' (de hecho, la definición de la NSA de un sistema de confianza es «*un sistema que puede romper la política de seguridad*»).

En el lado opuesto del debate, Richard Stallman, de la Fundación para el Software Libre (FSF), prefiere hablar de 'Informática traicionera' (*Treacherous Computing* en inglés), porque el objetivo real de la tecnología del TCG es que el propietario del PC deje de tener el control efectivo del mismo [15]. Yo utilizaré TC y así el lector puede convertirlo en las siglas inglesas de 'Informática de Confianza', 'Informática Fiable' o 'Informática Traicionera', lo que prefiera.

2.1. Control y gobernanación

Si el dueño de un ordenador ya no va a poder controlarlo, la gran pregunta es ¿quién ejercerá ese control? Las compañías relacionadas con TC han expresado diferentes opiniones dependiendo del momento. La especificación 1.0 de la TCPA original sugería una jerarquía en los organismos que pudiesen certificar los diferentes componentes de hardware y de software que fuesen a componer un sistema TC. De esta forma habría un control centralizado ejercido por un consorcio de empresas del sector.

La opinión actual de la industria es que los fabricantes de aplicaciones TC o del contenido de las mismas son los que deciden qué combinaciones de software y de hardware del sistema operativo son las aceptables. En el caso de DMR, por ejemplo, sería Disney --o quizá Microsoft como vendedor de Media Player-- el encargado de certificar una plataforma concreta como apta para reproducir Blancanieves. Un servidor mantenido por el fabricante de la aplicación suministraría lo necesario para cumplir las reglas que una aplicación concreta tiene que aplicar, como etiquetas para CD comerciales que digan «Prohibida su copia» o «Sólo copia de seguridad», o para películas que digan «Grabación permitida para visionado posterior; prohibida su copia».

3. Valor para usuarios corporativos y gubernamentales

Las aplicaciones de seguridad podrán especificar una amplia gama de políticas. Un sistema TC que, por ejemplo, se utilice para hacer cumplir una normativa gubernamental de protección de datos podría establecer que la información sólo puede enviarse hacia un nivel superior, de forma que un archivo 'confidencial' sólo se podría cortar y pegar en otro 'secreto', y no al revés. Pero resulta difícil que funcionen bien los mecanismos para controlar el flujo unidireccional de la información [1], así que es poco probable que estos mecanismos sean la aplicación rompedora (*killer app*) en lo que se refiere a la TC.

Utilizar sistemas TC para proteger secretos corporativos es la aplicación que se está utilizando para promocionar la TC. «*Tiene gracia*», comenta Bill Gates, «*esto se nos ocurrió para la música, pero luego nos dimos cuenta de que el correo electrónico y los documentos son terrenos mucho más interesantes*» [19]. En el Servidor Windows 2003 [16] se acaba de comercializar una implementación de los mecanismos de control de derechos de autor que se puede aplicar para controlar información confidencial en vez de música y vídeos.

El Servidor Windows 2003 permite al creador de un documento o de un archivo ejercer cierto control sobre el mismo, independientemente de donde se pueda enviar. Se podrá mandar un mensaje de e-correo con restricciones, como que el destinatario no lo pueda reenviar, o no lo pueda imprimir, o que sólo lo pueda leer si tiene una acreditación 'secreta', o que el documento sólo se pueda leer hasta finales de mes. Los usuarios de Windows que quieran utilizar esta función de TC pueden apuntarse a ella y entonces parece que será un servicio en línea el que decida si entrega una clave de descripción para la aplicación (esto se acababa de hacer público en el momento de redactar este artículo y no se habían decidido todavía los detalles de su funcionamiento).

Uno de los argumentos clave para la venta de esta tecnología es que la empresa puede hacer que, a los 90 días, los

mensajes de e-correo ya no se puedan leer. En Microsoft ya funciona esta norma interna. Dadas las tácticas cada vez más agresivas que se utilizan en los litigios legales, a algunas firmas les podría gustar la idea de que dichos mensajes sean tratados como llamadas telefónicas en vez de como cartas. Pero incluso una aplicación tan sencilla como ésta puede ser difícil de desplegar en el mundo real. Algunos bufetes de abogados podrían poner pegatas a recibir instrucciones de un cliente a través de un e-correo que sólo puede leer uno de los socios, o que no se pueda imprimir, o que sea ilegible a los 90 días. ¿Cómo se podría defender el bufete de acusaciones de mala práctica profesional, o qué garantías tendrían los otros socios?

Pero hay más: en muchos países, las leyes de exportación exigen a las compañías que conserven copias de las comunicaciones sobre software, documentación o *know-how* que se exporte y que estén comprendidas en la lista de doble uso. Esto puede suponer conservar los e-correos pertinentes durante tres años. Las normas contables pueden exigir la conservación de los e-correos pertinentes durante seis años. Se pueden prever innumerables escaramuzas entre políticas que exijan la destrucción y políticas que exijan la conservación. Como bien sabe cualquier director de Sistemas de Información gestor de Seguridad Interna, automatizar procedimientos que antes, dejando que fuese el raciocinio humano el encargado de dilucidar las cuestiones espinosas, habían evitado conflictos, es sinónimo de crear un campo de minas.

4. Valor para propietarios de contenidos

Las industrias discográfica y cinematográfica han presionado mucho para implantar mecanismos de tipo TC, buscando reforzar los sistemas de gestión de derechos digitales. Ya han conseguido una mayor protección legal para los sistemas existentes. Su argumento es que las copias digitales destruirán su negocio, pero el argumento está perdiendo fuerza porque hace ya varios años que resulta muy fácil copiar CD y no parece que eso haya afectado mucho a las ventas.

Analizando detenidamente la cuestión, no está claro que un mecanismo de DRM mucho más fuerte --como el prometido por TC-- supusiese una mejora importante en el status quo de los propietarios de contenidos [20]. Además, existe otro riesgo: si las máquinas TC se generalizan, los otros las pueden usar con la misma facilidad. Los usuarios pueden establecer 'redes negras' para intercambiar material prohibido de todo tipo y sería más fácil crear sistemas entre iguales (*peer-to-peer*), como Gnutella o Mojonation, pero que resisten mucho mejor los ataques de la industria discográfica, porque sólo podrán participar los verdaderos clientes. Ya no funcionarán los métodos actuales para atacar estos sistemas que consisten en denegaciones de servicio realizadas por clientes introducidos por un troyano [21]. Así que la implementación de TC podría tener como consecuencia no deseada que la industria discográfica fuese la víctima en lugar de la beneficiada.

5. Valor para fabricantes de hardware

La experiencia nos dice que los mecanismos de seguridad suelen favorecer los intereses de quien los paga y no los intereses de los clientes, para los que supuestamente se desarrollaron [1]. La publicidad nos contó que la autenticación y encriptación en los móviles GSM era para ofrecer una mayor seguridad a los usuarios, porque con los teléfonos analógicos eran más fáciles la clonación y las escuchas.

Pero la experiencia nos demuestra que los principales beneficiarios han sido las compañías telefónicas que financiaron esa mejora de seguridad.

Con los teléfonos analógicos, los usuarios que querían llamar gratis o defraudar al sistema llamando a números 900 controlados por asociados, podían clonar los teléfonos, y eso solía costarles dinero a las compañías telefónicas. Con el sistema GSM, los delincuentes pueden comprar teléfonos con tarjetas de crédito robadas (pasándoles el coste a los bancos) o, lo que es cada vez más frecuente, utilizan teléfonos robados en la calle (y el que paga el pato es el cliente). En cuanto a la confidencialidad, las agencias de espionaje de todo el mundo ya tienen acceso a las redes principales para conseguir datos claros de voz.

Por eso deberíamos examinar el posible efecto que TC tendría sobre la actividad empresarial de sus promotores.

En el caso de Intel, sus motivos para unirse a la TCPA fueron estratégicos. Intel copa la mayoría del mercado de microprocesadores para PC, de donde consigue la mayoría de sus beneficios, y la empresa crecería si crece el mercado de PC. Intel, por lo tanto, ha desarrollado un programa de investigación para apoyar una estrategia de «liderazgo en la plataforma», por la cual la empresa se pone a la cabeza de la industria para desarrollar tecnologías que amplíen la utilidad del PC, como el bus PCI y USB [23].

La parte positiva de esta estrategia es que Intel hizo crecer el mercado global de PC. La parte negativa es que utilizaron cárteles para la explotación de patentes y acuerdos obligatorios de licencias cruzadas para evitar que nadie de la competencia consiguiera una posición dominante en cualquier tecnología que pudiese poner en peligro su control sobre el hardware de PC. Los más descreídos señalan que Intel no podía permitirse que prevaleciera el bus de microcanal de IBM: no era sólo la competencia en la plataforma de hardware, era también que IBM no tenía interés alguno en ofrecer el ancho de banda necesario para que el PC compita con los sistemas superiores. En términos estratégicos, el efecto es parecido al que conseguían los antiguos romanos cuando destruían todos las edificaciones y todos los árboles que estaban cerca de sus carreteras y sus fortificaciones. La estrategia de Intel ha ido evolucionado hasta convertirse en una forma muy eficaz de sortear la ley antimonopolio.

6. Valor para fabricantes de software

El caso de Microsoft es aún más interesante. En su formato original, TC tenía potencial para eliminar el uso de software sin licencia: una plataforma 'de confianza' conectada con un servicio central de autorizaciones podría sencillamente impedir el uso de ese software ilegal. Los mecanismos para registrar el software se podrían hacer mucho más difíciles de sortear: el chip Fritz mantiene una lista de hardware y de componentes de software del sistema de una máquina TC, y se instala una comprobación online de los mismos.

Tras algunas protestas públicas, Microsoft afirma ahora que no habrá mecanismos de listas negras --por lo menos al nivel del sistema operativo [17]. El sistema Windows 2003 parece basarse en mecanismos más sutiles. El control no se ejercerá ahora de abajo a arriba por medio del hardware con TC, sino de arriba hacia abajo por medio de las aplicaciones. Disney

podrá decidir libremente en qué condiciones quiere ofrecer contenidos a sistemas con un determinado hardware y software. Si Disney decide cobrar 12,99 E por la versión en DVD de 'Blancanieves', 9,99 E por una descarga para TC/Windows utilizando Media Player, pero se niega a ofrecer contenidos para cualquier otra plataforma de ordenadores, Microsoft puede argumentar --frente a los medios y frente a las autoridades antimonopolio-- que quien niega los contenidos no es Microsoft sino Disney.

Los incentivos resultantes favorecen mucho a Microsoft: si TC/Windows se convierte en la plataforma dominante, los que desarrollan contenidos se lo ofrecerán a esa plataforma en primer lugar, y a los otros después (y no siempre) --igual que los productos se comercializaban primero para Windows y después para Mac (y no siempre), una vez que estaba claro que el mercado del PC se estaba decantando por Wintel. No puede sorprender que Apple esté tratando de desenfundar antes que Microsoft lanzando su propio servicio de descarga de medios.

6.1. Importancia de las aplicaciones

Microsoft parece estar invirtiendo en equipar la plataforma de sistemas operativos con mecanismos de TC para conseguir mayores ingresos con sus aplicaciones. Puede ser directamente (cobrando el doble por Office, por ejemplo), o indirectamente (cobrando un porcentaje de todos los contenidos comprados a través de Media Player, por ejemplo). Para las empresas de la competencia, todo dependerá de lo difícil que les resulte que sus contenidos interactúen con las aplicaciones y los contenidos de Microsoft. Y, por supuesto, a Microsoft le interesa que sea lo más difícil posible.

Supongamos que hay un servicio de suscripción a música popular que utiliza Media Player, y que Media Player ahora requiere una plataforma TC: el resultado es que los suscriptores tendrán que pasarse a una plataforma TC si quieren seguir accediendo a la música que ya han almacenado. Una vez que se haya extendido la utilización de una aplicación TC, con muchos usuarios cautivos en ella, se pueden implementar mecanismos para exigir una licencia, que pueden ser tan difíciles de evadir como difícil es sortear la tecnología. A continuación puede ponerse en práctica el modelo empresarial, que introdujo Nintendo y después siguieron otros fabricantes de consolas, en el que un software caro subvenciona a un hardware barato. El sistema operativo TC se convertiría entonces en un componente necesario y subvencionado cuya función real es maximizar los ingresos con productos de alto precio como Office, juegos y alquiler de contenidos. Si muchas empresas imponen controles obligatorios de acceso para e-correo con Windows 2003, y estos controles de acceso antes o después exigen una plataforma TC, los usuarios no tendrán más remedio que pasarse a ella. De hecho probablemente tengan menos opciones que los suscriptores a música, porque éstos siempre tendrán la alternativa de comprar nuevos CD, como hicieron cuando el CD sustituyó al vinilo. Pero si muchos archivos corporativos y oficiales están protegidos por claves criptográficas, las empresas no tendrán más opción que seguir los mecanismos que protegen y controlan estas claves.

6.2. Costes del cambio y clientes cautivos

En los últimos años las empresas han tenido en cuenta los costes de cambiar de operador a la hora de valorar los

productos y servicios de información. En industrias que necesitan tener cautivos a sus clientes --como la industria del software-- el valor neto de la base de clientes de una compañía es igual al monto total de los costes que sus clientes tendrían que pagar para cambiar a otra empresa de la competencia [22]. Si fuese mayor, a la competencia le compensaría sobornar a esos clientes para conseguirlos. Si fuese menor, la empresa podría sencillamente subir los precios.

Un efecto de TC es aumentar enormemente la posibilidad de tener clientes cautivos. Supongamos, por ejemplo, que el Director de Informática de una empresa quiere dejar de comprar Office y pasar a OpenOffice, con una plataforma GNU/Linux. En la actualidad tiene que afrontar los costes de volver a formar al personal, los costes de instalar el nuevo software y el coste de convertir los archivos o documentos ya existentes. También habrá costes permanentes por incompatibilidades ocasionales. En la actualidad, la teoría económica indica que los costes serían muy parecidos a los que hay que pagar por Office.

Sin embargo, TC dificultaría la conversión de archivos desde formato Office a cualquier otro, y los costes se dispararían [24]. Quizá ni siquiera exista el procedimiento o el mecanismo para exportar contenido TC a una plataforma que no sea de TC, incluso aunque el propietario del contenido lo permita. Y aunque existan los medios para esa exportación, lo más probable es que no sean suficientes por sí mismos si se generalizan los mecanismos obligatorios de control de acceso impuestos por TC. La razón es que muchos de los datos almacenados en los archivos de la compañía aparecerían como propiedad de terceros. Un bufete de abogados, por ejemplo, puede recibir documentos confidenciales de un cliente, dirigidos a la atención de un número especificado de personas.

El bufete puede exigir el derecho a acceder a esos documentos durante seis años, por si tienen que defenderse de acusaciones de malas prácticas profesionales. Todo eso constaría codificado en los atributos de gestión de derechos del documento y se cumpliría gracias a los mecanismos TC. Las reglas de acceso sólo las podría contravenir el propietario del documento, es decir, su creador. Si el bufete de abogados quiere migrar de Office y Windows a OpenOffice y una futura plataforma TC/Linux, tendrían que obtener el permiso de sus clientes para exportar todos los documentos protegidos. Con el tiempo cualquier bufete puede llegar a tener miles de relaciones profesionales y algunas seguro que se tuercen; incluso dando por hecho que fuesen aceptables la logística y la política de pedir permiso a todos los clientes para exportar los documentos, algunos seguro que negaban ese permiso, por diferentes razones. El resultado es que el bufete se vería obligado a mantener un entorno TC/Windows, además del nuevo.

Hay otros efectos suaves, además de los duros. Por ejemplo, la controversia que se genere alrededor de TC puede aumentar la incertidumbre, que a su vez puede llevar a empresas y consumidores a pensar que *«más vale malo conocido que bueno por conocer»*. El resultado puede ser un aumento en los costes de cambio que pueden superar incluso los derivados de la tecnología. (Los lectores de más edad recordarán la controversia que se creó con el elemento de «miedo, incertidumbre y duda», FUD en inglés, que IBM introducía en su

marketing, cuando era IBM --y no Microsoft-- la que se llevaba el gato al agua).

6.3. Temas antimonopolio

Existen, por lo tanto, muchas posibilidades de que TC se extienda utilizando el efecto red, y de que las principales aplicaciones de TC --una vez que dominen en un determinado sector-- estén, en la práctica, completamente a salvo de los competidores. Esto arrojará nueva luz sobre los conocidos argumentos esgrimidos en los casos antimonopolio de la industria de la información. Muchos economistas de las industrias de la información han aceptado la competencia 'por el mercado' diciendo que es igual de justa que la competencia 'dentro del mercado', especialmente por el carácter volátil de la industria y porque cada pocos años todos tienen posibilidades dado que el progreso socava las normas establecidas y se reinventan sectores completos de la industria. Pero este argumento habrá que revisarlo si las enormes --y crecientes-- cantidades de datos de aplicaciones que las compañías y los particulares almacenan pueden quedar cautivos de unas compañías que, en la práctica, están a salvo de los competidores. En cualquier caso, el incentivo para Microsoft está claro: el valor de la compañía es equivalente a los costes directos e indirectos que a las empresas les suponga cambiar a la competencia; si ese cambio es el doble de difícil, el valor del software de Microsoft será también doble.

Hay también otros temas. Varian ya ha apuntado que TC puede reducir la innovación porque limita las posibilidades técnicas de modificar los productos existentes [9], y las cosas irán a peor cuando los datos estén cautivos. En la actualidad, muchas empresas recién creadas consiguen introducirse porque ofrecen formas adicionales de utilizar las grandes cantidades de datos de las aplicaciones existentes en los formatos más utilizados. Una vez que los propietarios de la aplicación principal se adhieran a TC, tendrán todos los incentivos para cobrar por acceder a esos datos. Esto parece pensado para favorecer a las grandes marcas y perjudicar a las pequeñas y a la competencia, además de dificultar la innovación.

Otros fabricantes de aplicaciones de software se enfrentarán a la amenaza de no poder acceder a los datos de las aplicaciones de otros fabricantes, pero también a la perspectiva de que, si consiguen que su producto arraigue y que muchos clientes lo utilicen para sus datos, podrán utilizar los mecanismos de TC para tener a esos clientes cautivos de una forma más eficaz que la que les permitían los anticuados mecanismos de formatos propietarios o los restrictivos contratos firmados a golpe de clic de ratón. Esto posibilitará una valoración mucho más alta de las compañías, así que muchos vendedores de software se verán muy presionados para que adopten TC y si pierden el tren será difícil cogerlo más adelante.

Algunos sectores concretos de la industria pueden sufrir más las consecuencias. Los vendedores de tarjetas inteligentes, por ejemplo, pueden enfrentarse a la perspectiva de que muchas de las aplicaciones que ellos querían colonizar con sus productos utilizan en su lugar las plataformas TC en PC, PDA y teléfonos móviles. En general, la industria de la seguridad en la información se enfrentará a cambios importantes cuando muchos productos migren a TC o se abandonen.

Es difícil encontrar analogías en otras situaciones históricas, quizá la más parecida se dio hacia 1830, cuando el ferrocarril

sustituyó a los canales en el transporte. Todo el que tuviera un barco podía transportar mercancías por el canal, pero el ferrocarril es un monopolio natural y suscitó ese tipo de críticas en su momento. El ferrocarril, por supuesto, no era en sí un desastre económico, pero sí contribuyó a crear concentraciones de poder económico y casos de competencia desleal que, a su vez, contribuyeron a que ciertos países se dotaran de leyes antimonopolio y a que en otros el ferrocarril pasase a manos públicas.

Es difícil hacer predicciones a largo plazo, pero a corto plazo parece razonable que la implantación de TC produzca efectos económicos: las compañías pequeñas saldrán perjudicadas y las grandes favorecidas; el mercado favorecerá a las empresas ya establecidas y penalizará a las nuevas; y se incrementarán los costes y los riesgos de montar nuevas empresas. Las industrias de informática y de comunicaciones se parecerán más a las de sectores tradicionales, como la automovilística o farmacéutica. Y eso no está muy claro si es bueno o malo.

7. ¿Qué consecuencias tiene la TC para los profesionales informáticos?

Durante muchos años, los ingenieros de seguridad se han quejado de que ni los fabricantes de hardware ni los de software se han preocupado de incorporar mecanismos de seguridad en sus productos. Las primeras investigaciones realizadas sobre economía de la seguridad nos permiten ahora saber el por qué [25]. Muchas empresas informáticas tuvieron que hacer frente a costes fijos altos, costes marginales bajos, altos costes para cambiar y efectos de red, y todo ello se tradujo en industrias con firmas dominantes y con claras ventajas para los que llegaron antes. Resulta vital reducir el plazo entre la creación de un producto y su comercialización y por eso era perfectamente racional la filosofía que adoptó Microsoft en los años 90 de «*lo sacamos el martes, y en la 3ª versión ya habremos corregido los fallos*».

Además, cuando se compite para dominar un mercado en red, las empresas tienen que contar con fabricantes de bienes y servicios complementarios. Los fabricantes de sistemas operativos tienen pocos incentivos para ofrecer complejos mecanismos de control de acceso porque éstos estorban a los que desarrollan las aplicaciones. La relativa poca importancia del usuario final, comparada con la de los fabricantes complementarios, llevó a las empresas a adoptar tecnologías (como PIK) que hacen que los fabricantes pasen a los usuarios finales los costes de seguridad y de administración.

El control de la interfaz de programación de la aplicación es vital para el propietario de la plataforma, así que intenta hacerlo propietario, complicado, extensible y, por lo tanto, sujeto a fallos. Es mucho más importante hacer posible la discriminación por precios que hacer posible la privacidad. Y por último, ante la falta de un amplio conocimiento sobre seguridad, los malos productos echaron del mercado a los buenos. ¿Qué le hizo cambiar tan súbitamente de opinión a Microsoft?

Un cínico diría que el reciente acuerdo antimonopolio al que Microsoft ha llegado con el Departamento de Justicia de los EE.UU. le obliga a compartir información sobre interfaces y sobre protocolos, excepto cuando afecte a la seguridad. Así que la solución es decir que todo lo que hace la compañía tiene

que ver con la seguridad. Microsoft también ha argumentado que la publicidad que se le ha dado a los ataques de varios tipos que ha sufrido la Red actúa como un incentivo para que se produzcan más ataques. Pero parece difícil que un gusano o dos al año justifique un cambio tan brusco de política y de dirección.

Este artículo sostiene que otro factor importante que ha decidido a Microsoft a gastar cifras de nueve dígitos en la seguridad de la información --después de haber desdeñado el tema durante décadas-- es la posibilidad de aumentar el número de clientes cautivos. (Es curioso comprobar que Intel, AMD, IBM y HP también están invirtiendo sumas considerables en TC, aunque no hay a la vista ninguna amenaza antimonopolio).

TC plantea otros muchos temas, desde censura hasta soberanía nacional, pasando por el futuro del movimiento por un software libre y de fuente abierta [2]. Pero para el hombre de negocios con el colmillo retorcido, TC será competencia pura y dura. La pregunta clave es: «¿Cómo podrá Microsoft sacarme dinero con esto?». La respuesta es muy sencilla: «Teniéndote cada vez más cautivo con plataformas Microsoft como Office».

¿Qué podrían hacer los legisladores y los reguladores? En las leyes de patentes podemos encontrar precedentes muy útiles. Durante años, un contrato que ataba de forma ilegal podía invalidar una patente en el Reino Unido: si el dueño de una patente de un proceso para fabricar harina le permite a otro utilizar ese proceso a condición de que le compre a él el grano, el dueño de la patente de hecho la está invalidando. Como poco, se podría sugerir que se invalidara la protección legal que la DMCA (Digital Millennium Copyright Act de los EE.UU) y la EUCD (Directiva europea sobre *copyright* 2001/29/EC) ofrecen a los mecanismos de TC, que se suponen que están a su vez protegiendo los derechos de autor, siempre que se utilizasen para dificultar la competencia --con controles adicionales o teniendo más cautivos a sus clientes.

La sugerencia es que los legisladores vean si la aplicación de TC aumenta o disminuye el saldo positivo de los consumidores. Es la prueba que propondría la jurisprudencia sobre patentes abusivas [26]. Ya que se afirma que TC creará más valor para los clientes y que, evidentemente, creará más valor para los fabricantes, y dada toda la polémica generada sobre los puntos positivos y negativos de la gestión de derechos digitales, quizá la forma más sencilla y más práctica de llegar a una política coherente y sólida sea ver si los clientes han salido favorecidos o perjudicados.

Referencias

- [1] **R.J. Anderson**, *Security Engineering - a Guide to Building Dependable Distributed Systems*, Wiley (2001) ISBN 0-471-38922-6.
 [2] **R.J. Anderson**, «TCPA/Palladium FAQ», en <<http://www.cl.cam.ac.uk/users/rja14/tpa-faq.html>>.
 [3] **M. Magee**, «HP inkjet cartridges have built-in expiry dates - Carly's cunning consumable plan», *The Inquirer*, 29 abril 2003, en <<http://www.theinquirer.net/?article=9220>>.
 [4] «**Ink Cartridges with Built-In Self-Destruct Dates**», Slashdot, en <<http://slashdot.org/articles/03/04/30/1155250.shtml>>.
 [5] «**Computer Chip Usage in Toner Cartridges and Impact on the Aftermarket: Past, Current and Future**», Static Control, Inc., en <<http://www.scc-inc.com/special/oemwarfare/whitepaper/default.htm>>.
 [6] «**Lexmark invokes DMCA in Toner Suit**», Slashdot, en <[<\[slashdot.org/article.pl?sid=03/01/09/1228217&mode=thread&tid=123\]\(http://slashdot.org/article.pl?sid=03/01/09/1228217&mode=thread&tid=123\)>.
 \[7\] «**Prepared Statements and Press Releases**», Static Control, Inc., en <\[http://www.scc-inc.com/special/oemwarfare/lexmark_vs_scc.htm\]\(http://www.scc-inc.com/special/oemwarfare/lexmark_vs_scc.htm\)>.
 \[8\] **M. Broersma**, «Printer makers rapped over refill restrictions», ZDnet 20 diciembre 2002, at <<http://news.zdnet.co.uk/story/0,,t269-s2127877,00.html>>.
 \[9\] **H.R. Varian**, «New Chips Can Keep a Tight Rein on Customers», *New York Times*, July 4 2002, en <<http://www.nytimes.com/2002/07/04/business/04SCEN.html>>.
 \[10\] «**Motorola Announces Availability of New Wireless Phone Batteries for Increased Performance and Safety, Featuring New Hologram Design**», nota de prensa de Motorola, 23 julio 1998, sacada tras ser mencionada en \[2\]; ahora archivada en <\[http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/mototola_battery_auth.html\]\(http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/mototola_battery_auth.html\)>.
 \[11\] **D. Becker**, «Sony loses Australian copyright case», on CNN.com, 26 julio 2002, en <<http://rss.com.com/2100-1040-946640.html?tag=m>>.
 \[12\] **N. Pickler**, «Mechanics Struggle With Diagnostics», AP, 24 junio 2002; antes en radicus.net; sacada tras ser mencionada en \[2\]; ahora archivada en <<http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/car-diagnostics.html>>.
 \[13\] **Trusted Computing Group**, <<http://www.trustedcomputinggroup.org/>>.
 \[14\] **J. Lettice**, «Bad publicity, clashes trigger MS Palladium name change», *The Register*, 27 enero 2003, en <<http://www.theregister.co.uk/content/4/29039.html>>.
 \[15\] **R. Stallman**, «Can you trust your computer?», en <<http://newsforge.com/newsforge/02/10/21/1449250.shtml?tid=19>>.
 \[16\] **Microsoft Corp.**, «Windows Server 2003», 20 feb 2003, en <<http://www.microsoft.com/windowsserver2003/mm>>.
 \[17\] **J. Manferdelli**, «An Open and Interoperable Foundation for Secure Computing», en *Hoja informativa* de Windows Trusted Platform Technologies, marzo 2003.
 \[18\] **A. Huang**, «Keeping Secrets in Hardware: the Microsoft Xbox Case Study», 26 may 2002, en <<http://web.mit.edu/bunnie/www/proj/anatak/AIM-2002-008.pdf>>.
 \[19\] **P. Thurrott**, «Microsoft's Secret Plan to Secure the PC», WinInfo, 23 junio 2002, en <<http://www.wininformant.com/Articles/Index.cfm?ArticleID=25681>>.
 \[20\] **S. Lewis**, «How Much is Stronger DRM Worth?» en *Second International Workshop on Economics and Information Security*, en <<http://www.cpppe.umd.edu/rhsmith3/index.html>>.
 \[21\] **S.E. Schechter, R.A. Greenstadt, M.D. Smith**, «Trusted Computing, Peer-To-Peer Distribution, and the Economics of Pirated Entertainment», en *Second International Workshop on Economics and Information Security*, en <<http://www.cpppe.umd.edu/rhsmith3/index.html>>.
 \[22\] **C. Shapiro, H. Varian**, «Information Rules», Harvard Business School Press \(1998\), ISBN 0-87584-863-X.
 \[23\] **A. Gawer, M.A. Cusumano**, «Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation», Harvard Business School Press \(2002\), ISBN 1-57851-514-9.
 \[24\] **J. Brockmeier**, «The Ultimate Lock-In», Yahoo News. 12 mar 2003, en <\[http://story.news.yahoo.com/news?tmpl=story2&cid=75&ncid=738&e=9&u=/nf/20030312/tc_nf/20982\]\(http://story.news.yahoo.com/news?tmpl=story2&cid=75&ncid=738&e=9&u=/nf/20030312/tc_nf/20982\)>.
 \[25\] **R.J. Anderson**, «Why Information Security is Hard - An Economic Perspective», en *Proceedings of the Seventeenth Computer Security Applications Conference* IEEE Computer Society Press \(2001\), ISBN 0-7695-1405-7, pp 358-365, en <<http://www.cl.cam.ac.uk/ftp/users/rja14/econ.pdf>>.
 \[26\] **C. Shapiro**, «Antitrust Limits to Patent Settlements», preprint, en <<http://faculty.haas.berkeley.edu/shapiro/settle.pdf>>](http://</p>
</div>
<div data-bbox=)

Lengua e Informàtica

Jordi Mas i Hernàndez
Softcatalà

<jmas@softcatala.org>

Resumen: *el software libre es un tipo de software que da libertad a sus usuarios. Con el advenimiento de Internet, el software libre se ha consolidado como alternativa, técnicamente viable y económicamente sostenible, al software de propiedad. Lenguas como el bretón, el gallego, el gaélico o el catalán han tenido un desarrollo muy escaso en el mundo del software propietario por las limitaciones que impone. En contraposición, en el mundo del software libre estas lenguas se han desarrollado con un éxito notable. Proyectos de software libre tan importantes como el navegador Mozilla, el entorno GNOME o KDE o el sistema GNU/Linux tienen traducciones completas o parciales en todas estas lenguas. El software libre representa una oportunidad sin precedentes para el desarrollo de lenguas minoritarias en las nuevas tecnologías gracias a las libertades que nos garantiza.*

Palabras clave: *lenguas minoritarias, localización, software libre, traducción.*

1. Introducción

El software¹ libre es un tipo de software que da libertad a sus usuarios. No sólo libertad para ejecutarlo y utilizarlo, sino también para muchas otras cosas: libertad para hacer copias, para distribuirlo y para estudiarlo (lo que implica tener siempre acceso al código fuente)². Además, cualquier usuario puede mejorar el software libre y puede hacer públicas estas mejoras (con el código fuente correspondiente), de tal manera que todo el mundo pueda beneficiarse de ello.

Actualmente existen empresas que producen y venden software propietario (contrapuesto al libre); de hecho, es este tipo de software el que utilizan la mayoría de usuarios hoy en día. El software propietario está sujeto a diversas limitaciones; de entrada, normalmente hay que pagar su licencia, se está sujeto a las posibles limitaciones técnicas de estos programas y a las que su licencia impone, con las consiguientes posibles incompatibilidades entre programas elaborados por empresas diferentes que trabajan con código cerrado; así pues, se está en cierta medida atado a la empresa que lo fabrica (por ejemplo, para traducirlo, para las actualizaciones, para complementos, etc.).

El software libre, en cambio, no está sujeto a estas limitaciones de mejora, ya que su licencia permite de manera explícita que cualquier usuario añada las mejoras (o adaptaciones) que

El software libre y las lenguas minoritarias: una oportunidad impagable

quiera, y con total libertad. Está disponible en forma de código fuente y, por lo tanto, todo el mundo puede acceder a él y lo puede utilizar como quiera. Éste es el espíritu del software libre: que todo el mundo pueda contribuir a mejorarlo sin tener que pagar ni pedir permiso a nadie, y que las mejoras se pongan a disposición de todo el mundo.

Con el advenimiento de Internet, el software libre se ha consolidado como alternativa, técnicamente viable y económicamente sostenible, al software propietario. Contrariamente a lo que a menudo se piensa, grandes empresas informáticas como AOL, IBM, Sun y Apple ofrecen apoyo financiero y comercial al software libre. Por ejemplo, hoy en día IBM facilita el uso de GNU/Linux en sus *mainframes* (grandes ordenadores), y las nuevas versiones del sistema operativo de los ordenadores Apple (MacOS X) están basadas en software libre (FreeBSD).

El software libre no excluye necesariamente el uso de software propietario (uno puede continuar usándolo si lo desea); al contrario, puede integrarse en él o bien complementarlo. Y por supuesto, puede efectivamente reemplazarlo.

Entre el software propietario más popular podemos encontrar conocidos programas que utilizan la mayoría de usuarios hoy en día, desde el Microsoft Office o el Microsoft Windows hasta el Acrobat Reader o el Internet Explorer.

2. Al principio todo era libre

En los inicios de la informática, prácticamente todo el software era libre. Los primeros grandes ordenadores de las universidades en los años sesenta y setenta venían del fabricante con un conjunto de software, su código fuente, y sin ninguna limitación para modificarlo o mejorarlo. De este modo, el software estaba sometido a un proceso de mejora continua con las diferentes aportaciones que hacía cada usuario. Eran comunes los casos en que los usuarios creaban programas propios, o mejoraban los que venían de serie con estos sistemas, y los enviaban al fabricante del hardware para que, si lo consideraba adecuado, los distribuyera en la próxima versión de su sistema.

Este modelo se empezó a romper hacia principios de los años setenta, cuando las empresas observaron que el software podía venderse y empezaron a cerrar el acceso al código fuente. Así, poco a poco, tener acceso al código fuente del

programa y otras libertades, que hasta entonces eran comunes, fueron desapareciendo progresivamente y, con ellas, las libertades de los usuarios.

En 1983, Richard Stallman, consciente de la pérdida de libertades que se iba experimentando en el mundo del software, abandonó su trabajo en los laboratorios de Inteligencia Artificial del MIT (Massachusetts Institute of Technology) para empezar un proyecto dirigido a construir un sistema operativo libre. Este proyecto, pocos meses después, lo llevó a crear la Free Software Foundation (FSF), una organización que ha sido la principal difusora de la filosofía del software libre durante los últimos veinte años y que organizó el desarrollo del principal software libre de sistemas.

3. El impulso: Internet y el GNU/Linux

El verdadero impulso del software libre llegó con la combinación de dos factores: la aparición del sistema operativo Linux y la extensión del uso de Internet.

El sistema operativo Linux nació de la mano de un joven estudiante finlandés de veintiún años llamado Linus Torvalds. En 1991 publicó el núcleo del sistema y, al mismo tiempo, pidió la ayuda de voluntarios que quisieran contribuir, de forma desinteresada, en el desarrollo y mejora del sistema. El GNU/Linux es el resultado del trabajo de toda una comunidad que se fue vertebrando en torno a este sistema, la cual, mediante Internet, fue mejorándolo para satisfacer sus necesidades.

En este sentido, la popularización del acceso a Internet fue un factor fundamental para que esta comunidad pudiera llegar a nuevos desarrolladores y para conseguir que se fueran creando nuevos proyectos. La posibilidad de poder ampliar y mejorar el sistema existente atrajo a miles de programadores que, con sus esfuerzos individuales, contribuyeron al establecimiento y la solidificación del movimiento.

Ahora bien, la comunidad que trabaja con y por el software libre no está formada sólo por programadores; también incluye los usuarios avanzados de las aplicaciones que se constituyen en grupos para dar conferencias y apoyo a los usuarios. También los hay que participan en la promoción del movimiento, o que escriben documentación para facilitar el uso de las aplicaciones.

4. Mitos del software libre

Desgraciadamente, hay algunos conceptos erróneos en torno al software libre y ciertos mitos que son falsos. Aclaremos algunos de ellos:

- *El software gratuito es libre*: un programa, por el simple hecho de ser gratuito, no es ni mucho menos libre. Por ejemplo, el Internet Explorer de Microsoft es un programa gratuito, pero no es libre, ya que no da a sus usuarios la posibilidad de estudiarlo (incluyendo el acceso a su código fuente), ni de mejorarlo, ni de hacer públicas estas mejoras con el código fuente correspondiente de manera que todo el mundo se pueda beneficiar de ello. El Internet Explorer es un programa de propiedad gratuita.
- *No hay software libre comercial*: cualquier software libre

puede venderse siempre que se respeten las libertades originales que lo definen. Por ejemplo, la empresa Mandrake, que produce una versión de GNU/Linux bastante popular, vende distribuciones de GNU/Linux a un precio razonable, y son software libre porque conservan intactas todas las libertades que definen el software libre.

- *El software libre sólo está disponible para GNU/Linux*: GNU/Linux es uno de los buques insignia del movimiento del software libre, ya que es un sistema operativo completamente libre, aunque también es verdad que hay muchos más programas libres, como el Open Office, el Mozilla, el Abiword, el GIMP o muchos otros que se encuentran disponibles para multitud de sistemas, incluyendo el Windows y en algunos casos el Mac. El software libre es un concepto que no está ligado a ningún sistema; de hecho, también se aplica para documentación y todo tipo de creaciones artísticas.

- *No tiene apoyo para empresas*: hoy en día, empresas como IBM, Sun, Compaq, Dell, RedHat, Suse, Mandrake y muchas otras ofrecen apoyo y servicios profesionales de sistemas basados en software libre a usuarios, empresas e instituciones.

- *El software libre no es de calidad*: hay que recordar que la gran mayoría de proyectos de software libre se hacen desde el voluntariado, y se tiende a compararlos con proyectos parecidos desarrollados por empresas con unos recursos de ingeniería y desarrollo muy grandes. Cada proyecto de software libre, del mismo modo que cada software de propiedad, tiene que valorarse por separado. Hoy en día, proyectos como el Mozilla, el Open Office, el Apache y otros están produciendo software de igual o mayor calidad que los productos equivalentes en el mundo del software de propiedad.

5. Las oportunidades para lenguas minoritarias en el mundo del software libre

Lenguas como el bretón, el gallego, el gaélico o el catalán han tenido un desarrollo muy escaso en el mundo del software propietario. Ello es debido básicamente a que el proceso de localización³ de este tipo de programas sólo puede llevarse a cabo por el fabricante del programa, que es quien determina, bajo criterios comerciales e incluso alguna vez políticos, que estos tipos de traducciones no son viables.

Por el contrario, en el mundo del software libre estas lenguas se han desarrollado con un éxito notable. Proyectos de software libre tan importantes como el navegador Mozilla, el entorno GNOME, el entorno KDE, o el sistema GNU/Linux poseen traducciones completas o parciales en todas estas lenguas⁴ y en muchas otras. No hay que pedir permiso para traducirlos ni hay que justificar ninguna viabilidad económica: sólo es preciso traducirlos o mejorar las traducciones ya existentes.

El software libre representa una oportunidad sin precedentes para el desarrollo de lenguas minoritarias, como por ejemplo el catalán, en las nuevas tecnologías gracias a las libertades que nos garantiza. A continuación, a modo de ejemplo, comentaremos algunos casos o algunas situaciones en que el software libre presenta ventajas importantes:

- *Cualquier persona o institución puede traducir y adaptar un software libre a cualquier lengua*: esto representa un gran paso en comparación con el software de propiedad, en

el que sólo la empresa que lo ha creado tiene los derechos para realizar su traducción o para cederla a terceros, y en muchas ocasiones, como nos demuestra la experiencia, pidiendo fuertes cantidades de dinero para darnos el derecho de efectuar su traducción.

- *Se garantiza la continuidad de las nuevas versiones:* si una persona o institución realiza una traducción de un programa libre, la traducción de este programa queda sometida a los mismos derechos que el propio programa, de tal manera que también pasa a ser libre. Ello nos garantiza que siempre tendremos acceso a la traducción, y, si alguna vez la persona o institución desiste de continuar la traducción, cualquier otra persona puede seguir su labor.
- *La lengua en que adaptamos el programa se convierte en una lengua del mismo orden que las demás:* si, por ejemplo, queremos localizar un programa libre al catalán, y éste no dispone para el catalán de las mismas características que para el resto, podremos desarrollar las herramientas y los elementos necesarios para equiparar la lengua a la que adaptamos el programa a las demás. Por ejemplo, si un programa libre no dispone de un sistema de corrección ortográfica, podemos desarrollar nuestro propio corrector o bien adaptar alguno de los correctores existentes en el mundo del software libre. Éste es el caso de Softcatalà, que desarrolló el corrector ortográfico en catalán para los proyectos Open Office, Abiword y Mozilla para hacer que trabajar en catalán con estos programas fuera equivalente a trabajar en otras lenguas.
- *El software libre suma:* cada vez que se crea un nuevo recurso lingüístico bajo una licencia libre, este recurso se pone a disposición de todo el mundo y todo el mundo puede beneficiarse de él. Cualquier persona puede acceder a las traducciones en cualquier lengua de cualquier proyecto de software libre y crear un glosario o una memoria de traducción⁵ que ayude a traducir un nuevo programa o mejorar los ya existentes.

En el mundo del software propietario, y a causa de su modelo cerrado, todas las traducciones son un oasis y todos los recursos lingüísticos acostumbran a ser específicos para aquel programa y, por lo tanto, difícilmente reutilizables en otros proyectos.

- *Coste mínimo para el usuario:* el software libre siempre tiene un coste mínimo, que suele ser el coste de la distribución, el tiempo de bajarlo desde Internet o el importe de un CD-ROM de bajo precio. Tras localizar un programa propietario a una lengua minoritaria, en la gran mayoría de los casos los usuarios deben adquirirlo.

El software libre, pues, es prácticamente gratuito, y hace posible una verdadera normalización lingüística de las lenguas minoritarias, ya que permite que todo el mundo pueda obtener una copia. En este sentido, podemos comparar el número de copias vendidas de las diferentes ediciones de Windows en catalán (por debajo de las 15.000 unidades) con las más de 250.000 copias que, por ejemplo, se han distribuido del Mozilla en catalán y los costes de desarrollo de ambos proyectos.

6. Un caso de estudio: Yahoo en catalán

El uso de software libre tiene implicaciones y ventajas que están fuertemente relacionadas no sólo con la lengua, sino

también con el desarrollo local, la liberación de monopolios y la independencia del proveedor, como seguidamente veremos en el análisis del caso Yahoo.

Hace aproximadamente un año se firmó un acuerdo entre Yahoo Ibérica y los gobiernos catalán, andorrano y balear para la creación de una versión en catalán del conocido directorio Yahoo, por un importe total de 540.000 euros (más de noventa millones de pesetas).

Por otro lado, hace más de cuatro años que existe el Open Directory, basado en la filosofía del software libre, por lo que cualquiera puede utilizar sus contenidos actuales y mejorarlos. Imaginémos por un momento que se hubiera firmado un acuerdo con una empresa en Cataluña para desarrollar el servicio de directorio a partir de los contenidos del Open Directory y que se hubieran hecho en él mejoras y ampliaciones. Como el Open Directory está liberado bajo la licencia de software libre, todas las mejoras que se realizaran en él también serían libres, ya que así lo exige y garantiza esta licencia.

Éstas son algunas de las ventajas que se habrían conseguido con la realización del servicio principal de Yahoo, es decir, el directorio, mediante una solución basada en software libre:

- *Derechos sobre el directorio:* los derechos de todos los recursos desarrollados bajo este acuerdo pertenecen a la empresa Yahoo. De esta manera, si alguna vez quiebra o se deja de prestar el servicio, nadie podrá utilizar todos los sitios web reunidos, ni su descripción, que forman hoy en día el Yahoo en catalán.

La alternativa libre habría sido partir de la base de la sección catalana actual del Open Directory Project, añadirle las entradas que faltan y hacer una corrección general de la misma. Por el hecho de tratarse de un recurso libre, cualquiera podría utilizarlo y se aseguraría su continuidad más allá de la empresa proveedora.

- *Desarrollo local:* Yahoo Ibérica, como su nombre indica, es una empresa estadounidense con sede en Madrid. Con la subvención de este tipo de empresas se da la espalda a las empresas instaladas en nuestro país, como Vilaweb, propietaria del directorio Nosaltres.com y con experiencia en este ámbito desde 1994. Mediante un acuerdo con una empresa local para realizar esta mejora del Open Directory se habría fomentado el tejido de empresas catalanas que se dedican a las nuevas tecnologías.

- *Derechos libres contra monopolio:* apostar por Yahoo en catalán fomenta la creación de un monopolio, ya que sólo ellos tienen derechos sobre el directorio. En cambio, apostando por el Open Directory nos encontramos con derechos libres y cualquiera puede hacerle la competencia con unos recursos que, en gran parte, se han creado con dinero público.

Por ejemplo, hoy en día podemos acceder al proyecto Open Directory desde <<http://www.aprop.net>>, un sitio independiente, <<http://www.eresmas.com>> y <<http://dmoz.org>>. Todo el mundo puede integrar este directorio como quiera, mejorarlo y retornar sus mejoras a la comunidad.

- *Independencia del proveedor:* el hecho de que el Open Directory sea libre nos garantiza que podamos contratar la empresa que deseemos para mejorarlo o ampliarlo; si decidimos finalizar este contrato, como todos los recursos son libres, el siguiente contrato lo podemos firmar con otra empresa. Con Yahoo en catalán, sólo Yahoo puede conti-

nuar el trabajo y estamos ligados a un único proveedor. Como podemos observar, las ventajas son evidentes. Cuando se piensa en software libre hay que pensar más allá de los beneficios que pueda acarrear a la lengua y tener en cuenta los no menos importantes beneficios que representa para el desarrollo local, porque fomenta la competencia, rompe monopolios y nos desliga de un único proveedor.

7. ¿Qué pueden hacer las administraciones e instituciones ante esta oportunidad?

- *Utilizar software libre*: las instituciones y administraciones públicas deben implicarse en el uso del software libre, especialmente cuando está disponible en catalán; y, si no lo está, aún mejor, porque tienen la posibilidad de utilizarlo y traducirlo a su idioma, con lo que se crea un nuevo recurso lingüístico disponible. Para más detalles sobre este planteamiento, véase el manifiesto de la campaña para el uso del software libre en la Administración pública que Softcatalà impulsó⁶ hace unos meses y la proposición de ley para el uso del software libre de Esquerra Republicana de Catalunya⁷.
- *Liberar recursos bajo licencias libres*: las universidades e instituciones que desarrollan herramientas de investigación lingüística o traducciones, en muchos casos con dinero público, deben aprender a romper el celo de guardar su investigación y entender que el conocimiento humano sólo ha progresado cuando se ha fomentado el libre intercambio de ideas. Mantener de puertas adentro las tecnologías de la lengua no beneficia en nada el desarrollo de las lenguas.
- *Difundirlo (con implicación)*: recomendarlo y distribuirlo, para que así las entidades, la población civil, las empresas y los organismos, entre otros, lo tengan al alcance.

8. Hacia un modelo de sociedad basada en el software libre

Bajo el prisma de muchos políticos, incluso europeos, parece que el problema principal de nuestra sociedad es garantizar el acceso a Internet a toda la población, especialmente el acceso con banda ancha a poblaciones alejadas de los grandes núcleos urbanos, un problema tan grave en nuestro país.

El ancho de banda anda en boca de todos como un problema que puede crear la tan temida fractura digital. Pero, ¿cuál es el coste del software necesario para que esta fractura no se produzca? El software que utilizan la mayoría de los usuarios hoy en día para acceder a las nuevas tecnologías es Microsoft Windows y Microsoft Office, que tienen un coste superior a los 700 euros por usuario. En España los índices de piratería son los más altos de Europa. ¿Qué tipo de sociedad de la información estamos creando? Una sociedad de la información basada básicamente en la piratería porque la mayoría de los usuarios no pueden asumir los costes desorbitados del software propietario. Y una vez tengamos accesos a Internet de alta velocidad, ¿para qué los utilizaremos? ¿Para acceder a contenidos y utilizar herramientas informáticas en otras lenguas? Tendremos quizás las autopistas, pero nos faltarán los contenidos y las herramientas informáticas propias y en la propia lengua para poder trabajar.

El software libre nos da la oportunidad de salvar muchos de estos obstáculos, ahorrando los costes del software propietario allí donde sea posible (es decir, allí donde haya alternativas libres) y reinvirtiéndolos en formación para alfabetizar

digitalmente a la población y en la creación de las herramientas libres necesarias para poder trabajar con nuestra lengua, tal como lo hacen las sociedades más avanzadas.

9. Conclusión

El software libre representa una oportunidad sin precedentes para el desarrollo de lenguas minoritarias como el catalán en el mundo de las nuevas tecnologías, el desarrollo local, y también una mejora en la alfabetización digital de la población. Aprovechar esta oportunidad depende en gran medida de las administraciones, las universidades y las instituciones.

Notas

¹ Conjunto de aplicaciones informáticas que permiten usar un sistema informático.

² Conjunto de instrucciones que forman un programa informático.

³ <<http://developer.gnome.org/projects/gtp/status/gnome-2.0-core/>>, <http://www.mozilla.org/projects/110n/mlp_status.html>.

⁴ *Localización* es la traducción y adaptación de una aplicación a una lengua y a su entorno cultural correspondiente.

⁵ Memoria de traducción: base de datos de referencia con cada unidad de traducción, de manera que toda la información que esta contenida puede ser reutilizada en traducciones posteriores.

⁶ <<http://www.softcatala.org/admpub/>>.

⁷ <<http://www.softcatala.org/admpub/proposta.htm>>.

Lenguajes informáticos

Pascual Julián Iranzo

Departamento de Informática, Universidad de Castilla - La Mancha; Grupo ELP (Extensiones de la Programación Lógica) del DSIC, Universidad Politécnica de Valencia

<Pascual.Julian@uclm.es>

Resumen: la Evaluación Parcial (EP) es una técnica de transformación automática de programas que persigue, entre otros objetivos, la optimización de programas con respecto a ciertos datos de entrada; de ahí que también haya recibido el nombre de especialización de programas. Este artículo trata de presentar la EP a un público no experto que desarrolla su actividad en otros campos de la informática. Con este fin se traza una panorámica general del área y de sus técnicas más importantes. Si bien se define con precisión qué se entiende por evaluación parcial y se ilustran sus métodos mediante diversos ejemplos sencillos, este trabajo se ocupa, principalmente, de los objetivos de la evaluación parcial y de sus aplicaciones en diversas áreas científicas y tecnológicas, incidiendo en los beneficios que pueden extraerse de ella.

Palabras clave: compiladores e intérpretes, especialización de programas, EP, evaluación parcial, generación automática de programas, mantenimiento del software, optimización de programas, transformación de programas, verificación de programas.

1. Introducción

La transformación de programas es un método para derivar programas correctos y eficientes partiendo de una especificación ingenua y más ineficiente del problema. Esto es, dado un programa P , se trata de generar un programa P' que resuelve el mismo problema y equivale semánticamente a P , pero que goza de mejor comportamiento respecto a cierto criterio de evaluación. En la literatura se ha propuesto una gran variedad de transformaciones para mejorar el código. Una de las mejor estudiadas es la *Evaluación Parcial* (EP) de programas, que ofrece un marco unificado para la investigación acerca de los procesadores de lenguajes, en particular, compiladores e intérpretes [16]. La EP es una técnica de transformación de programas que consiste en la especialización de programas respecto a ciertos datos de entrada, conocidos en tiempo de compilación, por lo que también se la denomina *especialización* de programas.

En general, las técnicas de EP incluyen algún criterio de parada para garantizar la terminación del proceso de la transformación. La EP es, por tanto, una técnica de transformación automática, lo cual la distingue de otras técnicas de transformación de programas tradicionales [8] [19]. La EP ha sido aplicada a los lenguajes imperativos tanto como a los declarativos y a una gran variedad de problemas concretos. Una panorámica general sobre este campo y su área de aplicación se presenta en [16]. Un breve pero excelente tutorial sobre EP (si bien centrado en la especialización de programas imperativos y funcionales) es [10]. Otros trabajos que examinan aspectos concretos del área son [2] [14].

Evaluación parcial de programas y sus aplicaciones

La intención al escribir este artículo ha sido mostrar los fundamentos teóricos de la EP, y algunas de sus aplicaciones, en un formato asequible a un lector no especializado. Este artículo se ha estructurado como sigue.

En la sección 2, se describe con precisión el concepto de EP y se introducen algunas de sus principales técnicas. En la sección 3 se presentan algunos de los objetivos más importantes perseguidos por la EP, como son: aumentar la eficiencia de los programas; aumentar la productividad en el desarrollo de las aplicaciones; mejorar la reusabilidad y modularidad del software y conseguir evaluadores parciales autoaplicables (esto es, evaluadores parciales capaces de especializarse a ellos mismos). En la sección 4 se exponen varias de las aplicaciones de la EP en áreas industriales y en otros campos de investigación más básica. Esta sección es una primera aproximación que sólo pretende llamar la atención del lector sobre cada uno de esos campos de aplicación, ya que un tratamiento más técnico está fuera del alcance del artículo.

Posteriormente, el lector puede completar su formación atendiendo a sus intereses; para ello se incluye un anexo en el que se facilitan una serie de direcciones de sitios web donde encontrar información general sobre el campo de la EP, grupos de investigación que desarrollan una actividad intensa en este área y evaluadores parciales desarrollados para diferentes lenguajes (tanto declarativos como imperativos). Finalmente se presentan las conclusiones, seguidas del anexo antes citado.

2. La Evaluación Parcial

La idea de especializar funciones con respecto a uno o varios de sus argumentos es vieja en el campo de la teoría de funciones recursivas, donde recibe el nombre de *proyección*. Esta posibilidad está implícita en el teorema s - m - n de Kleene que establece que, dada una función computable $f \in F^{n+m}$, cuyo tipo es $f: S^{n+m} \rightarrow S$, se cumple que la función $f_{e_1, \dots, e_n}: S^m \rightarrow S$, tal que

$$f_{e_1, \dots, e_n}(d_1, \dots, d_m) = f(e_1, \dots, e_n, d_1, \dots, d_m),$$

es computable.

Ejemplo 1: sea N el conjunto de los números naturales. La función *suma*: $N^2 \rightarrow N$, definida como $\text{suma}(x, y) = x + y$, puede especializarse para un valor conocido de x , por ejemplo 2, transformándose en una función $\text{suma}_2: N \rightarrow N$, tal que $\text{suma}_2(y) = 2 + y$.

Se debe notar que en el ejemplo 1 los valores conocidos simplemente se substituyen por los correspondientes parámetros formales de la función; no se realiza ningún cómputo. Además, en este proceso, que recuerda a la

curricación de funciones en los lenguajes funcionales, el objetivo de lograr una mayor eficiencia está ausente. Los problemas de eficiencia eran irrelevantes para las investigaciones efectuadas por Kleene, centradas en establecer los límites entre lo que es o no es computable. El hecho es que la técnica propuesta por Kleene produce funciones especializadas cuyo comportamiento operacional, en algunos casos, es ‘peor’ que el de las originales. Por otro lado, la evaluación parcial trabaja con programas (textos) más bien que con funciones matemáticas. Por consiguiente, la EP es una técnica que va más allá de la simple proyección de funciones matemáticas.

La EP establece como ejecutar un programa cuando sólo conocemos parte de sus datos de entrada. De forma más precisa, dado un programa P y parte de sus datos de entrada $in1$, el objetivo de la EP es construir un nuevo programa P_{in1} que cuando recibe el resto de los datos de entrada $in2$, computa el mismo resultado que produce P al procesar toda su entrada $in1 + in2$. Esto último asegura la corrección de la transformación efectuada.

El programa que realiza el proceso de EP recibe el nombre de *evaluador parcial* y el resultado de la EP, el programa P_{in1} , se denomina *programa especializado*, *programa evaluado parcialmente* o también *programa residual*. La idea que se esconde detrás del proceso de EP consiste en: i) realizar tantos cálculos como sea posible en tiempo de EP, haciendo uso de los datos de entrada conocidos $in1$, también denominados datos *estáticos* (por contra posición con los datos de entrada desconocidos $in2$, que son denominados *datos dinámicos*, y que sólo se conocen en tiempo de ejecución del programa residual); ii) generar código relacionado con aquellos cálculos que no puedan realizarse por depender de los datos de entrada desconocidos. Así pues, un evaluador parcial realiza una mezcla de acciones de cómputo y de generación de código; esta es la razón por la que Ershov denominó a la EP *computación mixta* (*mixed computation*).

Cuando se realizan cálculos en tiempo de EP, haciendo uso de los datos de entrada conocidos, decimos que hay *propagación de la información*. El objetivo de la EP es obtener la mejor de las especializaciones posibles maximizando la propagación de la información. Se espera que el programa resultante pueda ejecutarse de forma más eficiente ya que, usando el conjunto de datos (parcialmente) conocidos, es posible evitar algunas computaciones (en tiempo de ejecución) que se realizarán (una única vez) durante el proceso de EP. Para cumplir estos fines, la EP utiliza, además de la computación simbólica, algunas técnicas bien conocidas provenientes de la transformación de programas (funcionales) [8], procurando su automatización:

1. **Definición:** permite la introducción de funciones nuevas o la extensión de las existentes.
2. **Instanciación:** permite especializar una función asignando datos de entrada conocidos a sus argumentos.
3. **Desplegado** (*unfolding*): permite el reemplazamiento de una llamada a función por su respectiva definición.
4. **Plegado** (*folding*): como su nombre indica, es la transformación inversa del desplegado, es decir, el reemplazamiento de cierto fragmento del código por la correspondiente llamada a función.

La aplicación reiterada de cada una de estas reglas de transformación supone la introducción de nuevas instrucciones de programa y da lugar a una secuencia de

transformación de la que, finalmente, se extraerá el programa transformado. Una estrategia que reduce el alto grado de indeterminismo existente en la aplicación de las reglas de la transformación, es la presentada en el siguiente esquema básico de transformación:

1. **Repetir** hasta que convenga:
 - buscar una **definición**, e
 - **instanciar** la definición para permitir,
 - pasos de **desplegado** en diversos puntos, seguidos por,
 - un **plegado** de las instrucciones resultantes haciendo uso de alguna de las instrucciones obtenidas.
2. Extraer el programa transformado.

De entre todas estas técnicas, el desplegado es la herramienta de transformación fundamental de la EP. Para programas funcionales, los pasos de plegado y desplegado sólo involucran ajuste de patrones (*pattern matching*). En el caso de los programas lógicos, el ajuste de patrones se substituye por la unificación, obteniéndose así una mayor potencia de propagación de la información. Una técnica específica empleada en la EP es la denominada *especialización de puntos de control* del programa, que combina las reglas de definición, desplegado y plegado. Esta técnica puede entenderse como un proceso consistente en una definición al que le sigue una sucesión de pasos de desplegado (tantos como sea posible) que se detienen en cuanto se reconoce una configuración “ya vista” anteriormente, momento en el que se genera código para llamar a esa configuración “ya vista” (es decir, se realiza un paso de plegado). En un lenguaje imperativo, un *punto de control* es una etiqueta del programa; en un lenguaje declarativo puede considerarse que es la definición de una función o predicado.

La idea es que una etiqueta o una función del programa P pueda aparecer en el programa especializado P_{in1} en varias versiones especializadas, cada una correspondiente a datos determinados conocidos en tiempo de EP. Es conveniente notar que esta técnica es un reflejo del esquema básico de transformación anteriormente esbozado. Otra de las técnicas empleadas por la EP es la *abstracción*¹, consistente en generalizar una expresión cuando no es posible su especialización; en cierto sentido, puede verse como la transformación inversa de la instanciación y puede caracterizarse en términos de un proceso de definición al que le sigue uno de plegado.

A continuación aclaramos algunos de los conceptos y técnicas de transformación introducidos, mediante una serie de ejemplos. Con el fin de mantener la discusión en términos tan simples como nos ha sido posible, se han seleccionado los ejemplos atendiendo a su sencillez y adecuación para ilustrar aspectos concretos del proceso de EP. En esta línea de propiciar la sencillez, los ejemplos 2 y 3 hacen uso de una sintaxis funcional (algebraica) en la que los programas son conjuntos de ecuaciones² que definen funciones (recursivas).

Esta es una visión muy conveniente a nuestros fines, pues nos permite escapar de los detalles y la complejidad que introduciría el uso de lenguajes de programación de corte imperativo.

Ejemplo 2: sea el fragmento de un programa P que computa la función x^n :

$$(1) \quad \text{exp}(0, X) = 1$$

$$(2) \quad \text{exp}(N, X) = X * \text{exp}(N-1, X)$$

Si queremos especializar el programa para el dato de entrada conocido $N=3$, los pasos que podría realizar un hipotético

evaluador parcial serían:

```
% Definición
(3) exp3(X) = exp(3,X)
% Instanciación, desplegado y computación
simbólica
(4) exp(3,X) = X * exp(3-1,X)
(5) exp(3,X) = X * exp(2,X)
(6) exp(3,X) = X * (X * exp(2-1,X))
(7) exp(3,X) = X * (X * exp(1,X))
(8) exp(3,X) = X * (X * (X * exp(1-1,X)))
(9) exp(3,X) = X * (X * (X * exp(0,X)))
(10) exp(3,X) = X * (X * (X * 1))
(11) exp(3,X) = X * (X * X)
% Desplegado final
(12) exp3(X) = X * (X * X)
```

La ecuación (4) se ha obtenido a partir de la ecuación (2) por instanciación. La (5) de (4) por computación simbólica. La ecuación (6) se ha obtenido desplegando la llamada $\text{exp}(2, X)$ en la ecuación (5), esto es, sustituyendo $\text{exp}(2, X)$ por su definición (suministrada por la ecuación (2)). El resto de las acciones se generan por aplicación reiterada de pasos de computación simbólica y desplegado. Finalmente, de este conjunto de ecuaciones podemos extraer el programa especializado P :

```
exp3(X) = X * (X * X)
que computa la función  $x^3$ .
```

De manera simple, podemos entender que la nueva definición de la función $\text{exp3}(X)$ constituye una especialización del punto de control $\text{exp}(N, X)$ en la que no ha sido necesario realizar pasos de plegado, debido a que el programa especializado no contiene llamadas a función. Tampoco se han requerido pasos de abstracción para lograr la especialización. Por otra parte, nótese que las ecuaciones (1) y (2), que definen la función original $\text{exp}(N, X)$, no forman parte del programa residual debido a que $\text{exp3}(X)$, en particular, no contiene llamadas a $\text{exp}(N, X)$.

Ejemplo 3: consideremos de nuevo el programa del ejemplo 2. Si queremos especializar el programa con respecto a la expresión $\text{exp}(M, B) * \text{exp}(N, B)$, los pasos que podría realizar un hipotético evaluador parcial serían:

```
% Definición
pexp(M,N,B) = exp(M,B) * exp(N,B)
% Instanciación, desplegado y com exp(putación
simbólica
pexp(0,N,B) = exp(0,B) * exp(N,B)
pexp(0,N,B) = 1 * exp(N,B)
pexp(0,N,B) = exp(N,B)
% Instanciación, desplegado y computación
simbólica
pexp(M,0,B) = exp(M,B) * exp(0,B)
pexp(M,0,B) = exp(M,B) * 1
pexp(M,0,B) = exp(M,B)
% Desplegado y computación simbólica
pexp(M,N,B) = B * exp(M-1,B) * exp(N,B)
pexp(M,N,B) = B * exp(M-1,B) * B * exp(N-1,B)
pexp(M,N,B) = B * B * exp(M-1,B) * exp(N-1,B)
% Plegado
pexp(M,N,B) = B * B * pexp(M-1,N-1,B)
```

Pudiendo extraerse el programa especializado:

```
pexp(0,N,B) = exp(N,B)
```

```
pexp(M,0,B) = exp(M,B)
pexp(M,N,B) = B * B * pexp(M-1,N-1,B)
exp(0,X) = 1
exp(N,X) = X * exp(N-1,X)
```

que permite el cómputo de la expresión $\text{exp}(M, B) * \text{exp}(N, B)$ de manera más eficiente que el programa original.

La definición de la función $\text{pexp}(M, N, B)$ constituye una especialización del punto de control $\text{exp}(N, X)$ que facilita el cómputo de la expresión $\text{exp}(M, B) * \text{exp}(N, B)$. En este ejemplo se ha podido realizar un paso de plegado, debido a que en el proceso de especialización hemos reconocido la aparición de una regularidad: la expresión «ya vista» $\text{exp}(M-1, B) * \text{exp}(N-1, B)$, que puede substituirse por su definición $\text{pexp}(M-1, N-1, B)$, dando lugar a un paso de plegado. El programa especializado, si bien no obtiene una mejora en cuanto al número de multiplicaciones a realizar, consigue fundir las secuencias de llamadas recursivas iniciadas por $\text{exp}(M, B)$ y $\text{exp}(N, B)$, que deberían realizarse por separado, en una única secuencia en la que se simplifica el control. Si se hubiese partido de un programa iterativo, escrito en un lenguaje de programación convencional, el efecto hubiese sido la combinación de dos bucles en uno solo. Aunque al comentar los ejemplos 2 y 3 no se ha hecho énfasis en el control del proceso de EP, estos ejemplos ilustran una forma de EP, denominada *online*, en la que todas las decisiones de control (e.g. ¿qué evaluar?, ¿cuándo parar?) se toman en tiempo de EP. El próximo ejemplo sirve para ilustrar una aproximación diferente a la EP *online*.

Ejemplo 4: consideremos de nuevo la función del ejemplo 2 pero expresada en un lenguaje imperativo como el lenguaje C.

```
int exp(int n, int x)
{
    int exponencial = 1;
    while (n>0) {
        exponencial = exponencial * x;
        n = n - 1;
    };
    return exponencial;
}
```

Un evaluador parcial típico de un lenguaje imperativo necesita que se le proporcione como entrada el texto del programa así como ciertas especificaciones que le permitan distinguir entre los datos estáticos y los dinámicos. El usuario podría indicar que el primer argumento es estático y el segundo dinámico. Con estas indicaciones el evaluador parcial realiza un análisis del programa en el que introduce una serie de anotaciones, indicando qué partes del código del programa original serán evaluadas en tiempo de EP. En el código que se muestra a continuación estas anotaciones aparecen en **negrita**.

```
int exp(int n, int x)
{
    int exponencial = 1;
    while (n>0) {
        exponencial = exponencial * x;
        n = n - 1;
    };
    return exponencial;
}
```

Una vez realizado este análisis, el usuario debe suministrar valores a los argumentos estáticos, e.g. asignar el valor 3 al argumento n de la función. Entonces, el evaluador parcial

obtiene el siguiente programa residual:

```
int exp3(int x)
{
    int exponencial = 1;
    exponencial = exponencial * x;
    exponencial = exponencial * x;
    exponencial = exponencial * x;
    return exponencial;
}
```

Los evaluadores parciales también permiten un postproceso de *compresión de código* que conduciría al programa transformado óptimo:

```
int exp3(int x)
{
    int exponencial;
    exponencial = x * x * x;
    return exponencial;
}
```

Este ejemplo revela que uno de los efectos de la EP consiste en eliminar la parte «interpretativa» de los programas (imperativos): el bucle en el que se comprueba reiteradamente si « $n > 0$ » ha desaparecido del programa residual.

La clasificación de las variables del programa en estáticas o dinámicas recibe el nombre de *división*. Esta tarea es más compleja de lo que podría suponerse a simple vista, siendo difícil de implementar de forma automática, por lo que suele requerir la intervención humana. El proceso de computar una división adecuada partiendo de una división inicial provisional de las variables que se consideran la entrada del programa, recibe el nombre de *binding-time analysis* pues en él se determina el momento en el cual puede computarse el valor de una variable, es decir, cuándo un valor se enlaza a la variable. En el ejemplo 4 se ha descrito la EP como un proceso consistente en varias fases: *binding-time analysis*, generación de anotaciones y EP propiamente dicha. Este tipo de aproximación se denomina EP *offline*, por contraposición con la EP *online*. Debido a que el proceso de *binding-time analysis* es siempre aproximado y porque en un evaluador parcial *online* la decisión de qué expresión debe evaluarse se toma en tiempo de EP, lo que supone una ventaja, los evaluadores parciales *online* permiten alcanzar una mayor precisión en la especialización de los programas que los evaluadores parciales *offline*. En los evaluadores parciales *offline* es crítico encontrar la división adecuada ya que ésta determina el grado de especialización que se alcanzará [10].

Un tema importante relativo a la EP es su capacidad para reestructurar los puntos de control. La reestructuración de puntos de control tiene que ver con las relaciones que se establecen entre los puntos de control del programa original y los del programa residual. Podemos distinguir las siguientes capacidades de reestructuración:

- *Monovariante*: cualquier punto de control del programa original da lugar, como mucho, a un punto de control en el programa residual; «como mucho» quiere decir que un punto de control del programa original puede desaparecer (esto es., puede dar lugar a cero puntos de control) en el programa residual. El ejemplo 2 muestra una reestructuración monovariante, en la que el punto de control $\text{exp}(N, X)$ del programa original da lugar a un único punto de control especializado $\text{exp3}(X)$ en el programa residual.
- *Polivariante*: cualquier punto de control del programa original puede dar lugar a uno o más puntos de control en

el programa residual. El ejemplo 3 muestra una reestructuración polivariante, en la que el punto de control $\text{exp}(N, X)$ del programa original da lugar a dos puntos de control, $\text{exp}(N, X)$ y $\text{pexp}(M, N, X)$, en el programa residual.

- *Monogenético*: cualquier punto de control del programa residual se produce a partir de un único punto de control del programa original. El ejemplo 2 muestra una reestructuración monogenética, en la que punto de control especializado $\text{exp3}(X)$, en el programa residual, se produce a partir del punto de control $\text{exp}(N, X)$ del programa original.
- *Poligenético*: cualquier punto de control del programa residual se produce a partir de uno o más puntos de control del programa original. El ejemplo 3 muestra una reestructuración poligenética, en la que el punto de control especializado $\text{pexp}(M, N, X)$, en el programa residual, combina varias definiciones de función del programa original: la función $\text{exp}(N, X)$ y el operador «*».

Una buena técnica de EP debe ofrecer una capacidad de reestructuración tanto polivariante como poligenética. En [1] [17] se detalla un algoritmo de EP *online*, capaz de producir especialización polivariante y poligenética, que engloba la EP de programas lógicos y la EP de programas funcionales.

3. Objetivos de la Evaluación Parcial

En esta sección se discuten varios de los objetivos más importantes de la EP.

3.1. Aumento en la eficiencia de los programas

Una de las principales motivaciones de la EP es el aumento en la *eficiencia (speedup)* de los programas. Debido a que parte de los cálculos se han realizado previamente, en tiempo de EP, esperamos que el programa especializado sea más rápido que el programa original. Es común realizar una medida del aumento de la eficiencia, obteniendo la razón entre el tiempo de ejecución del programa original y del especializado [16]. A la hora de medir la eficiencia de la EP, debe considerarse también el coste del tiempo de especialización del programa original. La EP es claramente ventajosa cuando un (procedimiento de un) programa debe ejecutarse reiteradamente para una porción de su entrada, ya que entonces el coste que pueda suponer la especialización del programa será ampliamente amortizado por las sucesivas ejecuciones del programa especializado, que será más ‘rápido’ que el original. Neil D. Jones argumenta en [16] que la EP puede ser ventajosa incluso para una única ejecución, ya que muchas veces sucede que el tiempo invertido en la especialización del programa original sumado al tiempo de ejecución del programa especializado es inferior al tiempo que resultaría si se ejecutase el programa original con toda su entrada.

Como ejemplo, puede afirmarse que el evaluador parcial de programas declarativos multiparadigma INDY (ver Anexo) consigue mejoras en la eficiencia de los programas especializados que en muchos casos superan el 100%.

3.2. Productividad, reusabilidad y modularidad

Otro objetivo de la EP es propiciar la *productividad* en el desarrollo de programas mediante el aumento de la *reusabilidad* y la *modularidad* del código. De todos es sabido que es más fácil establecer el significado declarativo (correcto) para una especificación sencilla de un problema;

por contra, la ejecución de esta especificación como programa puede resultar ineficiente.

Un evaluador parcial puede facilitar y hacer más ágil el desarrollo de los programas al permitir explotar una biblioteca de plantillas genéricas y simples (cuyo significado declarativo sea, sin duda, el esperado) que posteriormente se especializan de forma automática para producir un código más eficiente. La corrección del proceso de EP asegura la corrección semántica del programa especializado. Disponer de una biblioteca de plantillas genéricas, para las tareas más comunes, también mejoraría la reusabilidad del código.

Por otra parte, cuando programamos, muchas veces nos encontramos con un conjunto de tareas similares para resolver y que corresponden a diferentes aspectos de un mismo problema. Una forma de afrontar esta situación es escribir un procedimiento específico y eficiente para cada una de estas tareas. Podemos enumerar dos desventajas en esta forma de proceder:

1. Debe de realizarse un exceso de programación, lo que aumenta el coste de creación del programa y de su verificación.
2. El mantenimiento del programa se hace más dificultoso, ya que un cambio en las especificaciones puede requerir el cambio de cada uno de los procedimientos.

Con ser grave la primera de las deficiencias apuntadas, la segunda es la que puede producir mayores costes a largo plazo. Muchos estudios indican que el mayor coste en el *ciclo de vida* de un programa no es el coste inicial de diseño, codificación y verificación, sino el coste posterior asociado al mantenimiento del programa mientras está en producción y uso. Una solución alternativa, que elimina las deficiencias comentadas anteriormente, consiste en escribir un procedimiento altamente parametrizado capaz de solucionar cada uno de los aspectos del problema. Nuevamente, podemos apuntar dos deficiencias:

1. La dificultad propia de programar un procedimiento genérico que cubra todas las alternativas de forma eficiente.
2. La ineficiencia inherente a este tipo de procedimientos, ya que un procedimiento altamente parametrizado gastará mucho de su tiempo de ejecución en la comprobación e interpretación de los parámetros y (relativamente) poco tiempo en los cómputos que debe realizar.

La EP puede ayudarnos a vencer esta disyuntiva. Podemos escribir un procedimiento genérico altamente parametrizado (posiblemente ineficiente) y utilizar un evaluador parcial para especializarlo, suministrando los valores de los parámetros adecuados para cada una de las tareas específicas a resolver. Esto permite obtener automáticamente un conjunto de procedimientos específicos y eficientes, tal y como deseábamos, sin aumentar las tareas de programación y mantenimiento.

3.3. Autoaplicación y generación automática de programas

Uno de los objetivos más perseguidos en el campo de la evaluación parcial es lograr evaluadores parciales autoaplicables. Esto es, evaluadores parciales que pueden especializarse con respecto a ellos mismos. La autoaplicación permite llevar a la práctica los resultados teóricos formulados por Futamura y Ershov, que propician la generación automática de programas y que hoy se conocen como proyecciones de Futamura. La primera proyección afirma que se puede compilar un programa fuente especializando su intérprete con respecto a dicho programa fuente. La segunda dice que se puede

obtener un compilador mediante autoaplicación del evaluador parcial, esto es., especializando el propio evaluador parcial con respecto a un intérprete del lenguaje. La tercera establece que es posible obtener un generador de compiladores (capaz de transformar un intérprete en un compilador) mediante una doble autoaplicación del evaluador parcial. Así pues, la EP permite la compilación (primera proyección), la generación de compiladores (segunda proyección) y la generación de generadores de compiladores (tercera proyección).

Dentro de la generación automática de programas una de las aplicaciones más notables es la *generación de compiladores dirigida por la semántica* [20]; por ello entendemos lo siguiente: dada una especificación de un lenguaje de programación, basada en una semántica formal³, transformarla automáticamente en un compilador. La motivación para la generación automática de compiladores es clara: el ahorro en esfuerzos de programación que supone la construcción de un compilador, que en ocasiones no es correcto con respecto a la semántica propuesta para el lenguaje que compila. La corrección de la EP permite que la transformación automática de una especificación semántica de un lenguaje en un compilador haga desaparecer estos errores. Las tareas de diseñar la especificación de un lenguaje, escribir el compilador y mostrar la corrección del compilador, se reducen a una sola tarea: escribir la especificación del lenguaje en una forma adecuada para ser la entrada de un generador de compiladores. Como puede apreciarse, la EP y la autoaplicación tienen unas posibilidades muy prometedoras. Aunque todavía se necesita algún esfuerzo de investigación para entender perfectamente su teoría y sus técnicas prácticas, la EP ya ha tenido sus primeras aplicaciones industriales en diversos campos tecnológicos.

4. Aplicaciones de la Evaluación Parcial

La EP ha sido aplicada intensivamente tanto a los lenguajes imperativos (e.g., el lenguaje C) como a los declarativos (e.g., el lenguaje PROLOG, el lenguaje Scheme y otros) y extensivamente a una gran variedad de problemas concretos, una muestra de los cuales se presenta a continuación:

4.1. Mantenimiento del software y comprensión de programas

En [6] [7] se desarrollan técnicas para el mantenimiento del software y la comprensión de programas (*program understanding*) basadas en la EP. Los autores desarrollan un evaluador parcial para programas escritos en el lenguaje FORTRAN que aplican a la comprensión de viejos programas científicos (para la gestión de centrales atómicas y satélites de comunicaciones). Con el paso del tiempo, estos programas se han hecho inmanejables, debido a numerosas extensiones, requiriéndose un gran esfuerzo para su mantenimiento. El evaluador parcial utiliza, principalmente, propagación de constantes y simplificación de alternativas por una de sus ramas con el fin de obtener un programa (especializado para ciertos valores de sus variables) más simple y fácil de entender.

4.2. Panificación de tripulaciones en compañías aéreas

En una compañía aérea, después de los gastos en combustible se sitúan en cuantía los gastos que originan el movimiento de las tripulaciones. En las grandes compañías aéreas estos gastos pueden superar los mil millones de dólares. Por consiguiente, se ha invertido un gran esfuerzo en mejorar la planificación relativa a estos aspectos. En [3] se presenta una solución a este problema basada en el uso de la EP para la

mejora de los complicados programas de planificación que son necesarios. El sistema de planificación de tripulaciones desarrollado por Carmen Systems AB incorpora un evaluador parcial, escrito en el lenguaje funcional Haskell, que consigue aumentos en la eficiencia con respecto a los programas originales de entre un 30% y 65%. Si se tiene en cuenta que la ejecución de este tipo de programas requiere varias horas, ese aumento en la eficiencia supone una ganancia considerable. En la actualidad el sistema es utilizado por Lufthansa, la mayor compañía aérea de la Unión Europea.

4.3. Protocolos de procedimiento de llamada remota

La EP también se ha empleado para la optimización de *software* de sistemas. Un protocolo de procedimiento de llamada remota permite que un procedimiento remoto se ejecute aparentemente como si se tratase de uno local, siendo transparente para el usuario del sistema el hecho de que, en realidad, los cómputos tengan lugar en una máquina remota dentro de un sistema operativo distribuido. En [18] se informa de la optimización de partes del protocolo RPC (*Remote Procedure Call*) de Sun, mediante el empleo de Tempo [12], un evaluador parcial para el lenguaje C. Se han obtenido mejoras en la eficiencia del orden del 1.35 en los procedimientos de codificación/decodificación de argumentos y de hasta un 3.75 en el protocolo RPC mismo.

4.4. Simulación de circuitos

Los simuladores de circuitos toman como entrada una descripción de un circuito eléctrico, construyen las ecuaciones diferenciales que describen su comportamiento y las resuelven mediante el empleo de métodos numéricos. Berlin y Weise [4] citan mejoras importantes en la eficiencia como resultado de especializar un simulador de circuitos, escrito en el lenguaje funcional Scheme, para un circuito particular.

4.5. Software adaptable

Recientemente, en [9] se ha estudiado cómo emplear la EP para incluir un comportamiento adaptable en programas Java existentes. Para conseguir este tipo de comportamiento se emplean las denominadas *clases especializadas*. La idea que se esconde detrás de las clases especializadas consiste en asociar implementaciones alternativas a una clase Java ordinaria. Las implementaciones alternativas se diferencian por su estado interno (una serie de predicados sobre los atributos). El comportamiento adaptable consiste en utilizar la implementación alternativa adecuada al estado interno del objeto. De esta forma, el objeto se adapta por sí mismo a la situación descrita por la clase especializada. Aquí, el papel de la evaluación parcial es suministrar, automáticamente, una implementación optimizada para cada caso específico.

Las técnicas de EP y adaptabilidad del software también se han aplicado a los sistemas operativos [5]. Por último, en [11], se sugiere como utilizar las clases especializadas para integrar la EP en el proceso de desarrollo del *software*. También, en esta referencia, se comentan otras aplicaciones concretas de la EP en el campo de la ingeniería del *software*.

4.6. Verificación de programas

Una de las mayores dificultades a la hora de emplear técnicas de verificación formal, como la comprobación de modelos (*model checking*), es la generación de modelos a partir del programa fuente que puedan constituir la entrada de diversas

herramientas para la comprobación de modelos. Para cubrir esta tarea es necesario el empleo de técnicas muy sofisticadas de transformación, abstracción y análisis de programas [13]. En [15] se utiliza una combinación de las técnicas de interpretación abstracta y EP para la construcción de modelos abstractos, a partir del código fuente de programas escritos en el lenguaje ADA, para la verificación de dichos programas. Posteriormente, esas mismas técnicas se han aplicado al lenguaje Java dentro del proyecto Bandera (ver Anexo).

5. Conclusiones

En este artículo se ha introducido la técnica de transformación automática de programas denominada evaluación parcial y se han descrito sus características y principales objetivos. También se han presentado algunas de sus aplicaciones prácticas más recientes, de interés científico y tecnológico, como son: mantenimiento del *software* y comprensión de programas; optimización de *software* de aplicación y de sistemas; simulación de circuitos; obtención de software adaptable; y verificación de programas. Por la relevancia de los objetivos perseguidos, su incidencia positiva en el desarrollo del *software* y el interés de sus aplicaciones, se puede afirmar que las ventajas prácticas de la evaluación parcial están fuera de toda duda y que el uso de sus técnicas en la ingeniería del *software* puede resultar altamente beneficioso.

Referencias

- [1] E. Albert, M. Alpuente, M. Falaschi, P. Julián, and G. Vidal. «Improving Control in Functional Logic Program Specialization». In *Proc. of SAS*, Springer LNCS 1503:262-277, 1998.
- [2] M. Alpuente, M. Falaschi, and G. Vidal. «Partial Evaluation of Functional Logic Programs». *ACM TOPLAS*, 20(4):768-844, 1998.
- [3] L. Augustson. «Partial Evaluation in Aircraft Crew Planning». In *Proc. of PEPM, ACM Sigplan Notices*, 32(12):127-136, 1997.
- [4] A. Berlin and D. Weise. «Compiling Scientific Code Using Partial Evaluation». *IEEE Computer*, 23(12):25-37, 1990.
- [5] A. Black, C. Consel, C. Cowan, C. Krasik, C. Pu, E.N. Volanschi, and J. Walpole. «Specialization classes: An object framework for specialization». In *fifth IEEE IWOOS*, pp 286-300. 1996. Also available at <http://compose.labri.fr/papers/#spec_decl>.
- [6] S. Blazy and P. Facon. «Partial evaluation for the understanding of fortran programs». In *Soft. Engineering and Knowledge Engineering*, pp 517-525, 1993.
- [7] S. Blazy and P. Facon. «An automatic interprocedural analysis for the understanding of scientific application programs». In *Dagstuhl Int'l Seminar on Partial Evaluation*, Springer LNCS 1110:1-16, 1996.
- [8] R.M. Burstall and J. Darlington. «A Transformation System for Developing Recursive Programs». *Journal of the ACM*, 24(1):44-67, 1977.
- [9] C. Consel, C. Cowan, G. Muller, and E.N. Volanschi. «Declarative specialization of object-oriented programs». In *ACM SIGPLAN conf. on OOPSLA*, pp 286-300, 1997. Also available as TR RR-3118, INRIA, 1997.
- [10] C. Consel and O. Danvy. «Tutorial notes on Partial Evaluation». In *Proc. of POPL*, pp 493-501, ACM Press, 1993.
- [11] C. Consel, L. Hornof, J. Lawall, R. Marlet, G. Muller, J. Noyé, S. Thibault, and E.N. Volanschi. «Partial Evaluation for Software Engineering». In *ACM Computing Surveys*, 30(3), 1998.
- [12] C. Consel, L. Hornof, F. Noël, J. Noyé, and E.N. Volanschi. «A Uniform Approach for Compile-Time and Run-Time Specialisation». In *Proc. of Dagstuhl Seminar on Partial Evaluation*, Springer LNCS 1110:54-72, 1996.
- [13] J. Corbett, M. Dwyer, J. Hatcliff, S. Laubach, C.S. Pasareanu, Robby, and H. Zheng. «Bandera : Extracting finite-state models from java source code». In *Proc. of ICSE 2000*, 2000. Available at <<http://www.cis.ksu.edu/santos/bandera/>>.
- [14] J. Gallagher. «Tutorial on Specialisation of Logic Programs». In *Proc. of PEPM*, pp 88-98. ACM Press, 1993.
- [15] J. Hatcliff, M.B. Dwyer, and S. Laubach. «Staging static analyses using abstraction-based program specialization». In *Proc. of PLILP*.

Springer LNCS 1490, 1998.

[16] N.D. Jones, C.K. Gomard, and P. Sestoft. «*Partial Evaluation and Automatic Program Generation*». Prentice-Hall, 1993.

[17] P. Julián-Iranzo. «*Especialización de Programas Lógico Funcionales Peresosos*». PhD thesis, DSIC-UPV, May. 2000.

[18] R. Marlet, G. Muller, and E.N. Volanschi. «Scaling up Partial Evaluation for optimizing the sun commercial rpc protocol». In *Proc. of PEPM, ACM Sigplan Notices*, 32(12):116-126, 1997.

[19] A. Pettorossi and M. Proietti. «Transformation of Logic Programs: Foundations and Techniques». *Journal of Logic Programming*, 19,20:261-320, 1994.

[20] M. Tofte. «*Compiler Generators: What They Can Do, What They Might Do, and What They Will Probably Never Do*». *EATCS Monographs* 19. Springer-Verlag, 1990.

Anexo: la Evaluación Parcial en Internet

Sin ánimo de ser exhaustivo, en este anexo se enumeran y comentan algunas direcciones de interés sobre EP. Debido a la inestabilidad de la estructura de los sitios web de Internet, la información aquí suministrada es presumible que no sea un fiel reflejo de la realidad futura y tenga fecha de caducidad.

Informaciones de caracter general

<<http://www.dina.kvl.dk/~sestoft/pebook/>>: texto completo y material en línea relacionado con el libro «*Partial {E}valuation and Automatic Program Generation*» de N.D. Jones, C.K. Gomard y P. Sestoft [16].

Proyectos y grupos de investigación

- Proyecto Bandera <<http://www.cis.ksu.edu/santos/bandera/>>: proyecto del laboratorio SAnToS (*Laboratory for Specification, Analysis and Transformation of Software*) de la Universidad Estatal de Kansas. El objetivo del proyecto es la definición de técnicas de transformación para la construcción de modelos que permitan la verificación de programas.
- Grupo CLIP Lab <<http://www.clip.dia.fi.upm.es/>>: el grupo CLIP Lab (*Computational Logic, Implementation, and Parallelism Laboratory*) de la Universidad Politécnica de Madrid. Este grupo tiene una amplia experiencia en el campo de la transformación y optimización de programas declarativos mediante técnicas de especialización abstracta.
- Proyecto Compose <<http://compose.labri.fr/>>: proyecto conjunto del INRIA (*Institut National de Recherche en Informatique et en Automatique*), el IRISA (*Institut de Recherche en Informatique et Systèmes Aléatoires*) y la Universidad de Rennes para el diseño y desarrollo de programas y sistemas adaptables mediante el empleo de técnicas de EP.
- Grupo DTAI <<http://www.cs.kuleuven.ac.be/~dtai/>>: el grupo DTAI (*Declaratieve Talen en Artificiele Intelligentie*) de la Universidad Católica de Leuven, Bélgica. Realizan una extensa investigación en el campo de la EP de programas lógicos.
- Grupo ELP <<http://www.dsic.upv.es/users/elp/>>: Grupo ELP (Extensiones de la Programación Lógica) de la Universidad Politécnica de Valencia. Sus áreas de interés se orientan, fundamentalmente, a la integración de lenguajes lógicos y funcionales y a la EP de programas lógico-funcionales, así como otras técnicas de transformación y manipulación de programas.
- Grupo TOPPS <<http://www.diku.dk/forskning/topps/>>: el grupo TOPPS (*Teori Og Praksis for ProgrammeringsSprog*) del Departamento de Ciencias de la computación de la Universidad de Copenhague (DIKU - *Datalogisk Institut Kobenhavns Universitet*) puede considerarse como el grupo pionero en el estudio de la EP. El grupo ha impulsado una buena parte de los desarrollos teóricos y prácticos en este área de conocimiento.

3. Evaluadores parciales y otros productos

- C-Mix <<http://www.diku.dk/forskning/topps/activities/cmix/>>: evaluador parcial *offline* para el lenguaje C desarrollado por el grupo TOPPS del DIKU, Dinamarca.
- DPPD <<http://www.staff.soton.ac.uk/~mal/systems/dppd.html>>: DPPD es el acrónimo de «*Dozens of Problems for Partial Deduction library of benchmarks*», una librería de programas lógicos, con los que realizar *benchmarks*, mantenida por M. Leuschel.

- ECCE <<http://www.staff.soton.ac.uk/~mal/systems/ecce.html>>: evaluador parcial *online* para programas lógicos implementado por M. Leuschel en la Universidad Católica de Leuven, Bélgica.
- INDY <<http://www.dsic.upv.es/users/elp/indy/>>: el sistema INDY (*Integrated Narrowing - Driven specialization system*) es un evaluador parcial *online* para programas lógico-funcionales desarrollado por E. Albert y G. Vidal en la Universidad Politécnica de Valencia, España.
- JSCC <<http://compose.labri.fr/prototypes/jssc/>>: especializador de clases Java desarrollado, dentro del proyecto Compose, en la Universidad de Rennes, Francia.
- Mixtus <<http://www.sics.se/ps/mixtus.html>>: evaluador parcial *online* para el lenguaje Prolog desarrollado por D. Sahlin en el SICS (*Swedish Institute of Computer Science*), Suecia.
- PEVAL <<http://www.dsic.upv.es/users/elp/peval/>>: un evaluador parcial para el lenguaje de programación multiparadigma Curry, desarrollado por E. Albert y G. Vidal en la Universidad Politécnica de Valencia, España, y Michael Hanus de la Universidad Christian-Albrechts de Kiel, Alemania.
- SAGE <<http://www.cogsci.ed.ac.uk/~corin/goedel.html>>: SAGE (*Self-Applicable Goedel partial Evaluator*) es un evaluador parcial autoaplicable para programas Goedel desarrollado por Corin Gurr en la Universidad de Bristol, Inglaterra.
- Schism <<http://compose.labri.fr/prototypes/schism/>>: evaluador parcial *offline* autoaplicable desarrollado por Charles Consel, para un subconjunto del lenguaje Scheme.
- Similix <<http://www.diku.dk/forskning/topps/activities/similix.html>>: Similix fue implementado por Bondorf y Danvy en el DIKU, Dinamarca. En su origen, fue un evaluador parcial *offline* autoaplicable para un subconjunto (de primer orden) del lenguaje Scheme. En la actualidad, se ha logrado extender a un amplio subconjunto de dicho lenguaje con características de orden superior.
- SML-mix <<http://www.diku.dk/forskning/topps/activities/sml-mix.html>>: prototipo de evaluador parcial para un subconjunto apreciable del lenguaje funcional Standard ML. Fue desarrollado principalmente por L. Birkedal y M. Welinder en el DIKU, Dinamarca.
- Tempo <<http://compose.labri.fr/prototypes/tempo/>>: evaluador parcial *offline* para el lenguaje C que está siendo desarrollado conjuntamente por el IRISA, el INRIA y la Universidad de Rennes dentro del proyecto Compose. Tempo puede considerarse como el evaluador parcial para C más preciso de entre todos los existentes.

Notas

¹ La abstracción es una técnica compleja cuya caracterización precisa está fuera del alcance pretendido en esta introducción.

² En este contexto, las ecuaciones juegan el mismo papel que las instrucciones en los lenguajes de programación más convencionales.

³ Nótese que la semántica operacional de un lenguaje puede considerarse que es un intérprete (abstracto) del lenguaje.

Agradecimientos

Agradezco a **María Alpuente** la lectura atenta de un primer borrador de este trabajo y sus valiosos comentarios que han contribuido grandemente a su mejora.

Lenguajes informáticos

Joaquín Ezpeleta Mateo, Pedro Gascón Campos,
Natividad Porta Royo

Depto. de Informática e Ingeniería de Sistemas, Centro
Politécnico Superior, Universidad de Zaragoza

<ezpeleta@posta.unizar.es>

<pedrogascon74@yahoo.es>

<nporta@aragob.es>

Resumen: en este artículo se presenta un compilador para un lenguaje de programación imperativo con aserciones embebidas. El código generado por el compilador comprueba, en tiempo de ejecución, si las aserciones son satisfechas por los datos correspondientes a dicha ejecución. Como lenguaje de especificación se usa la Lógica de Primer Orden. El lenguaje de especificación dispone, además de aserciones genéricas, de elementos especializados: aserciones Pre/Post para la especificación de procedimientos y funciones y aserciones de invarianza y expresiones de cota para bucles. Dado que uno de los objetivos buscados era que la herramienta fuera utilizable en diferentes plataformas, el compilador ha sido desarrollado en Java, y el código generado son bytecodes de Java.

Palabras clave: aserciones, compiladores, enseñanza de la programación, especificación y verificación formal de programas.

1. Introducción

Todos los currícula de estudios universitarios de informática contienen asignaturas que, bajo la denominación de Programación Metódica o Metodología de la Programación, por ejemplo, inciden en los «aspectos formales» de la programación secuencial. Debemos entender aquí aspectos formales como los relativos a las especificación, verificación y derivación de algoritmos usando técnicas formales ([1] [4] [7] son excelentes referencias introductorias). La orientación elegida se basa en una especificación, formal, Pre/Post del programa a desarrollar: Pre (abreviación de pre-condición) representa las propiedades que deben cumplir los datos de entrada al programa, mientras que Post (abreviación de post-condición) representa las propiedades que deben cumplir los resultados del programa (es de esperar que éstos contengan la solución al problema planteado). Este enfoque clásico también suele denominarse «diseño por contrato» [6], entendiendo que la Pre es lo que el usuario se compromete a suministrar, mientras que la Post es lo que el programa entregará a cambio.

El adjetivo formal hace referencia a que las propiedades que queremos establecer relativas a los datos estarán expresadas mediante un lenguaje formal que, en este caso, es la Lógica de Primer Orden. El enfoque se basa también en el concepto de estado del programa. Un estado está determinado por la tupla de valores de sus variables. Así, una acción A se especificará mediante una terna $\{Pre\} A \{Post\}$. Tanto en la precondición

COMPAS: un compilador para un lenguaje imperativo con aserciones embebidas

Este trabajo ha sido financiado por el proyecto TIC2000-1568-C03-01

como en la postcondición las variables libres (en el sentido de predicado de la Lógica) corresponden a variables y/o parámetros del programa A . La interpretación de la terna es la siguiente: la terna es totalmente correcta si, haciendo la invocación a la acción A desde un estado que verifica Pre , se puede asegurar que su ejecución termina y el estado en que termina verifica la aserción $Post$. De acuerdo con esta perspectiva, un programa es un transformador de predicados. El razonamiento sobre la corrección se apoya en la definición de una semántica formal del lenguaje de programación: cada instrucción del lenguaje se define en términos de la manera en que transforma predicados. Como paso metodológico previo a su verificación, el programa suele estar anotado: se añaden aserciones (normalmente en forma de comentarios especiales) en aquellos puntos del código que el programador considera oportuno, de manera que éstas serían suficientes para construir la demostración de su corrección. Un valor añadido de un programa correctamente anotado es que las aserciones sirven también como una documentación precisa, compacta y no ambigua.

Desde el punto de vista de su corrección, se pueden adoptar tres estrategias fundamentales de tratamiento de un programa anotado.

1.1. Verificación estática completa

Habitualmente se lleva a cabo «a mano» o asistida por herramientas (probadores de teoremas, por ejemplo). Es la propuesta en [1] [4] [7]. Su principal ventaja: es el método que permite tener la certeza de la corrección del software, pues está basado en su verificación matemática. Su principal inconveniente es que requiere conocimientos profundos de Lógica y manipulación simbólica, además de ser costosa en tiempo.

1.2. Verificación estática parcial

Dentro de un enfoque estático, pero sin llegar a probar la corrección del software, se han desarrollado entornos capaces de detectar algunas propiedades relativas a la corrección. En [2] se describe un entorno, denominado SPARK, muy interesante para el trabajo con un subconjunto de Ada, mientras que [9] lo hace para Modula-3 (actualmente los autores de [9] están desarrollando una versión para Java, denominado ESC/Java). Básicamente, el entorno de desarrollo contiene, además del compilador del lenguaje, herramientas que son capaces de procesar las aserciones insertas en el código, informando al programador de problemas encontra-

dos o causas de posibles problemas. Dado que se trata de un análisis estático, la información dada gira en torno a dependencias entre los datos, a uso de variables no inicializadas, a declaración de variables que no se utilizan, a guardas de bucle cuyo valor es constante, etc. Se trataría de herramientas que suponen una evolución y enriquecimiento de otras bien conocidas, como lint de Unix, por ejemplo. Respecto a la opción anterior, presentan la ventaja de que las comprobaciones se llevan a cabo de manera automática. Su principal inconveniente es que sólo analizan algunos aspectos de la corrección. Los entornos no suelen adoptar esta opción como única, sino que, como ocurre en [2] y [9], suelen ir acompañados de probadores de teoremas para adoptar complementariamente el enfoque anterior siendo, además, capaces de generar código para algunas de las aserciones consideradas, pudiendo, por tanto, ser usadas también para un enfoque dinámico (que se describe a continuación).

1.3. Comprobación dinámica

Las aserciones insertas en el código pueden, con un compilador adecuado, ser transformadas en código ejecutable, de manera que en tiempo de ejecución se convierten en un mecanismo para asegurar la corrección de la ejecución. Lenguajes como C o C++ incluyen instrucciones del tipo `assert` con esta finalidad. Eiffel va un poco más allá, incluyendo aserciones del tipo `requiere/promise` para métodos (equivalentes a nuestras Pre/Post) e `invariant` para invariantes para una clase de objetos. Una característica fundamental de este enfoque reside en su simplicidad, ya que, con un lenguaje de aserciones adecuado, permite que cualquier programador especializado pueda adoptarla. Tiene dos inconvenientes fundamentales. El primero es claro: las aserciones se chequean en tiempo de ejecución con los datos correspondientes a dicha ejecución; por lo tanto, no se prueba que el programa en sí sea correcto. «Sólo» se está estableciendo un mecanismo para asegurar que la ejecución no viola ninguna de las propiedades descritas en las aserciones. El segundo inconveniente viene determinado porque cuanto más potente es el lenguaje usado para especificar (como se ha comentado, aquí proponemos la Lógica de Primer Orden, que es mucho más potente que las simples expresiones booleanas permitidas en instrucciones `assert` o `requiere/promise`) más costoso será ejecutar el código que las aserciones generen. Esto se puede aliviar estableciendo niveles de severidad en las aserciones, de manera que algunas aserciones que han sido utilizadas en las etapas de desarrollo del software no generan código para la versión de explotación.

En este artículo presentamos un compilador que hemos desarrollado, y que permite trabajar de acuerdo con el enfoque dinámico, siendo la Lógica de Primer Orden el lenguaje usado para las aserciones. Este enfoque ya fue usado en [8] con C como lenguaje soporte. A diferencia de la propuesta de [8] o de la que siguen las aserciones en Eiffel [6], en nuestro caso hemos impuesto que las aserciones sean entes «observadores» del programa, de manera que la verificación de una aserción nunca pueda llevar a cabo un efecto lateral que modifique el propio estado en el que la aserción se está comprobando (excepto en el caso de que la aserción no sea verificada, en que se aborta la ejecución del programa). Hemos

tratado también, dentro de lo posible, de independizar el lenguaje de especificación del lenguaje de programación, buscando que el primero fuera lo más parecido posible al lenguaje de la Lógica que manejan los libros de texto utilizados en nuestras asignaturas.

En la siguiente sección hacemos una breve descripción tanto del lenguaje de programación como del de especificación. Posteriormente mostramos la arquitectura del compilador desarrollado y hacemos algunos comentarios respecto a su desarrollo. La sección 4 muestra un ejemplo de uso del entorno desde el punto de vista del usuario.

2. Breve descripción del lenguaje de programación enriquecido

Nuestro compilador corresponde a un lenguaje imperativo tipo Pascal. Se trata del lenguaje algorítmico que utilizamos en los cursos de introducción a la programación, y está muy próximo al utilizado en [1]. El fuente contiene dos tipos de elementos. Por un lado, instrucciones en el sentido de transformaciones de estados (como en cualquier lenguaje imperativo); por otro, aserciones referentes al estado del programa. Posteriormente ejemplificaremos las características del lenguaje. De momento, vamos a hacer una breve enumeración de los principales elementos.

2.1. Elementos del lenguaje imperativo

Dada su similitud con Pascal (por ello, precisamente, hemos denominado Pascual al lenguaje), no vamos a entrar en detalles de cuáles son estos elementos. La siguiente enumeración, junto con el ejemplo en el anexo, debe ser suficiente para obtener una idea de cómo es el lenguaje. La diferencia fundamental con Pascal está en las clases de parámetros. Para ellas se ha elegido las que implanta Ada, por lo que los parámetros pueden ser de entrada, de salida, o de entrada/salida, en el caso de los procedimientos, mientras que en el caso de las funciones sólo pueden ser de entrada.

- Tipos de datos: maneja tipos escalares primitivos (entero, real, booleano y carácter) y cadenas de hasta 255 caracteres.
- Dispone de los constructores `vector`, `registro` y `fichero` secuencial (de cualquier tipo previamente definido, excepto `fichero`).
- Estructuras de control: selección tipo «*If...Else...*», *selección múltiple* (varias alternativas, con semántica de una y sólo una activa), e iteración tipo «*While ...*»
- La recursividad está también permitida.

2.2. Elementos del lenguaje de aserciones

Como se ha dicho, el lenguaje de aserciones, que hemos denominado LANOESPE (Lenguaje para ANOtación y ESPEcificación) implementa un lenguaje para la Lógica de Primer Orden, enriquecido a su vez con operadores de suma, producto y conteo sobre dominios finitos. Su características principales son las siguientes:

- Las fórmulas atómicas pueden ser cualquier expresión booleana aceptada por el lenguaje. Éstas hacen referencia a los valores de las variables usadas por el programa, pero no pueden invocar funciones del mismo.

- Maneja la implicación y doble implicación entre fórmulas.
- Maneja los cuantificadores «para todo» y «existe». Considerando un vector v de dimensión n , de enteros, por ejemplo, el siguiente predicado indica que dicho vector está ordenado en sentido estrictamente creciente (PT representa «Para Todo»):

$$\text{PT } \alpha \text{ EN } [1..n-1] . v[\alpha] < v[\alpha+1]$$

- Dispone de operadores para hablar acerca del estado de los ficheros secuenciales. Si bien desde el punto de vista del lenguaje Pascal se manejan únicamente ficheros secuenciales, desde el punto de vista de su especificación, Lanoespe dispone de operadores para hablar sobre su modo de apertura (lectura o escritura), el número de datos en el mismo, la posición de su *buffer* o alguna propiedad sobre un dato concreto. Así, por ejemplo, si f es una variable de tipo fichero de enteros, el siguiente aserto establece que el fichero está abierto para lectura, que el siguiente dato que se leería es el que ocupa la posición 27 y que todos sus elementos son positivos:

$$(f_MODO = LECT) \text{ AND } (f_POS = 27) \text{ AND} \\ (\text{PT } \alpha \text{ EN } [1..f_NDATOS] . f_DATOS[\alpha] > 0)$$

- Tiene definidos, para dominios discretos y finitos, los operadores de conteo (número de elementos de un conjunto), y la suma y el producto de expresiones indexadas por dichos dominios. Por ejemplo, las siguientes aserciones representan, respectivamente, que dos elementos del conjunto $\{ 1.5, 3.75, 6.0 \}$ son mayores que el valor de la variable a , y que la suma de todos ellos es positiva:

$$\text{CARD } \alpha \text{ EN } (1.5, 3.75, 6.0) (\alpha > a) = 2 \\ \text{SUM } \alpha \text{ EN } (1.5, 3.75, 6.0) (\alpha) > 0.0$$

- Admite la definición de predicados funcionales parametrizados, como forma de notación compacta (una especie de macros enriquecidas). Lo que sigue define el predicado parametrizado que indica que las componentes $\{ aa[ii], aa[ii+1], \dots, aa[dd] \}$ están ordenadas:

$$\text{Def: ordenado } (aa, ii, dd) == \\ \text{PT } \alpha \text{ en } [ii..dd-1] . aa[\alpha] \leq aa[\alpha + 1]$$

Cuando se desee afirmar que el vector v de dimensión n está ordenado, se indicará mediante $\text{ordenado}(v, 1, n)$ en la aserción. Cuando, por el contrario, se necesite afirmar que el vector w tiene sus componentes 3 a 29 ordenadas, será suficiente con usar $\text{ordenado}(w, 3, 29)$. Como se puede apreciar en el ejemplo en el anexo esto da una gran flexibilidad al uso de aserciones.

- Distingue aserciones especiales de tipo *Pre* (precondición) y *Post* (postcondición). Éstas se usan para especificar procedimientos y funciones y son chequeadas, respectivamente, en el momento de la invocación y el abandono del procedimiento o función ($\{ -4 \}$ y $\{ -5 \}$ en el anexo, por ejemplo).
- Dispone también de aserciones especiales *Inv* (invariante para un bucle) y expresiones *Var* (variante para un bucle). Una aserción de tipo invariante, ligada a un bucle, es chequeada antes de cada iteración y después de la última. Una expresión *Var* (que debe ser entera) representa una expresión de cota para el bucle. Se chequea que cuando la guarda del bucle está abierta la expresión sea no negativa. También se chequea que sus valores correspondientes a dos iteraciones consecutivas decrezcan estrictamente ($\{ -8 \}$ en el anexo, por ejemplo).
- Por último, admite también aserciones en cualquier parte del código donde una instrucción tenga sentido (téngase presente que las aserciones generarán código para su verificación).

3. Estructura del compilador

La **figura 1** muestra un esquema de la arquitectura de la aplicación. Básicamente, además de mostrar su viabilidad, el objeto de la herramienta es que sea útil para el aprendizaje de la programación (a pequeña escala) que se lleva a cabo en los cursos introductorios de esta materia. Por ello, uno de los requisitos que establecimos inicialmente fue que el compilador fuera multi-plataforma desde el punto de vista de su ejecución y que, por otra parte, el código generado fuera a su vez ejecutable en múltiples plataformas. Para cumplir el primer requisito optamos por desarrollar la herramienta en Java. Para cubrir el segundo, optamos porque el código generado por el compilador fueran bytecodes de Java, ejecutable en cualquier máquina virtual Java (JVM).

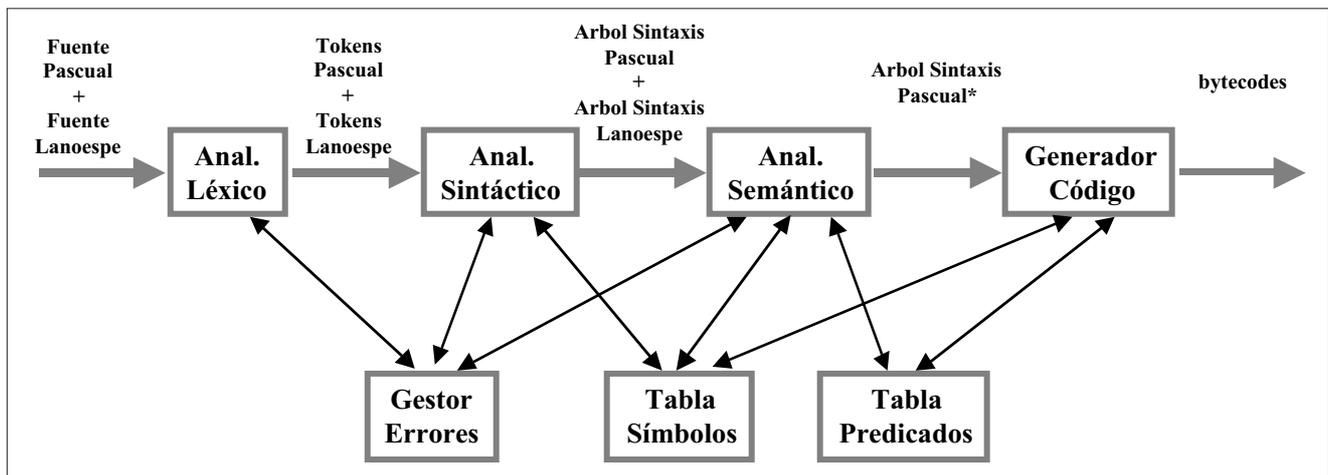


Figura 1. Esquema de la arquitectura del compilador

El compilador funciona como sigue. El fuente de entrada contiene Pascual y, opcionalmente, también aserciones escritas en Lanoespe (una opción en la invocación al compilador determina si las aserciones han de ser consideradas como tales o como simples comentarios). Al terminar correctamente las fases de análisis léxico y sintáctico, se han generado dos árboles de sintaxis, uno correspondiente al fuente Pascual y otro al fuente Lanoespe. Si la fase de análisis semántico se lleva a cabo satisfactoriamente, ésta traduce el fuente Lanoespe a Pascual, integrando el código correspondiente a las aserciones más el código Pascual en un único árbol de sintaxis decorado. Este árbol integrado, que es correcto, es pasado al generador de código, que genera un programa en bytewords para la JVM. El código generado, dependiendo de las estructuras de datos manejadas en el programa fuente, estará formado por uno o varios ficheros «.class» para la JVM.

Para la implementación del analizador léxico se ha utilizado Jlex [5], un generador de analizadores léxicos en Java (análogo a Lex de Unix o Flex de GNU). En cuanto al analizador sintáctico, se ha usado CUP [3], un generador de analizadores sintácticos en Java para gramáticas de la clase LALR (análogo a Yacc de Unix o Bison de GNU). El resto de compilador se ha implementado directamente en Java.

4. Ejemplo de uso del entorno

El **anexo 1** muestra un ejemplo de programa anotado. Una sesión típica de uso del compilador con dicho programa sería la que sigue (aquí usamos «\$» para identificar las instrucciones del usuario. Las líneas que no van precedidas por dicho símbolo corresponden a información dada por el entorno):

```
$java compas mezclar.pas -a
Se ha generado el fichero: mezclar.class.
$java mezclar
V ANTES de ser ordenado
v[ 1]=10 v[ 2]=9 v[ 3]=8 v[ 4]=7 v[ 5]=6 v[ 6]=5
v[ 7]=4 v[ 8]=3 v[ 9]=2 v[ 10]=1
V DESPUES de ser ordenado
v[ 1]=1 v[ 2]=2 v[ 3]=3 v[ 4]=4 v[ 5]=5 v[ 6]=6 v[ 7]=7
v[ 8]=8 v[ 9]=9 v[ 10]=10
```

Supongamos ahora el caso de un error sintáctico: hemos olvidado el punto y coma de la última instrucción de la acción `escribeVector`:

```
$java compas mezclar.pas -a
-----
ERROR SINTACTICO: 130, 1 -> Fin
Se espera el simbolo ; para finalizar instruccion.
-----
```

Veamos por último el caso de una ejecución que viola una aserción. Supongamos que en el punto `{ -8- }` del fuente el invariante estuviera escrito como

```
{ # INV: PT x en [ I..otroI ].vAux[ x]=v[ x] #}
que presupone que las componentes desde I hasta otroI
ya han sido copiadas, cuando en realidad sólo han sido
copiadas hasta la otroI-1. La ejecución es la siguiente:
```

```
$java mezclar
V ANTES de ser ordenado
v[ 1]=10 v[ 2]=9 v[ 3]=8 v[ 4]=7 v[ 5]=6 v[ 6]=5
v[ 7]=4
```

No se cumple la asercion. Linea: 73. Columna: 13.

5. Conclusiones

En este artículo hemos presentado un prototipo de compilador desarrollado para facilitar el aprendizaje del uso de técnicas de especificación formal de programas. A diferencia de otras herramientas existentes, el lenguaje utilizado para la especificación es el de la Lógica de Primer Orden, que es el mismo utilizado en los textos usados para nuestras asignaturas de programación metódica. Esto extiende la potencia del lenguaje de especificación permitiendo, además de lo asumible en instrucciones del tipo `assert` de otros lenguajes, el manejo de cuantificadores. Desde el punto de vista del lenguaje imperativo, éste es suficientemente amplio como para cubrir los aspectos tratados en las asignaturas básicas de programación. Actualmente estamos evaluando dos posibles líneas alternativas (aunque no excluyentes) de continuación del trabajo. Por un lado, mejorar el prototipo, extendiendo el entorno, tanto desde el punto de vista del lenguaje (introducir compilación separada, por ejemplo) como desde el punto de vista de la edición (desarrollar un editor específico que permita usar la simbología matemática en el fuente). Una segunda línea se encamina hacia la inclusión del lenguaje de especificación manejado, Lanoespe, en un compilador existente de (un subconjunto de) algún lenguaje estándar, como puede ser Pascal o Ada.

Referencias

- [1] J.L. Balcázar, *Programación metódica*, McGraw-Hill, 1993.
- [2] J. Barnes, *High Integrity Software: The SPARK Approach to Safety and Security*, Addison-Wesley, 2003.
- [3] CUP, *LALR Parser Generator for Java*. <<http://www.cs.princeton.edu/~appel/modern/java/CUP/manual.html>>.
- [4] D. Gries, *The science of programming*, Texts and Monographs in Computer Science, Springer-Verlag, 1981.
- [5] J. Lex, *A lexical analyzer generator for Java*. <<http://www.cs.princeton.edu/~appel/modern/java/JLex/current/manual.html>>.
- [6] B. Meyer, Applying «Desing by Contract», *IEEE Computer*, Vol. 25, Octubre 1992, pp.40-51.
- [7] R. Peña, *Diseño de programas. Formalismo y abstracción*, Prentice-Hall, 1998.
- [8] D.S. Rosenblum, «A Practical Approach to Programming With Assertions», *IEEE Transactions on Software Engineering*, Vol. 21, n° 1, Enero 1995, pp. 19-31.
- [9] K. Rustan, M. Leino, G. Nelson, «An Extended Static Checker for Modula-3», en Kai Koskimies, Editor. *Compiler Construction: 7th International Conference, CC'98, volume 1383 of Lecture Notes in Computer Science*. Springer, April 1998, pp.302-305.

Nota

¹El compilador Compas, que implementa el entorno descrito, se puede descargar de la siguiente dirección URL: <<http://www.cps.unizar.es/~ezpeleta/MP/MP.htm>>.

Anexo 1. Un ejemplo de fuente con aserciones

El siguiente programa en Pascual implementa la ordenación por mezcla

de un vector de enteros. El objetivo es mostrar muchos de los aspectos de Pascual y de Lanoespe.

```

{ -----
- Fichero: mezclar.pas
-
----- }

Programa mezclar;
Constantes
    n = 10;

Tipos
    vect = vector[ 1..n ] de entero;

Variables
    i: entero;
    v: vect;

{ ----- Definición de predicados ----- }
{ # Def: ordenado (aa, ii, dd) ==
  PT x en [ ii..dd-1 ] . aa[x] <= aa[x+1] #} { -1- }

{ # Def: mezcla (v1, i1, d1, v2, i2, d2, v3, i3, d3) ==
  (d3 - i3 + 1 = d2 - i2 + 1 + d1 - i1 + 1)
  AND
  (PT ind en [ i3..d3 ] .
   CARD j3 en [ i3..d3 ] (v3[j3] = v3[ind]) =
   CARD j1 en [ i1..d1 ] (v1[j1] = v3[ind]) +
   CARD j2 en [ i2..d2 ] (v2[j2] = v3[ind]))
  #} { -2- }

{ # Def: permutacion (a, b, izq, dch) ==
  PT ind en [ izq..dch ] .
  (EX j en [ izq..dch ] . a[ind] = b[j])
  AND
  (CARD k en [ izq..dch ] (a[k] = a[ind]) =
   CARD l en [ izq..dch ] (b[l] = a[ind]))
  #} { -3- }

{ ----- }
Accion merge (ES v: vect; E I, M, D: entero);
{ # Pre: (1 <= I) AND (I <= M) AND (M <= D) AND (D <= n) AND
  ordenado (v!, I, M) AND ordenado (v!, M + 1, D) #} { -4- }
{ # Post: mezcla (v!, I, M, v!, M + 1, D, v, I, D)
  AND ordenado (v, I, D) #} { -5- }
{ Com: Estando ordenados v[ I ], v[ I+1 ], ..., v[ M ]
  y v[ M+1 ], v[ M+2 ], ..., v[ D ],
  los mezcla, dejando ordenado v[ I ], v[ I+1 ], ..., v[ D ] }

Variables
    vAux: vect;
    iI, iD, iNuevo, otroI: entero;
    { # DefConst: v! #} {v!: valor de v en entrada} { -6- }

Principio
    iI := I;
    iD := M + 1;
    iNuevo := I;
    { # VAR: D+1-iNuevo #} { -7- }
    { # INV: ordenado (vAux, I, iNuevo-1) AND
      mezcla (v!, I, iI-1, v!, M+1, iD-1, vAux, I, iNuevo-1) #}
    Mq (iI <= M) AND (iD <= D)
      Si v[iI] < v[iD] Ent
        vAux[iNuevo] := v[iI];
        iI := iI + 1;
      Si_No
        vAux[iNuevo] := v[iD];
        iD := iD + 1;
      FSi
        iNuevo := iNuevo + 1;
    FMq
      otroI := iI - 1;
    Mq otroI < M { copia lo que quede de v[ I ], ..., v[ M ] }
      otroI := otroI + 1;
      vAux[iNuevo] := v[otroI];
      iNuevo := iNuevo + 1;
    FMq
      otroI := iD - 1;
    Mq otroI < D { copia lo que quede de v[ M+1 ], ..., v[ D ] }
      otroI := otroI + 1;
      vAux[iNuevo] := v[otroI];
      iNuevo := iNuevo + 1;
    FMq
      otroI := I; { Copiar sol. de vAux a v }
      { # VAR: D-otroI #}
      { # INV: PT x en [ I..otroI-1 ] . vAux[x] = v[x] #}
      { -8- }
    Mq otroI <= D
      v[otroI] := vAux[otroI];
      otroI := otroI + 1;
    FMq
Fin
{ ----- }

```

```

Accion mergeSort (ES v: vect; E pI, pF: entero);
{ # Pre: (1 <= pI) AND (pI <= pF) AND (pF <= n) #}
{ # Post: permutacion (v, v!, pI, pF) AND (ordenado (v, pI, pF)) #}
{ Com: ordena por mezcla en v las componentes de pI a pF }
Variables
    medio: entero; { # DefConst: v! #}

Principio
    Si pI < pF Ent
      medio := (pF + pI) div 2;
      mergeSort (v, pI, medio);
      mergeSort (v, medio + 1, pF);
      merge (v, pI, medio, pF);
    FSi
Fin
{ ----- }
Accion ordenMezcla (ES v: vect);
{ # Pre: verdad #}
{ # Post: permutacion (v, v!, 1, n) AND (ordenado (v, 1, n)) #}
{ Com: ordena el vector v por mezcla }
{ # DefConst: v! #}

Principio
    mergeSort (v, 1, n);
Fin
{ ----- }
Accion escribeVector (E v: vect);
{ # Pre: verdad #}
{ # Post: verdad #}
{ Com: escribe v por la salida estándar }
Variables i: entero;
Principio
    i := 1;
    Mq i <= n
      escribir ('v[ ', i, ' ] = ', v[i] );
      i := i + 1;
    FMq
      escribirLinea;
Fin
{ ----- }
Principio
    i := 1;
    Mq i <= n { inicializa el vector }
      v[i] := n - i + 1;
      i := i + 1;
    FMq
      escribirLinea ('V ANTES de ser ordenado' );
      escribeVector (v);
      escribirLinea;
      ordenMezcla (v);
      escribirLinea ('V DESPUES de ser ordenado' );
      escribeVector (v);
Fin

```

Seguridad

David Atauri Mezquida¹, Luis Fernández Sanz¹,
Matías Alcojor², Ignacio Acero³

¹ Dpto. de Programación e Ingeniería del Software,
Universidad Europea CEES; ² Safo Sistemas; ³ Alma
Technologies

<{atauri, lufern}@dpris.esi.uem.es>
<alcojor@safo.es>
<nachoa@airtel.net>

Resumen: gracias a las técnicas de esteganografía podemos ocultar un fichero en otro y así enviarlo o distribuirlo sin que sea advertido. En este artículo se describe el algoritmo que inserta una imagen en formato BMP indexado dentro de otra imagen del mismo tipo. Por último, se hace una breve indicación de nuevas aplicaciones y de las implicaciones prácticas de esta tecnología.

Palabras clave: BMP, Esteganografía, formatos gráficos, ocultación de información.

1. Introducción

La esteganografía es el arte de esconder mensajes y así hacerlos pasar inadvertidos para quien no conoce la manera de desvelarlos. Ejemplo remoto e inocente es el acróstico de la Celestina, donde se oculta el nombre de su autor y su lugar de nacimiento: «Fernando de Roias acabo la comedia de Calysto y Melybea y fve nascido en la pvebla de Montalvan» (ver **anexo**). Utilizada de forma artesanal durante muchos siglos, fue ampliamente utilizada en el siglo XX como ayuda en el espionaje, principalmente para transmitir información durante la Guerra Fría.

Lejos de ser cosa del pasado, hoy está de rabiosa actualidad. De hecho, algunas informaciones aparecidas en el diario U.S. Today sugerían (aunque no sin controversia por parte de los expertos) que podía haber sido utilizada por Bin Laden para organizar el atentado de las Torres Gemelas del 11 de septiembre de 2001, pasando a sus comandos información escondida en imágenes de Internet, en foros de noticias o en los mensajes de correo electrónico (véase [1] [2] [3] y [4]).

Todos hemos escrito, de niños, mensajes invisibles con jugo de limón que luego desvelábamos con la llama de un mechero. Otro método clásico y sencillo de esteganografía es pintar diminutos puntos bajo ciertas letras de una carta aparentemente inocua: esas letras son las que forman el mensaje oculto. La evolución ha llevado a métodos, típicos de película de espías, que consisten, por ejemplo, en miniaturizar el documento que se quiere ocultar de forma que parezca el punto final de una frase en una carta cualquiera.

Evidentemente, la aplicación de las nuevas tecnologías de la información ha supuesto una gran contribución a la evolución de las técnicas esteganográficas. En este artículo, veremos algunas aplicaciones sencillas de la esteganografía en ficheros electrónicos.

Ocultación de imágenes mediante Esteganografía

2. Definición y fundamentos

Podemos definir la esteganografía como el conjunto de técnicas que nos permiten ocultar cualquier tipo de información. No hay que confundir la criptografía con la esteganografía: la primera modifica los datos para hacerlos incomprensibles, mientras que la segunda simplemente los oculta entre otros datos. A pesar del diferente enfoque de cada una, en muchas ocasiones se combinan ambas técnicas para lograr mejores resultados.

Las razones para el uso de la esteganografía pueden ser muy variadas pero pueden aparecer porque no existe soporte para encriptar los datos o porque existe una autoridad que no permite el paso de cierta información. Así, la información viaja en los ficheros sin que nadie sepa lo que realmente transporta en su interior. Una de las aplicaciones de la esteganografía que actualmente más interés está suscitando es la aplicación de 'marcas de agua' (*watermarking*), que supongan un aviso de *copyright* o *trademark* oculto en imágenes, música o software comercial. Por otra parte, desde nuestro punto de vista, algunas de las técnicas de ocultación de virus en ficheros se pueden considerar en rigor como técnicas esteganográficas de ocultación del código maligno frente al usuario o, incluso, frente a programas antivirus.

La llegada de la informática ha permitido lograr grandes avances en las posibilidades de la esteganografía, consiguiéndose además automatizar tareas que solían ser bastante costosas en tiempo y dinero. El tipo de información que se puede ahora ocultar no se reduce a mensajes escritos; podemos ocultar digitalmente tanto texto e imágenes como sonido o incluso programas ejecutables (caso de los virus informáticos).

La base de la esteganografía informatizada se basa en que toda información digitalizada se reduce a cadenas de bits: dichas cadenas se pueden insertar ocultas entre los bits de otros ficheros que sirven de soporte o contenedor y que mantienen un aspecto inalterado. En general, se aprovecha el espacio de los bytes de basura que tienen los formatos conocidos de ficheros y, también, la información redundante. Así, si usamos archivos de sonido, la información oculta aparece como ruido de fondo, pudiendo confundirse fácilmente con una simple grabación con algo de ruido.

De hecho, cuando se esteganografía algo en un fichero binario, lo que se hace es ocultar la información en ficheros

como imágenes, en los que se pueden cambiar ciertos bits sin dejarlos inservibles. Por ejemplo, si queremos ocultar una cadena de ocho bits, cambiamos los bits menos significativos de los ocho primeros bytes del mapa de bits de una imagen en escala de grises (en el que cada píxel se corresponde con un byte) por los bits que queremos ocultar. El cambio en la imagen será tan pequeño que el ojo humano no captará la diferencia.

Para comprender mejor la aplicación de la esteganografía, en este artículo mostraremos un sencillo algoritmo de creación propia para esconder una imagen en blanco y negro dentro de otra imagen de color que no se ve aparentemente alterada. Pero antes comenzaremos por repasar los fundamentos del formato gráfico BMP que ha sido el elegido como base de nuestra demostración esteganográfica. En internet existen multitud de aplicaciones que utilizan otros formatos como JPEG, GIF ó PCX (por ejemplo, en <http://personal1.iddeo.es/albertoaa/steganos.htm>) se pueden encontrar enlaces a varios de ellos).

3. El formato BMP

BMP es un formato gráfico creado por Microsoft para Windows [5]. Permite grabar imágenes de diferentes profundidades de bit o cantidad de colores. A continuación veremos el formato de 8 bits por píxel. Para su estudio utilizaremos un editor hexadecimal con el que podremos leer el fichero byte a byte. Cada byte puede almacenar un número entre 0 y 255 que se representa en un editor hexadecimal con 2 cifras (de 00 a FF).

Aunque BMP puede utilizar compresión RLE nos fijaremos en el caso de una imagen grabada sin compresión por mayor simplicidad.

El formato BMP de 8 bits por píxel puede representar 256 colores diferentes y es de tipo indexado. Esto quiere decir que los 256 colores disponibles para la imagen se almacenan en una tabla o paleta de colores que es exclusiva de dicha imagen. Dos imágenes diferentes tendrán por lo tanto dos paletas diferentes.

En el formato de fichero se distinguen los bloques que se explican a continuación.

3.1. Cabecera

Contiene información relativa al fichero pero no del color de los píxeles.

3.1.1. Identificador del formato

Los dos primeros bytes identifican el fichero como BMP y contienen siempre el valor hexadecimal 42, 4D (66 77 en decimal, y BM en Ascii).

3.1.2. Tamaño del fichero

El tamaño del fichero se localiza en los siguientes 4 bytes pero al contrario de los números decimales, el byte más significativo en una palabra es el de la derecha (el 4º) y el menos significativo el de la izquierda (por el standard llamado *little endian*). Por ejemplo, el tamaño de un fichero de un total de

1258 bytes se almacenaría de la siguiente manera: 234-4-0-0. El valor decimal de la palabra de cuatro octetos se calcula así: $234*1+4*256+0+0=1258$.

3.1.3. Píxeles de ancho y alto

El número de columnas del mapa de bits se almacena en otra palabra de 4 bytes a partir de la posición 18 y el número de filas en otros cuatro bytes a partir de la posición 22 de manera idéntica al tamaño del fichero.

3.1.4. Paleta de colores

Ya hemos comentado más arriba que BMP de 8 bits es un formato de color indexado. La paleta de colores se almacena en la cabecera a partir de la posición 54. Cada color se codifica con sus componentes RGB pero como forman parte de la misma palabra aparecerán, al seguir el standard *little endian*, en el primer byte el azul, luego el verde y por último el rojo (BGR). Después de cada tres bytes RGB se deja otro byte con el valor cero (relleno de la palabra) de modo que la paleta ocupará $256*4=1024$ bytes.

La paleta puede tener cualquier combinación de colores o una gama completa de grises (con lo que se obtendría una imagen en blanco y negro) pero cada uno de ellos, sea color o gris, se expresa con sus componentes RGB.

3.2. Información de cada píxel

El píxel de la esquina inferior izquierda es el primero en almacenarse y lo hace en la posición 1078. Este byte no almacena información de color sino el ordinal del color de la paleta (8 bits, 256 valores; uno por cada color de la paleta). El segundo píxel será el de la derecha, y así sucesivamente hasta terminar la primera fila (la fila inferior del mapa de bits). Una vez completada la fila, el siguiente byte codifica el píxel de la izquierda de la penúltima fila.

Píxel inferior izquierdo

<i>Byte</i>	<i>hex</i>	<i>Dec</i>	
1078	0A	10	décimo color de la paleta

Como cada palabra es de cuatro bytes, en aquellos mapas que tengan un número de columnas que no es múltiplo de 4, se completará la última palabra de la fila con bytes de relleno (lo que constituye información basura).

4. Un algoritmo de ocultación

Existen diferentes maneras de ocultar información en imágenes. A continuación mencionamos las más conocidas:

- Inserción de información en el bit menos significativo (LSB).
- Filtrado y enmascaramiento.
- Transformaciones y algoritmos

La inserción de información en el bit menos significativo es la elección que hemos utilizado para nuestra aplicación. Consiste en guardar información en los bits menos significativos de manera que los cambios, no sean percibidos por el ojo humano.

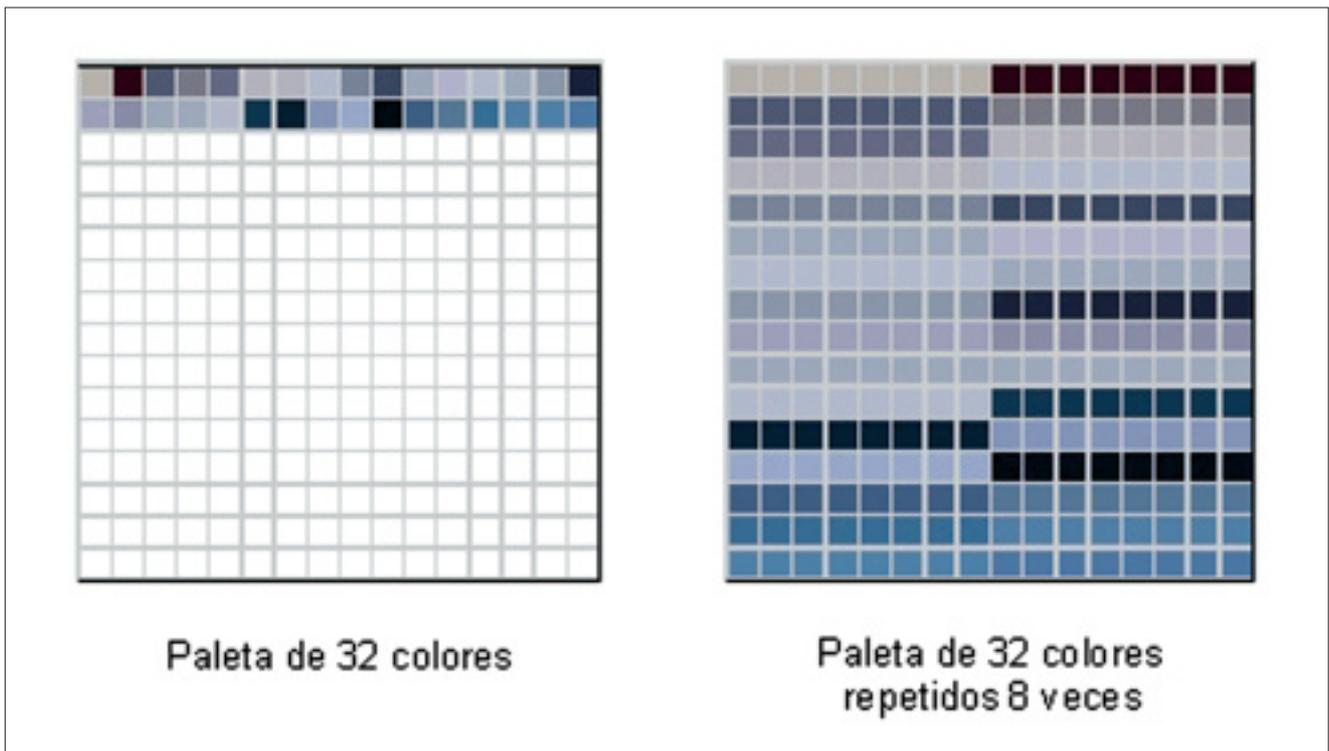


Figura 1. Paletas empleadas

Una manera de evitar esta pérdida de calidad en la imagen es utilizar una imagen modificada (como veremos a continuación) de manera que la información que, de otra manera se perdería, permanezca inamovible. En nuestro caso, las técnicas de filtrado y enmascaramiento están restringidas a imágenes de 24-bits y escala de grises.

Podemos ocultar información en la imagen aplicando una serie de modificaciones tales como la transformada de Fourier, la codificación de patrón redundante o la encriptación de la información que se ha ocultado [6].

La técnica LSB es la técnica más rápida y sencilla de esconder información. No es recomendable usarla con formatos de imagen que tengan compresión con pérdida de información. Ese es el motivo de que hayamos elegido el formato BMP para nuestra aplicación.

Para nuestra aplicación de esteganografía utilizaremos dos imágenes:

- Imagen 1 (visible): el formato de la imagen es un BMP indexado de 32 colores. La diferencia de calidad entre una imagen de 32 colores y una imagen de 256 es apenas



Figura 2. Imagen visible



Figura 3. Imagen a ocultar.

perceptible a simple vista.

· Imagen 2 (oculta): el formato de la imagen es un BMP indexado de 8 colores. Nos limitaremos a imágenes de en blanco y negro para tener una calidad razonable.

4.1. Proceso de ocultación de una imagen en otra

Para ocultar la imagen de 8 colores, lo primero que debemos hacer es transformar la imagen de 256 colores a otra imagen que solo tiene 32 colores pero cada color esta repetido 8 veces (figura 1).

Con esta transformación conseguimos liberar espacio en la imagen visible de tal forma que la imagen oculta, tenga cabida en el fichero de la imagen visible. En el mapa de bits de la imagen visible, modificamos la información correspondiente a un píxel, para que apunte a la primera posición (del grupo de 8 del mismo color).

De este modo, podemos almacenar la información del mapa de bits de la imagen oculta, en los 3 bit menos significativos, sin que ello, afecte al color del píxel que estemos modificando.

Si nuestro apuntador¹ es 16, significa que está apuntando al tercer color en la segunda fila y primera columna. El color 16

en binario es 00010000, aunque modifiquemos los 3 últimos bits, el color al que apuntamos no variará. Si tenemos 00010111 (23 en decimal), nuestro apuntador se moverá hasta la columna 8 de la misma fila, pero no variará su color.

Estos 3 bits últimos son los que usaremos para almacenar la *bitmap* de la imagen oculta. Para ello, simplemente debemos sumar el valor del apuntador de la imagen oculta a la imagen visible. A la vista del usuario la imagen no habrá variado, pero de esta manera, ya tenemos almacenado en la imagen visible, la información correspondiente al mapa de bits de la imagen oculta.

A continuación presentamos la estructura de las dos imágenes, la visible (figura 2) y la imagen que vamos a ocultar (figura 3). En este ejemplo, tenemos la imagen de un velero y dentro de ella vamos a ocultar una imagen del Golden Gate. El resultado es la imagen del velero sin ninguna modificación aparente (no la reproducimos ya que es idéntica al ojo humano que la imagen de la figura 2).

Ahora nos queda por almacenar la información de cabecera y la paleta de colores de la imagen a ocultar. Esta información la vamos a guardar en el 4º byte de la paleta de colores de la imagen original. Esto es posible, gracias a que en el formato

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00000000	42	4D	35	39	01	00	00	00	00	00	B6	00	00	00	28	00
00000016	00	00	90	01	00	00	C8	00	00	00	01	00	08	00	00	00
00000032	00	00	00	00	00	00	C0	1E	00	00	C0	1E	00	00	20	00
00000048	00	00	20	00	00	00	B7	E5	C4	00	28	20	54	00	7A	6E
00000064	74	00	95	87	8F	00	8A	7B	83	00	C6	E9	C0	00	C5	B5
00000080	BC	00	CD	BD	C3	00	A0	91	96	00	6C	58	5E	00	BF	AE

Figura 4. Código hexadecimal de la imagen visible.

GoldenGate.bmp																
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00000000	42	4D	D5	38	01	00	00	00	00	00	56	00	00	00	28	00
00000016	00	00	90	01	00	00	C8	00	00	00	01	00	08	00	00	00
00000032	00	00	00	00	00	00	12	0B	00	00	12	0B	00	00	08	00
00000040	00	00	00	00	00	00	D2	D2	D2	00	A9	A9	A9	00	0F	0F
00000064	8F	00	7E	7E	7E	00	71	71	71	00	64	64	64	00	4C	4C
00000080	4C	00	2C	2C	2C	00	03	03	03	03	03	03	03	03	05	04

Figura 5. Código hexadecimal de la imagen a ocultar.

BMP, la paleta de colores sólo utiliza los tres primeros bytes para codificar los componentes RGB de una imagen en color, y el 4º byte está vacío.

La paleta de colores empieza en el byte 54 de la imagen visible, tal como se muestra en la **figura 4**. El primer byte libre se encuentra en la posición 57, en esta posición almacenamos el primer byte de la imagen a ocultar (ver **figura 5**). La siguiente posición vacía es la 61: en ella almacenaremos el segundo byte de la imagen oculta, Y así sucesivamente hasta haber almacenado toda la información de cabecera y paleta de la imagen oculta.

Como podemos observar en la **figura 6**, las posiciones 57,61,65,69,... se corresponden con los bytes vacíos en la representación de un color RGB. Ahora están ocupados por cada uno de los bytes que componen la cabecera y la paleta de colores de la imagen que queremos ocultar. En la primera parte, en color amarillo, podemos ver la cabecera de la imagen. En la segunda parte, en color verde, podemos ver el cuerpo de la imagen y cómo se van insertando los bytes (rodeados por un círculo) de la imagen oculta.

5. Algunas reflexiones sobre la Esteganografía

Aunque en este mismo artículo hemos comenzado apelando a los efectos más espectaculares y vinculados a la delincuencia (es decir, la esteganografía como medio de burlar una autoridad que no permite el paso de cierta información), sin embargo ésta técnica puede aportar soluciones a problemas habituales. Así, si queremos proteger el uso no autorizado de imágenes que hemos creado o de las que disponemos los

derechos de explotación, podemos usar la esteganografía (insertando una firma oculta en la imagen) para verificar el origen de una determinada foto. También permitiría incorporar comentarios o descriptores de las imágenes que se encuentran incorporados en el mismo fichero, haciendo así que la referencia de información a ficheros gráficos, sonoros, etc. sea más duradera y segura frente a pérdidas. De hecho, el uso de la esteganografía no se limita a ocultar información en imágenes, sino que pueden usarse otros orígenes de datos como un fichero de Audio WAV ó MP3 (por ejemplo, el programa Mandelstag (disponible en <<http://www.arnal.es/free/cripto/estega/mandelst.htm>>).

Evidentemente cuando se proporciona un medio para ocultar información, existe la corriente tecnológica contraria que trata de contrarrestar la ocultación mediante técnicas de detección conocidas como esteganálisis. Uno de los medios empleados consiste en aplicar análisis estadísticos a la codificación binaria de los ficheros para tratar de detectar patrones de repetición anormales, por ejemplo, para lo que suele ser una imagen normal. También existen otras posibilidades de sospechar la ocultación de información en un fichero y de adoptar contramedidas contra el análisis [7].

Referencias

- [1] P. Pardo, «Los infinitos agujeros de la red», Expansión, 19 de septiembre de 2001 (disponible en <<http://www.expansiondirecto.com/edicion/noticia/0,2458,56468,00.html>>).
- [2] AP, «Bin Laden's cybertrail proves elusive», USA Today, 2 de octubre de 2001 (disponible en <<http://www.usatoday.com/life/cyber/tech/2001/09/20/attacks-cybertrail.htm>>).
- [3] AP, «150 people linked to bin Laden in custody», USA Today, 11

VeleroEncriptado.bmp																
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00000000	42	4D	B5	3C	01	00	00	00	00	00	36	04	00	00	28	00
00000016	00	00	90	01	00	00	C8	00	00	00	01	00	08	00	00	00
00000032	00	00	80	38	01	00	C0	1E	00	00	C0	1E	00	00	00	01
00000048	00	00	00	01	00	00	B7	B5	C4	42	B7	B5	C4	4D	B7	B5
00000064	C4	D5	B7	B5	C4	38	B7	B5	C4	01	B7	B5	C4	00	B7	B5
00000080	C4	00	B7	B5	C4	00	28	20	54	00	28	20	54	00	28	20

Figura 6. Código hexadecimal de la imagen oculta en la visible.

de octubre de 2001 (disponible en <<http://www.usatoday.com/news/attack/2001/10/05/custody.htm>>).

[4] Reuters, «Researchers: No secret bin Laden messages on sites», USA today, 3 de noviembre de 2001 (disponible en <<http://www.usatoday.com/life/cyber/tech/2001/10/17/bin-laden-site.htm>>).

[5] W. Wouters, «BMP format», Clean Coding Company, 21 de febrero de 1997 (disponible en <<http://www.wotsit.org/search.asp?s=windows>>).

[6] N.F. Johnson, S. Jajodia, «Steganography: Seeing the Unseen», IEEE Computer, febrero, 1998, pp. 26-34.

[7] N.F. Johnson, S. Jajodia, «Steganalysis of Images Created Using Current Steganography Software», Lecture Notes in Computer Science, Vol. 1525, Springer-Verlag, 1998, pp. 273-289.

Nota

¹ Apuntador: en una imagen BMP indexada, el mapa de bits contiene un número que codifica el píxel y cuyo valor está definido en la paleta de colores que hemos definido con anterioridad. Llamamos apuntador a dicho número.

Anexo

El silencio escuda y suele encobrir
la[s] falta[s] de ingenio y *torpeza de* lenguas;
blasón que es contrario, publica sus menguas
a[!] *quien* mucho habla sin mucho sentir.
Como [la] hormiga que dexa de yr
holgando por tierra con la provisión,
jactóse con alas de su perdición;
lleváronla en alto, no sabe dónde yr.

El ayre gozando ageno y estraño,
rapina es ya hecha de aves que buelan;
fuerte más que ella, por cevo la llevan;
en las nuevas alas estava su daño.
Razón es que aplique a mi pluma este engaño,
no *despreciando* a los que *me* arguyen,
assí que a mí mismo mis alas destruyen,
nublosas y flacas, nascidas de ogaño.

Donde ésta gozar pensaba volando,
o yo de *screvir* cobrar más honor,
del[o] uno [y] del otro nació disfavor;
ella es comida y a mí están cortando
reproches, revistas y tachas. Callando
obstara y los daños de invidia y murmulos;
insisto *remando*, y los puertos seguros
atrás quedan todos ya quanto más ando.

Si bien *queréys ver* mi limpio motivo,
a qual se endereça de aquestos extremos,
con qual participa, quién rige sus remos,
Apolo, *Diana* o *Cupido* altivo,
buscad bien el fin de aquesto que escribo,
o del principio leed su argumento;
leeldo [y] veréys que, aunque dulce cuento,
amantes, que os muestra salir de cativo.

Como el doliente que píldora amarga
o *la* rescela o no puede tragar,
métenla dentro del dulce manjar,

engañase el gusto, la salud se alarga,
desta manera mi pluma se embarga,
imponiendo dichos lascivos, rientes,
atrae los oídos de penadas gentes,
de grado escarmientan y arrojan su carga.

Estando cercado de dubdas y antojos,
compuse tal fin que principio desata;
acordé [de] dorar con oro de lata
lo más fino tíbar que vi con mis ojos,
y encima de rosas sembrar mill abrojos.
Suplico, pues suplan discretos mi falta;
teman grosseros y en obra tan alta,
o vean y callen o no den enojos.

Yo vi en Salamanca la obra presente;
movíme [a] acabarla por estas razones;
es la primera, que esté en vacaciones,
la otra, *inventarla persona prudente*,
y es la final ver la más gente
buelta y mezclada en vicios de amor;
estos amantes les pornán temor
a fiar de alcahueta ni *falso* sirviente.

Y así que esta obra *en el proceder*
fue tanto breve, quanto muy sutil;
vi que portava sentencias dos mill;
en forro de gracias, lavor de plazer.
No hizo Dédalo cierto a *mi ver*
alguna más prima entretalladura,
si fin diera en esta su propia escriptura
Cota o Mena con su gran Saber.

Jamás [yo] no *vide en lengua romana*,
después que me acuerdo, ni nadie la vido,
obra de estilo tan alto y sobido
en tusca ni griega ni en castellana.
No *trae* sentencia de donde no mana
loable a su autor y eterna memoria,
al qual Jesuchristo reciba en su gloria
por su pasión sancta que a todos nos sana.

Vosotros, los que amáys, tomad este enxemplo,
este fino arnés con que os defendáys;
bolved ya las riendas por que n'os perdáys;
load siempre a Dios visitando su templo.
Andad sobre aviso; no seáys dexamplio
de muertos y bivos y propios culpados;
estando en el mundo yazéys sepultados;
muy gran dolor siento quando esto contemplo.

[Olvidemos los vicios que así nos prendieron;
no confiemos en vana esperança.
Temamos aquél que espinas y lança,
açotes y clavos su sangre vertieron.
La su santa faz herida escupieron;
vinagre con hiel fue su potación;
a cada santo lado consintió un ladrón.
Nos lleve, le ruego, con los que creyeron.]

Seguridad

José A. Mañas Argemí

Depto. de Ingeniería de Sistemas Telemáticos,
Universidad Politécnica de Madrid; socio de ATI

<jmanas@dit.upm.es>

Resumen: en este artículo se pasa revista a conceptos aparentemente sencillos como “confianza” o “seguridad” y a su significado en un entorno como el actual en el que las personas físicas realizan transacciones a través de la Red, ya sea entre ellas, con empresas o con Administraciones Públicas. Se analiza la caracterización de estos individuos como sujetos activos y como sujetos pasivos que se mueven en este territorio con más o menos confianza y se responde a preguntas tales como ¿podemos fiarnos de la gente?, ¿de quién y de qué se fía la gente?

Palabras clave: confianza, firma electrónica, seguridad, servicios electrónicos.

1. Introducción

El éxito técnico de Internet como medio de comunicación universal ha abierto enormes expectativas empresariales e individuales. Decimos que Internet es universal en el sentido de que es potencialmente capaz de llegar a lugares recónditos, como refleja la simpática frase «tener tono Internet». También es Internet universal en el sentido de ser neutral respecto de lo que se comunica por ella, privado, público, mercantil, administrativo o lúdico. Y esta sensación de tener un medio virgen para explotar ha lanzado atrevidas aventuras empresariales (punto.com) cuya fragilidad hemos apreciado en los últimos años, aunque probablemente el fracaso de las empresas más aventureras no implique la aniquilación del medio. ¿O sí? Porque parece cundir la sensación de que Internet sea territorio apache, zona peligrosa donde cada uno debe velar por su propia seguridad o está muerto. Así pues la pregunta: ¿es Internet digna de confianza?

Hay muchas expectativas en lo que se podría hacer, algunas puro cuento de la lechera y otras con un claro recorrido. Que las relaciones entre empresas se realicen por vía electrónica es un hecho aceptado, con muchos años a sus espaldas, y donde Internet no abre camino, sino que avanza un paso más en una dirección consolidada. Lo que sí es novedad relativamente reciente es la aparición de los individuos como actores de las transacciones electrónicas: el individuo que compra, el individuo que vende, el individuo que se relaciona como ciudadano con su administración pública. Estos individuos pueden ser consumidores de servicios o prestadores de servicios, en un sentido muy general, tanto de compra-venta como de intercambios de opinión o de bienes en formato electrónico (por ejemplo, música). No es cuestión aquí de

La confianza y la seguridad aspectos vitales para los servicios electrónicos

Este artículo fue seleccionado para la monografía «e-AA.PP.» (Novática 162, marzo-abril de 2003) pero no pudo publicarse por falta de espacio.

entrar en las innumerables posibilidades de comportamiento, unas tradicionales, otras imprevisibles, como corresponde a sujetos creativos en terreno inexplorado.

Lo que aquí nos preocupa es la caracterización de estos individuos como sujetos activos y como sujetos pasivos que se mueven en este territorio con más o menos confianza, ofreciendo más o menos confianza a los demás. ¿Podemos fiarnos de la gente? ¿De quién y de qué se fía la gente?

2. ¿Qué es confianza?

Definir «confianza» no es trivial. ¿A qué se refiere la gente cuando habla de confianza? La misma Real Academia Española, punto obligado de partida, abre más puertas que cierra.

confianza. (De confiar).

1. Esperanza firme que se tiene de alguien o algo.
2. Seguridad que alguien tiene en sí mismo.
3. Presunción y vana opinión de sí mismo.
4. Ánimo, aliento, vigor para obrar.
5. Familiaridad (en el trato).
6. Familiaridad o libertad excesiva. Ú. m. en pl.
7. V. Abuso de confianza
8. Pacto o convenio hecho oculta y reservadamente entre dos o más personas, particularmente si son tratantes o del comercio.

de confianza

1. Dicho de una persona: Con quien se tiene trato íntimo o familiar.
2. Dicho de una persona: En quien se puede confiar.
3. Dicho de una cosa: Que posee las cualidades recomendables para el fin a que se destina.

en confianza

1. Confiadamente.
2. Con reserva e intimidad.

La confianza es un sustantivo, es un adjetivo y es parte de numerosas locuciones o formas de hablar. La confianza es un estado subjetivo, con más o menos fundamento, que nos lleva a tomar decisiones de actuación moduladas en su alcance y forma. Si algo no inspira confianza lo evitamos, o nos atrevemos timoratamente y con cautelas; en todo caso cuando no hay confianza no hay velocidad de crucero. Y al revés, cuando hay confianza nos lanzamos en la fe de que no habrá problemas o, si los hubiera, tendrían solución pronta y adecuada. Es en cierta medida la confianza la medida de la carencia de riesgo o, apurando las palabras, la medida del control del riesgo pues los sujetos convivimos con el riesgo bajo control (la conducción de coches) y huimos del riesgo descontrolado (la guerra).

En actividades tradicionales, la confianza es parte de la educación y convivimos inconscientemente con ella, tomando decisiones continuamente sin un análisis cerebral puntual. En territorios inexplorados, sin referentes claros (de confianza, y lamento el círculo vicioso), necesitamos hablar de los procesos de creación y destrucción de la confianza. Y como la confianza tiene un grado, tendremos que saber qué la incrementa y qué la socava.

No todos los sujetos parten del mismo punto. Tendremos visionarios, tendremos innovadores, tendremos una masa sin criterio y tendremos tradicionalistas. Y la diferencia es de tiempo, ritmo u oportunidad. Los visionarios aceptan un riesgo elevado, bien por inconsciencia, bien por una esperanza desmedida en el beneficio potencial. Las primeras incursiones en nuevos terrenos las protagonizan estos visionarios que exploran lo desconocido y emiten mensajes para los demás; mensajes en general muy subjetivos, bien ensalzando los éxitos (los padres de los nuevos servicios electrónicos) bien definiendo criterios de valoración (los académicos) bien generando criterio (los intelectuales). El futuro de una tecnología puede verse seriamente afectado por la experiencia y dotes de comunicación de este segmento de visionarios que pueden lanzar cohetes (fenómeno punto.com) o derruir la confianza. Es difícil recuperar la confianza tras una mala experiencia inicial.

Si los visionarios abren brecha, los innovadores están dispuestos a la aventura a poco fundamento que se le vea y pocos mensajes positivos lleguen de los visionarios. Firms creyentes de que el que da primero tiene una ventaja competitiva, conscientes de que el tiempo es oro y aspirantes cautos a medallas; todos ellos avanzan rápidamente sobre el precario camino abierto por los visionarios que, a estas alturas, probablemente estén en otras aventuras.

No está claro quién escucha a los innovadores, porque la siguiente remesa de usuarios, la masa, se caracteriza por su pasividad (dicho sin ánimo despectivo; simplemente sus intereses inmediatos están en otra parte). Si el mensaje boca a boca animó a los innovadores es porque estaban esperando una excusa; pero la masa lo que necesita son incentivos o, en términos de comunicación, una comercialización de la actividad (o sea, marketing). En cada actividad concreta la masa suele ser mayoría, sus movimientos lentos, para bien o para mal, para mayor o menor confianza (que es lo que nos ocupa). La masa se mancha de aceite lentamente, y también lentamente se lava los fracasos.

Y nos queda el segmento que podríamos llamar tradicional a ultranza que presenta una resistencia activa a la innovación y sólo se apunta a «lo de toda la vida» porque cualquier otra aproximación a la nueva actividad le resulta sospechosa e indigna de confianza. Hay múltiples ejemplos en la historia de la ciencia en la que las nuevas teorías jamás han convencido a los viejos barones, llegando al extremo de requerir cambios generacionales para que triunfe la novedad. Desde Galileo hasta Einstein, por citar personas libres de toda sospecha de irracionalidad. Simplemente, no lo entienden.

La confianza social o sentir popular de la confianza que merece una innovación será la media ponderada (de alguna forma) de

la confianza que impera en estos diferentes grupos sociales. Hay un efecto de arrastre (liderazgo) y hay un efecto de empuje (marketing), ambos coloreados por las experiencias buenas o malas que se van conociendo y aireando. Podemos decir que confianza es creer saber lo que va a ocurrir. El punto de partida es diferente para cada uno de los grupos caracterizados en los párrafos anteriores, punto a partir del cual crece o disminuye, al principio con fuertes oscilaciones, y con el tiempo de forma perezosa.

La confianza crece con la satisfacción de las expectativas. Cuando uno cree que va a ocurrir esto, y es esto justamente lo que ocurre. Cada vez que alguien hace lo que promete, cada vez que funciona un servicio por Internet y, encima, funciona como parecía entenderse en las páginas de anuncio. Cada vez que un producto defectuoso se ve inmediatamente reemplazado por otro impecable. Cada vez que un servicio deficientemente prestado se ve prontamente reparado. Porque la confianza es mayor sin fallos; pero se rehabilita si son pocos y de pronta recuperación. Y la confianza crece cuando otros, desinteresadamente, hablan bien y te cuentan su experiencia positiva y te transmiten confianza. Los visionarios se animan solos, una minoría. Los innovadores se animan con los visionarios. A la masa hay que empujarla poco a poco. Y los recalitrantes no salen en la foto.

La confianza disminuye con cada sorpresa, sobre todo con las negativas, con los fallos de las expectativas. Cuando uno visita una página web y le desconcierta un anuncio (*pop up*), cuando un correo infecta con un virus; cuando una factura se sale del tiesto; cuando un proceso de compraventa no lleva a ningún término, o termina en un producto muy por debajo de las promesas, o de imposible reposición; cuando un servicio se presta incorrectamente o simplemente se suspende de su ejecución a medio camino, incompleta.

En páginas web todos estamos hartos de anuncios emergentes, de propaganda agresiva, de correos basura, de enlaces engañosos, de enlaces rotos, de contenidos obsoletos, de informaciones no contrastadas, de un largo etcétera que se ve facilitado por la novedad y falta de regulación del medio, así como por la carencia de criterio de los usuarios junto con la ausencia de reglas y leyes que marquen la frontera entre lo correcto y lo incorrecto; es decir, porque no sabemos las reglas del juego y somos víctimas (o creemos serlo, que a efectos de la confianza hay poca diferencia) del medio o de la otra parte (que a efectos de la confianza, tampoco supone gran diferencia).

La confianza es pues subjetiva y hay que cuidarla y mimarla. La confianza es individual y colectiva y el que aspire a merecerla la necesita colectiva e individualmente. La confianza necesita referentes, marcas y saber popular, sobre todo en el segmento de la masa. Pero este es un argumento delicado porque no hacemos sino transferir la confianza en una marca a las cosas que la ostentan y nos lleva a hablar de transferencia de confianza y a hablar de origen de la confianza, lo que nos introduce en el bonito mundo de la lógica matemática y a tecnologías que instrumentan la transferencia fiable de confianza.

Y me gustaría acabar estos intentos de intentar contextualizar la confianza con una mención a la lógica y a las paradojas. La

confianza crece cuando el comportamiento es lógico; pero no en el sentido matemático, sino en el sentido visceral. Una página web que asegure que sólo dice mentiras (lo que es falso porque lo que dice es verdad) sólo encandilará a los matemáticos viciosos; pero seguro que espanta al común de los mortales que busca seguridad, no que le tomen el pelo (por muy meritoria del Nobel que sea la genialidad).

3. ¿Qué es seguridad?

Definir «seguridad» tampoco es trivial. Como en el caso del término «confianza», la misma Real Academia Española, abre más puertas que cierra. ¿A qué se refiere la gente cuando habla de seguridad?

seguridad. (Del lat. securitas, -atis).

1. Cualidad de seguro.
2. Certeza (conocimiento seguro y claro de algo).
3. Fianza u obligación de indemnidad a favor de alguien, regularmente en materia de intereses.

Seguridad jurídica.

1. Cualidad del ordenamiento jurídico, que implica la certeza de sus normas y, consiguientemente, la previsibilidad de su aplicación. En España es un principio constitucional. **Seguridad social.**

1. Organización estatal que se ocupa de atender determinadas necesidades económicas y sanitarias de los ciudadanos.

de seguridad.

1. Dicho de un ramo de la Administración Pública: Cuyo fin es el de velar por la seguridad de los ciudadanos. Agente de seguridad.
2. Dicho de un mecanismo: Que asegura algún buen funcionamiento, precaviendo que este falle, se frustre o se violente. Muelle, cerradura de seguridad.

No está la seguridad lejos de la confianza. De hecho es fácil decir que «seguridad es confianza en lo que va a ocurrir» y «confianza es estar seguro de lo que va a ocurrir», siendo la falta de sorpresa el fundamento común de ambas.

Pero mientras la confianza es terreno de trabajo de psicólogos y sociólogos, la seguridad aparece como «más técnica», afrontada por matemáticos e ingenieros que buscan modelos, tecnologías y soluciones para fundar y medir la seguridad de los productos y los sistemas. En la línea abierta por Sir Isaac Newton de que el tratamiento matemático permite modelar la realidad y por tanto simularla y predecir lo que va a ocurrir. No siempre con lógica cartesiana, sino a veces con razonamientos difusos; pero siempre con una capacidad de tratamiento analítico impersonal.

Un tratamiento analítico de moda es el análisis del riesgo como métrica del estado de seguridad. La idea es simple, cuantifiquemos e interpretemos. El qué es más complicado pues se trata de estimar estados indeseables (de inseguridad o de carencia de seguridad). Un análisis de riesgo parte de una catalogación de activos cuyo valor nos permite comparar, ordenar y priorizar. Catalogar activos es una tarea laboriosa, sobre todo si buscamos una aproximación detallada (en oposición a una aproximación holística). Cualquier proceso humano en general, y de servicios electrónicos en particular, está tan imbricado con otros procesos, actores y componentes que un planteamiento de laboratorio es a la vez imposible y necesario. Imposible porque toda simplificación

falsea la realidad; y necesario para poder aprehender la realidad. No quiero perder al lector en disquisiciones; estoy hablando de que hay medios físicos (tan básicos como la electricidad), elementos humanos (tan necesarios de tomar en consideración como los administradores, los gerentes y los piratas informáticos), elementos informáticos (equipamiento, sistemas operativos, aplicativos de consumo, aplicaciones a medida, bases de datos, etc.) y procesos de negocio (relaciones entre empresas y con el cliente final). Y hasta aquí podemos saber lo que nos cuesta reponer un componente defectuoso (incluido despedir a un empleado y formar a otro nuevo), lo que nos cuesta una multa de la Agencia de Protección de Datos y el lucro cesante de una parada en la cadena de producción. Y aún así tendríamos que cuantificar activos intangibles como la credibilidad del mercado y la capacidad de existir tan siquiera. Quizás estamos hablando de cuantificar la confianza de la otra parte, con lo que entramos de nuevo en un círculo vicioso.

La valoración de activos puede ser muy precisa (valor económico) o puramente relativista para centrarnos en lo que percibimos como importante cuando no como crítico. Pero en todo caso no vamos a limitarnos a una visión pasiva, como espectadores, sino que vamos a entrar en un juego de identificación de posibles amenazas que se ciernen sobre aquellos activos y cuya materialización va a suponer un impacto a evaluar. Esta evaluación se suele llamar riesgo, se evalúa en base a estimaciones de la posibilidad de que ocurra y el daño que causaría sobre nuestros preciosos activos, para concluir en una métrica que puede ser económicamente precisa (pérdida estimada) o simplemente relativa (elementos críticos).

El hecho de llegar hasta aquí ya es un gran paso hacia una mayor confianza, en muchos frentes: del inversor en su aventura empresarial, del cliente en su proveedor, del socio en sus interlocutores, del ciudadano en su administración. ¿Por qué? Porque al otro lado tenemos un profesional que sabe lo que tiene entre manos.

Pero tampoco echemos las campanas al vuelo a la primera buena noticia. Ni el responsable de seguridad ni el auditor se van a contentar con tener una foto, salvo que esta fuera idílica. Típicamente habremos detectado una serie de puntos críticos de alto riesgo que requieren un tratamiento inmediato o un plan para afrontarlos. El tratamiento puede suponer la eliminación de la causa, o la suscripción de un seguro que nos ampare por si lo que pudiera ocurrir, ocurre. O bien entraremos en una dinámica de mitigación del riesgo, desplegando contramedidas. Hay contramedidas meramente organizativas (como controles de personal, conciliación de datos parciales,...) y otras de tipo técnico (control de acceso, criptografía,...) Hay contramedidas que dicta el sentido común o la práctica habitual del sector: se supone que las hemos tomado; no haberlo hecho será un serio motivo de desconfianza pues la otra parte se sentirá engañada en sus suposiciones. Hay contramedidas que impone la ley o reglamentos sectoriales; ahora ya no es sólo desconfianza puntual, sino que puede suponer una penalización económica que probablemente busque ser ejemplar pues minar la confianza en un sector o en la Justicia puede suponer un serio quebranto en la confianza colectiva.

En algunos casos las contramedidas no son ni baratas ni evidentes, y habrá que hacer más números hasta ver en que momento el coste de la contramedida se ve adecuadamente recompensado en la reducción del riesgo. Si eso no se consiguiera, más vale cambiar de negocio.

Es habitual que las empresas e instituciones alardeen de sus contramedidas de seguridad, dando por supuesto que han analizado sus riesgos y saben lo que hacen. El mensaje que se transmite es puro marketing: confíe en nosotros por todas estas razones.

Y para terminar esta sección, pongámonos en el otro lado y preguntémonos sobre cómo valorar ese mensaje de seguridad. ¿Cómo y en qué medida convertimos unas garantías de seguridad en una cuantía de confianza? Pues salvo los especialistas con tiempo libre para pensar y calibrarlo todo reposadamente (o sea, salvo nadie), todos los demás recurriremos a referentes sectoriales o generales. Hay sellos de calidad que se emiten tras una auditoría (que es un proceso pautado, reglamentado y ejecutado por expertos) y que reconocen formalmente el valor del estado de riesgo.

Y existen certificaciones emitidas por laboratorios, con el mismo fin. Informalmente podemos decir que se certifican productos y se auditan sistemas u organizaciones; pero no quisiera entrar en un terreno abonado a la polémica. Basta citar el esfuerzo de los Criterios Comunes de Certificación de la Seguridad para destacar la ímproba tarea que supone acordar a nivel mundial unos criterios objetivos de medición y aseguramiento de la calidad con el ánimo, evidentemente, de ser eficaces en un mundo que es cada día más global.

Aunque, digamos lo que digamos, y con todos los sellos y certificados que queramos lanzar al público para alimentar su confianza, esta se demuestra en última instancia andando, pues el tiempo pone a cada uno en su justo lugar y medida; lo que no pretende ser un mensaje derrotista, sino un aviso a navegantes de que hay que estar siempre alerta y que mientras las metodologías de análisis de riesgo y las certificaciones de seguridad consoliden su valor final, deberemos practicarlas de regularmente, incorporando nuevas funciones, amenazas y contramedidas.

4. Tecnología

No creo que se pueda escribir nada sobre confianza y seguridad sin mentar la tecnología, aunque el mundo tecnológico es actualmente más fácil que el de las sensaciones personales y sociales. Y no porque sea la tecnología sencilla, sino porque es predecible. Y ¿alguien no confía en las matemáticas?

La tecnología lo impregna todo. Los ordenadores (como activos de producción) cambiaron el mundo hace ya cincuenta años y son actualmente parte vital de los sistemas. Internet ha cambiado el perímetro de las organizaciones, abriendo un amplísimo y aún en exploración universo de amenazas remotas. La misma tecnología cierra muchas posibles amenazas con un buen diseño (componentes, compartimentos estancos, controles de acceso, etc.) y permite el contraataque reactivo para perseguir el delito.

Esta tecnología de comunicaciones y sistemas informáticos es la que constituye los servicios electrónicos en un sentido muy general y es en la que estamos construyendo la confianza. Los visionarios de los centros de investigación ya casi están en otras cosas y los innovadores viven tiempos duros de presupuestos y sorpresas. Estamos transitando hacia las masas que, como miden las encuestas, es cada vez más usuaria de Internet, al tiempo que va renqueando la confianza en el medio y en los compañeros de viaje. Hay, como en todas partes, buenos y malos; buenos (en los sentidos de buena intención y de competencia tecnológica) que van poco a poco contribuyendo a la confianza. Y hay malos (en los sentidos de mala intención e incompetentes tecnológicos) que la minan de forma espectacular.

El medio, Internet, es demasiado prometedor como para admitir que vaya a fracasar; pero el camino va a ser duro pues si el medio es débil a la hora de imponer calidades (todo es intercambiable por medio de bits), entonces necesitaremos construir la confianza en niveles superiores de servicio. Esto no es novedad para los ingenieros: hace siglos que se construye en el agua, en terrenos inconsistentes y en zonas de actividad sísmica. Simplemente hay que saber hacerlo.

5. Transferencia de confianza

Por esto, buscando seguridad sobre bases inseguras, no es de sorprender que la criptografía como tecnología y los servicios web como organización, busquen aunarse para ofrecer servicios de fiables donde la confianza que nos ofrecen algunos actores (pongamos el Banco de España) se pueda aplicar con garantías en los servicios (pongamos transferencias económicas).

Hay cosas sencillas como garantizar por medio de cifrado el secreto de las comunicaciones, lo que ya es un pilar importante pues crea una red (virtual) entre puntos de confianza. Los desarrollos de la criptografía en este terreno han sido no sólo espectaculares, sino además efectivos y estamos en la situación idílica del viejo Arquímedes: «*Dadme un punto de apoyo y moveré la Tierra*» que en criptografía se lee: «*dadme una clave segura y nadie interceptará vuestras comunicaciones*». El talón de Aquiles se llama ahora gestión de las claves.

Quizás la tecnología más inquietante, por su potencia, novedad y dificultad de uso práctico, sea la firma electrónica basada en la famosa criptografía de clave pública. La capacidad individual de crear firmas que protejan la integridad de los datos y sean imputables al signatario, es fácilmente asimilable al invento de los contratos por escrito que revolucionó el comercio hace miles de años. Alguno dirá, y no le falta razón, que consolidar el valor probatorio de un contrato escrito ha costado unos miles de años y aún es motivo diario de controversia judicial entre sujetos, empresas, instituciones e incluso estados. Habrá que tener paciencia y cuidar el crecimiento de tan potente herramienta. A veces el término «contrato» hay que entenderlo en un sentido muy general: documentos firmados, declaraciones, acusos de recibo, etc. Todo aquello que permita imputar al signatario a través de una fácil verificación de firma.

Porque la firma electrónica se puede verificar con cierta facilidad; facilidad que no es tanta si trabajamos fuera de línea (*off line signature verification*); pero sí es realmente fácil si contamos con la disponibilidad de las comunicaciones. La verificación puede ser humana (en relaciones interpersonales, comerciales o administrativas) o automática (en sistemas robotizados y procesos de negocio), lo que permiten diseñar procesos distribuidos (o sea, relaciones entre dos o más participantes).

El talón de Aquiles de la firma electrónica probablemente esté en su valoración: ¿qué valor tiene? o, en términos de riesgo ¿qué riesgo mitiga? ¿cuánto mitiga los riesgos que mitiga? ¿qué nuevos activos y nuevas amenazas introduce? ¿cuánto riesgo introduce? Son todas preguntas que se prestan a respuestas gloriosas (cuentos de la lechera o afirmaciones de escaso fundamento) por visionarios; pero que los innovadores han descubierto que cuesta lograr que esta tecnología valga efectivamente lo que promete teóricamente.

De nuevo hay que pedir paciencia y experiencia hasta que calibremos la aportación de la firma electrónica a la confianza. En esta línea se sitúa la legislación relativa a firma electrónica, buscando un terreno de juego fiable (seguridad jurídica) con fundamentos técnicos en el origen (prestadores acreditados de servicios de certificación) y en los medios (dispositivos seguros de firma).

6. Origen de confianza

La firma electrónica es un instrumento de transferencia de confianza. ¿En qué?

La primera aplicación que se ha anunciado con oropeles ha sido la certificación de identidad que se debe leer como que alguien conocedor de la identidad física o jurídica de otro, reconoce y firma para que así conste un certificado que podrá ser usado por el portador donde le convenga; es decir donde haya confianza y se lo admitan. Confianza en la tecnología y en el signatario.

Una segunda aplicación, aún en estado incipiente, es el aval de potestades. Lo mismo que una llave habilita al portador a abrir una puerta, o un cheque personal habilita a una persona a su cobro, o un poder habilita la firma de un apoderado para firmar por la empresa, los llamados certificados de atributos permiten al beneficiario alguna actividad electrónica. Hay certificados nominales que requieren el concurso de un certificado de identidad (por ejemplo el reconocimiento como médico colegiado) y hay certificados al portador (como una tarjeta de control de acceso al edificio). Todo tipo de variantes que en realidad no imponen limitaciones a la creatividad de los que diseñan procesos de negocio e imponen reglas de avance (la firma de tres miembros de los veinte miembros del consejo de seguridad; un millón de firmas de ciudadanos mayores de edad; ...).

La confianza en el signatario es muy interesante por cuanto un signatario ampliamente conocido puede avalar a un colectivo amplio de usuarios. Así Verisign certifica a millones de personas y servidores web en Internet, y el Estado Español planea certificar a todos los ciudadanos con el llamado DNI

electrónico. Los colegios profesionales, las cámaras de comercio, las asociaciones internacionales, etc. etc. son posibles generadores de certificados de identidad o de capacidad. Y los ejemplos están elegidos para destacar la colisión que va a aparecer entre la certificación pública y la iniciativa privada, cuyo desarrollo ya veremos con el tiempo; pero de momento podemos centrarnos en la confianza que inspira el emisor del certificado (o proveedor de servicios de certificación si queremos reconocer que esto de los certificados es un conjunto de servicios y no un producto). La firma es mero vehículo.

Lo malo es que la firma del certificador es también electrónica y entramos en un círculo vicioso de aquellos de quien verifica al verificador. Pregunta que es tan básica y elemental como la de por qué nos fiamos de las conclusiones de las matemáticas: porque llevan años acertando prácticamente siempre. Es decir, confianza ganada día a día durante siglos. Lo malo es que ahora tendremos que fiarnos de los ordenadores, con un penoso historial: llevan decenas de años fallando“ todos los días.

No es por ser catastrofista; sólo destacar que la calidad de los sistemas, necesaria para confiar en la firma, aún es claramente mejorable y, probablemente, jamás llegue a la difícilmente superable cota alcanzada por las matemáticas; pero seguro que aprendemos a vivir en un mundo razonablemente seguro, simplemente cuando no haya sorpresas, los fallos no sean catástrofes, el riesgo esté bajo control y equilibremos el riesgo con el beneficio. Esta cuantificación justa de la confianza que nos merece la tecnología de firma electrónica es la que aún ignoramos.

7. Para concluir

Hemos intentado en los párrafos anteriores entender qué es la confianza, que es un valor percibido por los participantes y que modula la disponibilidad de las partes para involucrarse en un servicio electrónico. Sin confianza no hay servicios; pero la confianza depende de las personas y de los colectivos, hay que crearla, mimarla y empujarla.

Desde un planteamiento más técnico hemos visto cómo medir y certificar la seguridad y hemos divagado sobre algunas tecnologías que prometen sustentar una elevada confianza. Son tecnologías que se proponen a sí mismas como habilitadoras de los servicios electrónicos, y sin duda lo son; pero aún necesitan otro hervor. Hablamos de la prometedor firma electrónica y los difusos servicios de certificación de identidad y de capacidades.

Todo junto podemos resumirlo en que estamos viviendo unos tiempos interesantes y que con Internet muchas cosas ya no serán igual; siempre y cuando haya confianza en la seguridad que ofrece.

Sistemas de Tiempo Real

Javier Miqueleiz Álamos

Grupo de Redes, Sistemas y Servicios Telemáticos,
Departamento de Automática y Computación, Universidad
Pública de Navarra

<javier.miqueleiz@unavarra.es>

Resumen: Linux implementa el estándar POSIX para soft real-time, pero no proporciona un buen servicio a las aplicaciones de tiempo real. Muchos desarrolladores han modificado el kernel para mejorar esta situación. En este artículo se describirán las extensiones de Linux para tiempo real que existen a día de hoy.

Palabras clave: extensiones, kernel, Linux, tiempo real.

1. Introducción

Los sistemas operativos de tiempo real existen comercialmente desde hace años. Se caracterizan por proporcionar buen rendimiento, un entorno completo de desarrollo y buena asistencia técnica. Sus aspectos negativos son un coste alto y su naturaleza *closed source*.

Las variantes de Linux para tiempo real ofrecen una alternativa *open source* a las soluciones comerciales. En este trabajo se van a describir sus objetivos, garantías ofrecidas y características.

2. RTLinux

RTLinux (*Real-Time Linux*) ofrece prestaciones de *hard real-time*, por lo que es adecuado para tareas con límites temporales estrictos.

RTLinux consiste en un microkernel sobre el que se ejecutan tareas de tiempo real. El kernel Linux está implementado como una tarea de prioridad mínima y no tiene control directo sobre el bloqueo de las IRQs. Cuando hay tareas de tiempo real que deben ser ejecutadas, el microkernel interrumpe la ejecución de Linux. El mecanismo para conseguir esto es una técnica de emulación software del controlador de interrupciones.

La filosofía de RTLinux es separar por completo la funcionalidad de tiempo real de la de propósito general, ya que sus principios de diseño son contrapuestos. En la parte de propósito general se optimizan los casos comunes a costa de empeorar los casos peores. En la de tiempo real importa el determinismo y no la rapidez de respuesta. En opinión de los desarrolladores de RTLinux, un sistema que incorpore ambas funcionalidades no será eficiente en ninguna de ellas.

Las tareas de RTLinux no puede hacer un uso directo de los servicios del kernel, ya que no es posible que realicen llama-

Sistemas Linux de tiempo real

das al sistema. Lo que sí es posible es un uso indirecto si los hilos de tiempo real se comunican con procesos Linux. Para realizar esto, RTLinux incorpora dos mecanismos: la memoria compartida y las colas FIFO de tiempo real. Las garantías que se obtienen con este uso indirecto son las habituales de los procesos Linux.

3. RTAI

El enfoque de diseño de RTAI (*Real Time Application Interface*) es similar al de RTLinux: tratar a Linux como una tarea de prioridad baja que no tiene control directo sobre el programador de interrupciones. El mecanismo para conseguirlo es diferente en los detalles de implementación. RTAI utiliza una capa HAL (*Hardware Abstraction Layer*) que simplifica el interfaz entre el microkernel de tiempo real y el kernel Linux. La función de esta capa es sustituir las llamadas a ciertas funciones del kernel por las llamadas propias de RTAI. La ventaja de esta técnica frente a la de RTLinux es una implementación más clara y sencilla.

Las diferencias más importantes entre RTAI y RTLinux radican en las funcionalidades que proporcionan. La filosofía de los desarrolladores de RTAI es implementar muchas funcionalidades con el fin de que los usuarios no echen en falta ninguna. Por el contrario, los diseñadores de RTLinux prefieren ofrecer pocos servicios y poner énfasis en la corrección y la rapidez.

RTAI proporciona garantías de *hard real-time*, por lo que resulta adecuado para el mismo tipo de aplicaciones que RTLinux. En sus primeras versiones presentaba los mismos problemas que aquél para la implementación de aplicaciones multimedia. Sin embargo, en las versiones más recientes se ha añadido el módulo LXRT (*Linux Real-Time*), que incluye funcionalidades muy interesantes:

- Proporciona una API simétrica, que se utiliza de forma idéntica desde los módulos del kernel o los procesos de usuario.
- Permite acceder a los servicios de RTAI en modo de usuario sin renunciar a la realización de llamadas al sistema y con rendimiento de *firm real-time*.
- Proporciona prestaciones de *hard real-time* desde procesos de usuario, evitando de esta forma los problemas de incompatibilidad de versión de los módulos del kernel.

A modo de conclusión, la orientación de RTAI es la misma que

la de RTLinux, pero sus desarrolladores han optado por añadirle más funcionalidades. RTAI es más flexible y puede dar garantías de *firm real-time* en espacio de usuario sin renunciar a la realización de llamadas al sistema.

4. KURT

Los desarrolladores de KURT (*Kansas University Real-Time*) han pretendido implementar en Linux características de tiempo real para que pudiera ser usado en algunos de sus proyectos de investigación. Su objetivo no ha sido diseñar una extensión para tiempo real de propósito general, sino que han buscado soluciones *ad hoc*.

La clase de servicio que proporciona KURT es de *firm real-time*. Para conseguir el aumento de la resolución temporal con respecto al kernel Linux, KURT funciona sobre el paquete UTIME. Este paquete reprograma dinámicamente la frecuencia con la que interrumpe el reloj HW y proporciona resolución de microsegundos. Puede funcionar en dos modos, periódico y no periódico. En el primero de ellos la frecuencia de interrupción es fija, resultando adecuado para aplicaciones periódicas. En el segundo de los modos se programa el reloj para que interrumpa en los momentos en que hay eventos de tiempo real que deben ser atendidos. Si no hay ningún evento a atender, el reloj interrumpe con la frecuencia habitual: 100 Hz. Este esquema dinámico permite ahorrar tiempo de CPU cuando la resolución alta no es necesaria. El modo de funcionamiento periódico tiene como ventaja que no hay costos de CPU asociados a la reprogramación de la frecuencia del reloj.

KURT consiste básicamente en un planificador específico para tiempo real. Puede funcionar en tres modos distintos: normal, *mixed* y *focused*. El primero de ellos es equivalente a utilizar el kernel Linux normal, con la diferencia de que se tiene la resolución temporal alta de UTIME. Los dos últimos son los modos específicos de tiempo real. El modo *mixed* permite que los procesos normales coexistan con los de tiempo real. En el modo *focused* sólo se planifican los procesos de tiempo real, por lo que es adecuado para aplicaciones con requisitos exigentes.

El planificador de KURT es del tipo *time-driven* y se puede utilizar de dos formas distintas. En la primera de ellas se debe proporcionar un fichero con información sobre los tiempos en los que debe empezar la ejecución de los procesos o módulos de tiempo real. Esto resulta adecuado cuando las tareas a ejecutar son conocidas a priori y es factible el cálculo *off-line* de los tiempos de inicio de ejecución. La segunda de las formas es adecuada para procesos periódicos. Cada uno de éstos le comunica al kernel sus características para que éste realice un control de admisión. Los parámetros que le pasa al kernel son el periodo y el tiempo de CPU que se necesita por periodo.

A nivel interno, KURT se divide en dos partes: el núcleo KURT y los módulos KURT. El núcleo se encarga de invocar a los módulos para que éstos ejecuten las tareas de tiempo real. Hay un módulo especial, llamado Process RTMod, que se encarga de pasar el control a los procesos de usuario. La **figura 1** describe la relación entre los distintos componentes de KURT.

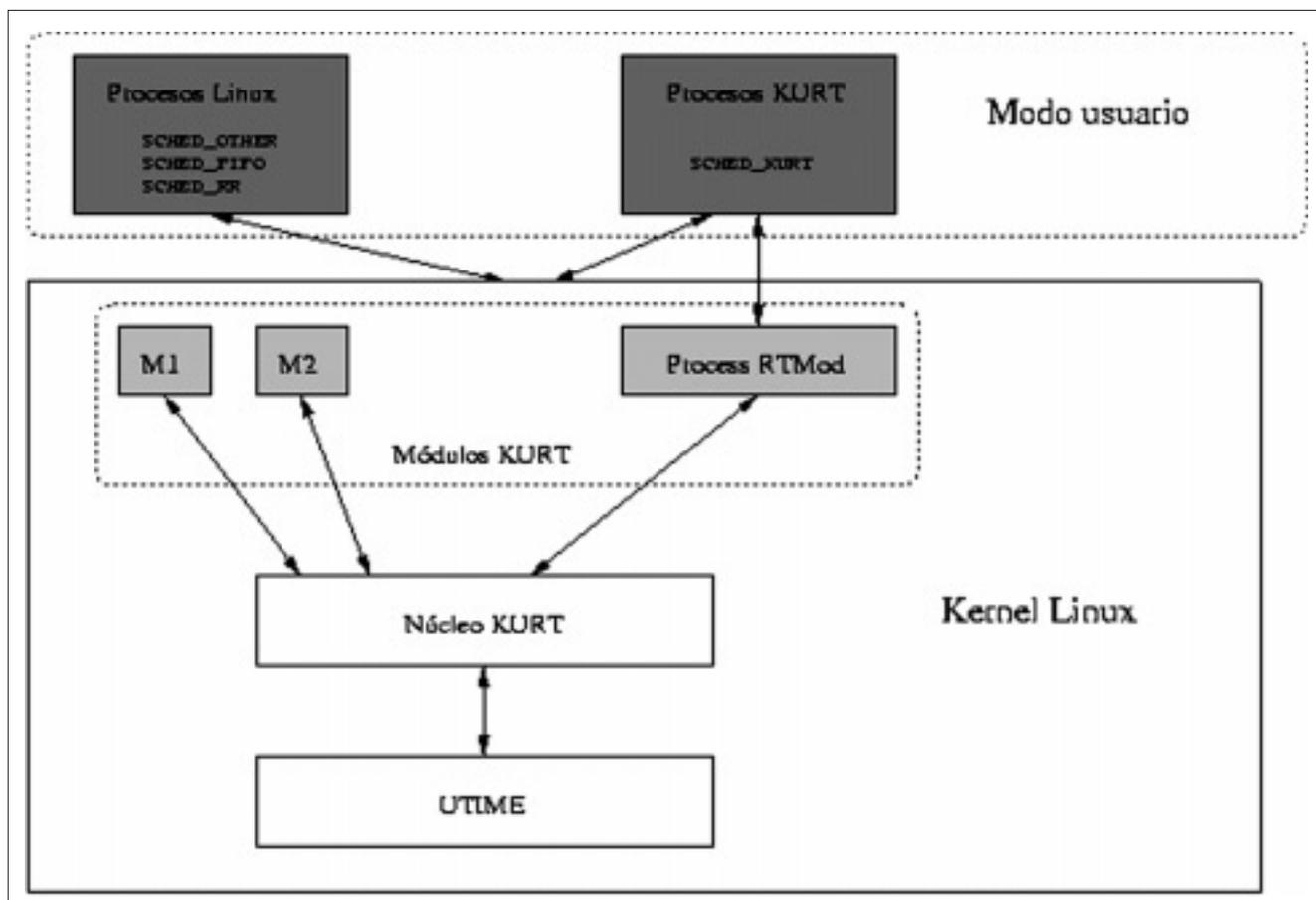


Figura 1. Arquitectura de KURT

A diferencia de RTLinux y de RTAI, en KURT la funcionalidad de tiempo real está integrada en el kernel, por lo que tanto los procesos como los módulos pueden utilizar todos sus servicios. Es preciso tener en cuenta que el rendimiento de KURT no es igual de bueno que el de las dos variantes de *hard real-time*. Las razón de esto es que KURT utiliza subsistemas del kernel Linux no diseñados para tiempo real. Estos subsistemas son el origen de distorsiones en los tiempos de planificación de las tareas.

5. RED-Linux

RED-Linux es el acrónimo de *Real-Time and Embedded Linux*. Consiste en un kernel Linux normal que ha sido modificado para mejorar sus propiedades con las aplicaciones de tiempo real. Las garantías que ofrece son de *firm real-time*. RED-Linux es compatible con las aplicaciones tradicionales y, al igual que en KURT, los procesos de tiempo real pueden solicitar cualquier servicio al kernel.

Sus autores han analizado las causas por las que Linux no es adecuado para tiempo real y han propuesto e implementado soluciones. Las funcionalidades implementadas han sido las siguientes:

- *Microtimer*: en un SO de propósito general como Linux, se utiliza un temporizador periódico para repartir la CPU entre los procesos. El periodo por defecto es de 10 ms, que es demasiado alto para un SOTR. En un sistema de este tipo resulta imprescindible tener un temporizador con resolución de microsegundos.

- Emulación software de las interrupciones: en Linux, cuando la CPU detecta una interrupción, pasa a atender la rutina de servicio a la interrupción. Este proceso requiere tiempo de CPU y puede afectar a las tareas de tiempo real. Los desarrolladores de RED-Linux han implementado un mecanismo de emulación software de interrupciones basado en el de RT-Linux. Su funcionamiento es el siguiente. Cuando se produce una IRQ, no se atiende la interrupción, sino que simplemente se registra el evento en una tabla. Posteriormente se comprueba si es necesario atender interrupciones y, en caso afirmativo, se ejecutan las rutinas de servicio. Este mecanismo permite reducir el tiempo de respuesta de las tareas de tiempo real.
- Puntos de interrupción: en el kernel Linux, la ejecución de una llamada al sistema es un proceso atómico que no puede ser interrumpido por ningún proceso. Esto se debe a que el kernel no es reentrante, por lo que no es posible detener la ejecución de una llamada al sistema para dar la CPU a un proceso de tiempo real, ya que éste puede invocar en su ejecución la misma llamada al sistema. Las consecuencia de esto es un incremento del tiempo de respuesta de las tareas de tiempo real, ya que éstas deben esperar a que termine la ejecución de las llamadas al sistema. Un posible solución a este problema sería convertir al kernel Linux en reentrante. Esto implicaría un tiempo de interrupción del kernel muy bajo, pero supondría un trabajo largo y difícil. Los desarrolladores de RED-Linux han optado por añadir al código de las llamadas al sistema puntos de interrupción. A grosso modo, su funcionamiento es el siguiente.

El código de las llamadas al sistema se divide en bloques más

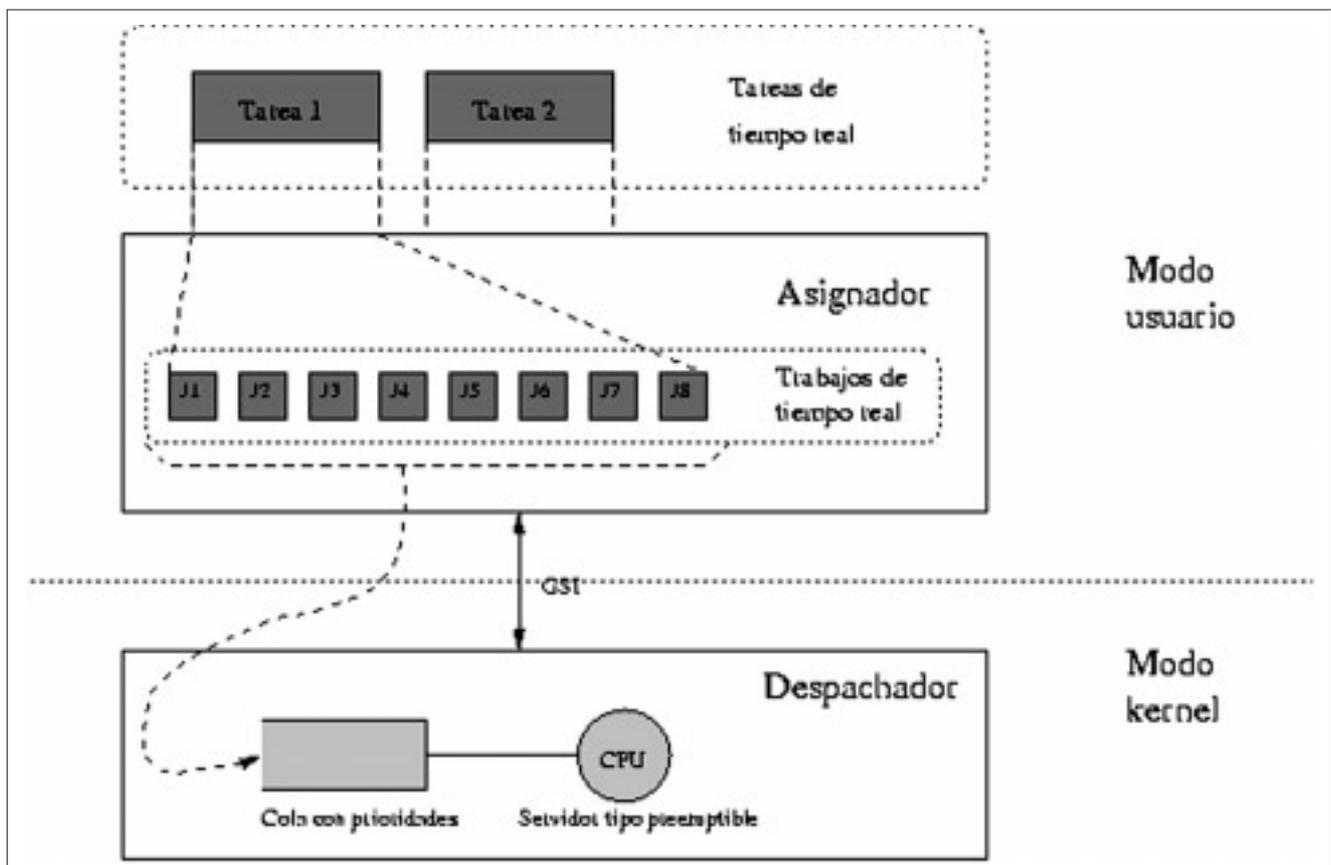


Figura 2. Planificador de RED-Linux

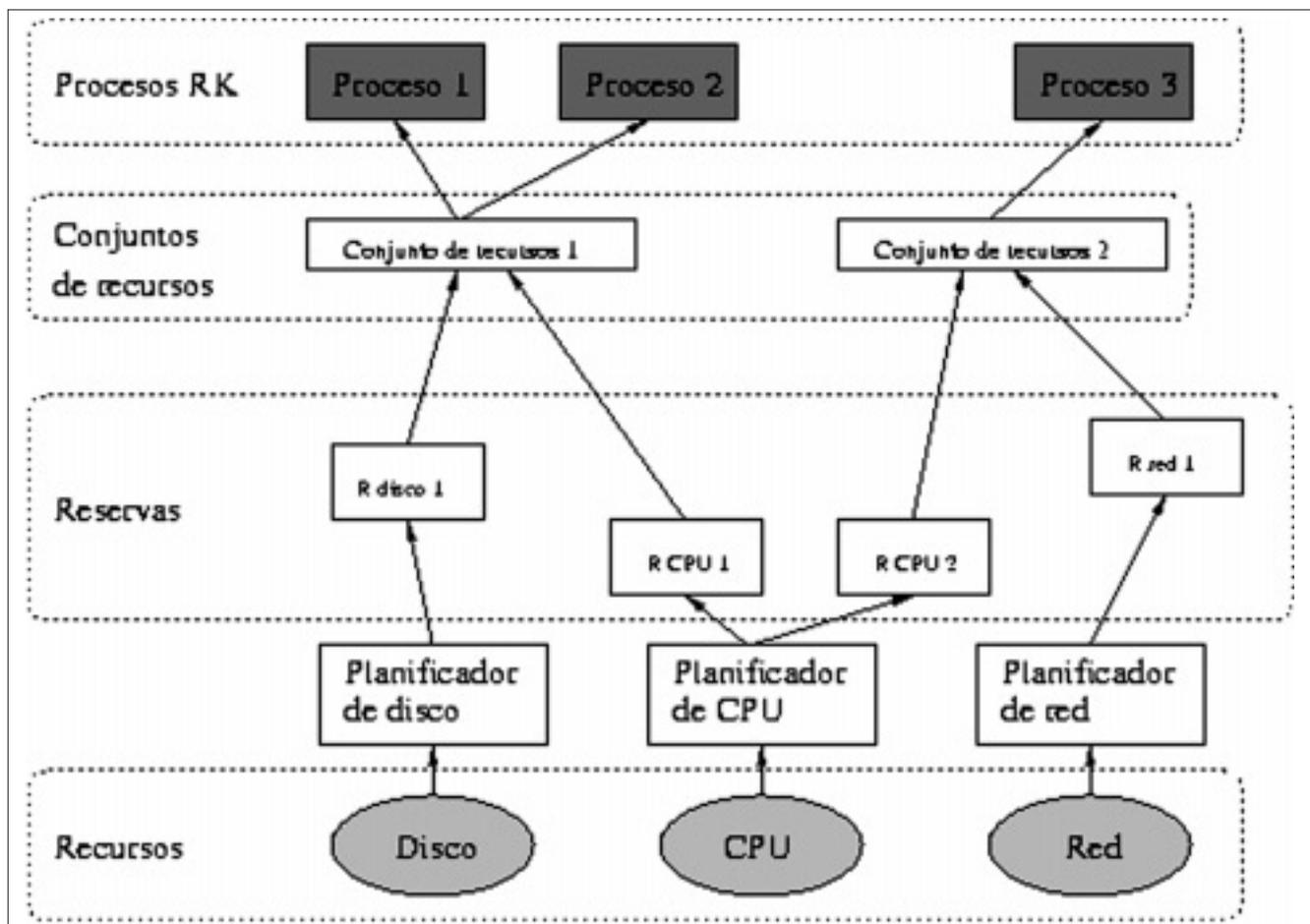


Figura 3. Arquitectura de Linux/RK

pequeños. Entre cada uno de esos bloques se inserta un punto de interrupción, es decir, un lugar en el que es seguro interrumpir la ejecución de la llamada y pasar a ejecutar un proceso más prioritario. En cada punto se comprueba si hay eventos de tiempo real que atender y si los hay, se invoca al planificador. Si no los hay, el *overhead* que introduce cada punto es muy pequeño, ya que sólo consisten en una comparación. Cuando se devuelve el control a la llamada que había sido interrumpida, ésta debe volver a comenzar en un punto seguro, pues no hay garantía de que los recursos compartidos del kernel no hayan sido modificados. La adición de estos puntos permite reducir el tiempo de interrupción del kernel. El valor máximo corresponde al tiempo que se tarda en ejecutar el bloque más largo.

La característica más novedosa de RED-Linux y que lo hace único es su planificador. En los SOTR habituales se ofrecen unos pocos algoritmos de planificación que están implementados en el kernel (o microkernel). La mayoría de esos algoritmos pertenecen al tipo *priority-driven*. Los desarrolladores de RED-Linux han pretendido que en él se puedan implementar la mayoría de los algoritmos de planificación, ya sean del tipo *time-driven*, *share-driven* o *priority-driven*.

En RED-Linux, el planificador está dividido en dos componentes, el despachador y el asignador (figura 2). El asignador divide las tareas en trabajos, asigna a cada trabajo cuatro parámetros de planificación y pasa el trabajo al despachador.

El despachador guarda en una cola los trabajos y los planifica teniendo en cuenta los parámetros que le pasa el asignador. Los cuatro parámetros de planificación son los siguientes:

1. Prioridad: define la importancia de cada trabajo y tiene la misma semántica que en otros planificadores.
2. Tiempo de comienzo: es el instante en que un trabajo puede comenzar a ejecutarse.
3. Tiempo de finalización: es el plazo de vencimiento del trabajo. Si este tiempo se supera, su ejecución debe ser detenida.
4. Cantidad: es la cantidad de tiempo de CPU (u otros recursos) que necesita el trabajo.

Además de dar valores a estos cuatro parámetros, el asignador también elige el orden de evaluación de éstos. El despachador ordena los trabajos según el orden establecido por el asignador y planifica el primero de ellos. El asignador determina el algoritmo de planificación y puede ser modificado para adaptarse a las necesidades del usuario. El despachador, en cambio, es siempre el mismo y no necesita ser modificado.

En la versión actual de RED-Linux, el despachador está implementado mediante un módulo del kernel y el asignador mediante un proceso de usuario de prioridad alta.

De momento, el desarrollador de aplicaciones debe ocuparse de la programación del asignador. La intención de los desarrolladores de RED-Linux es ofrecer en un futuro una

librería de asignadores en la que estén incluidos los algoritmos de planificación más conocidos.

6. Linux/RK

Linux/RK es el acrónimo de *Linux/Resource Kernel*. El equipo desarrollador de este sistema pertenece al Laboratorio de Tiempo Real y Multimedia de la Universidad de Carnegie Mellon (EE.UU.). Linux/RK está diseñado para ser útil en aplicaciones multimedia y de tiempo real.

El primer concepto que es necesario definir para describir Linux/RK es el de kernel de recursos. Es un kernel que proporciona a las aplicaciones un acceso temporal, garantizado y controlado a los recursos del sistema. Los programas de aplicación deben solicitar al kernel reservas de recursos. Si las solicitudes son admitidas, el kernel garantiza que se recibirán de manera exclusiva las cantidades solicitadas del recurso.

Una de las virtudes de este tipo de kernels es que las aplicaciones se deben ocupar únicamente de especificar sus necesidades temporales. La gestión de los recursos es responsabilidad exclusiva del kernel. Los algoritmos que utiliza internamente el kernel pueden ser cambiados sin necesidad de modificar las aplicaciones.

RK portable es un kernel diseñado para añadir al kernel de un

SO habitual las funcionalidades propias de un kernel de recursos. En realidad, no se trata de un kernel, sino de un subsistema que interactúa con el kernel normal. Su implementación no utiliza ninguna característica de un kernel concreto, por lo que es portable a cualquier plataforma. El único de sus componentes que debe ser adaptado es el encargado de implementar el interface con el kernel normal. Al conjunto formado por Linux y la implementación de RK portable en este SO se le denomina Linux/RK.

La API de Linux/RK está basada en el manejo de reservas y conjuntos de recursos. Una reserva es una cantidad de algún recurso del sistema que se destina para el uso exclusivo de una aplicación. Un conjunto de recursos agrupa varias reservas.

La **figura 3** describe gráficamente la relación entre aplicaciones, reservas, conjuntos de recursos y recursos. En el nivel inferior se encuentran los recursos del sistema. Cada uno de ellos es gestionado por un planificador, que se encarga de que a cada proceso se le dé la cantidad del recurso que ha solicitado. Cada proceso puede estar enlazado con un conjunto de recursos como máximo. Es posible enlazar varios procesos con un solo conjunto. Esto resulta interesante en aplicaciones que desempeñan una función mediante múltiples procesos.

En los recursos que son multiplexados temporalmente, la API

	Servicio	Paradigma	Llamadas al sistema	Microtimer	Tareas periódicas	Otras características
ART-Linux	Hard	Priority	No	Sí	Sí	
Fairsched	Soft	Share	Sí	No	No	Planificador HSFQ
Hard Hat Linux	Soft	Priority	Sí	No	No	Kernel interrumpible
KURT	Firm	Time	Sí	Sí	Sí	
Linux/RK	Firm	Priority	Sí	Sí	No	
Linux/RT	Firm	Priority	Sí	Sí	Sí	
Linux-SRT	Soft	Share	Sí	No	No	
QLinux	Soft	Share	Sí	No	No	Planificador HSFQ
RED-Linux	Firm	Priority, Share y Time	Sí	Sí	Sí	Planificador genérico, puntos de interrupción
RTAI	Hard y Firm	Priority	Sí (LXRT)	Sí	Sí	
RTLlinux	Hard	Priority	No	Sí	Sí	

Tabla 1. Características de las variantes para tiempo real del kernel Linux

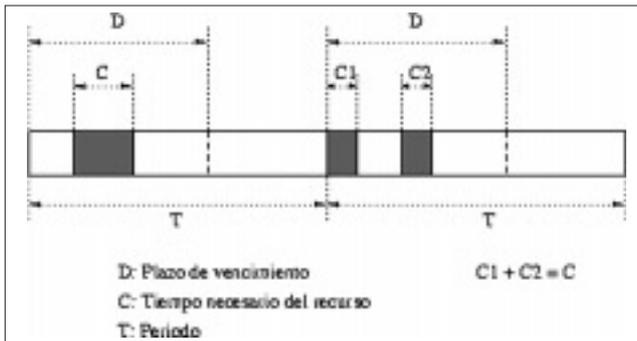


Figura 4. Parámetros de los recursos multiplexados en el tiempo

de Linux/RK permite especificar tres parámetros: T, C y D (figura 4). C es la cantidad de recurso que se debe proporcionar en cada periodo de duración T. D es el tiempo límite (referido al comienzo del periodo) en el que los recursos deben haber sido proporcionados. La implementación actual de Linux/RK únicamente permite realizar reservas de CPU. Los planes para el futuro son añadir las reservas de BW de red y de BW de disco.

7. Conclusión

En este trabajo se han descrito algunas de las extensiones para tiempo real del kernel Linux existentes en la actualidad. Existen otras variantes que no han sido tratadas y que se van a citar a continuación: ART-Linux, Fairsched, Hard Hat Linux, Linux/RT, Linux-SRT, QLinux y el *patch* de baja latencia de Ingo Molnar.

Ninguna de las extensiones descritas o citadas es lo suficientemente flexible como para dar servicio a todas las aplicaciones. Sin embargo, constituyen un punto de partida muy valioso e interesante.

La **tabla 1** sintetiza las características de las diferentes variantes. La columna «Servicio» especifica las garantías que proporciona cada variante. «Paradigma» se refiere al paradigma

de planificación utilizado. La posibilidad de realizar llamadas al sistema desde las tareas de tiempo real aparece indicada en la columna «Llamadas al sistema». El campo «Microtimer» especifica si la variante utiliza algún mecanismo para aumentar la resolución temporal. La columna denominada «Tareas periódicas» se refiere a si en la API se proporcionan funciones específicas para las tareas periódicas. Por último, en «Otras características» aparecen particularidades propias de cada variante.

Referencias

1. Pierre Cloutier, Paolo Mantegazza, Steve Papacharalambous, Ian Soanes, Stuart Hughes. «DIAMP-RTAI Position Paper, Nov. 2000». *Proceedings of the Real-Time Systems Symposium*, noviembre de 2000.
2. Robert Hill, Balaji Srinivasan, Shyam Pather, y Douglas Niehaus. «Temporal Resolution and Real-Time Extensions to Linux». Technical Report ITTC-FY98-TR-11510-03, Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Sciences. University of Kansas, junio de 1998.
3. David Ingram. «Integrated Quality of Service Management». Dissertation Submitted for the Degree of Doctor of Philosophy. Jesus College, University of Cambridge, agosto de 2000.
4. Douglas Niehaus, William Dinkel, Sean B. House. «Effective Real-Time System Implementation with KURT Linux». *Real-Time Linux Workshop*, Vienna, Austria, diciembre de 1999.
5. Suichi Oikawa, y Ragnathan Rajkumar. «Portable RK: A Portable Resource Kernel for Guaranteed and Enforced Timing Behavior». *Proceedings of the Real-Time Technology and Applications Symposium*, junio de 1998.
6. Suichi Oikawa, y Ragnathan Rajkumar. «Linux/RK: A Portable Resource Kernel in Linux». *Proceedings of the Real-Time Systems Symposium*, 1998.
7. Balaji Srinivasan, Shyamalan Pather, Robert Hill, Furquan Ansari, Douglas Niehaus. «A Firm Real-Time System Implementation Using Commercial Off-The-Shelf Hardware and Free Software». *Proceedings of the Real-Time Technology and Applications Symposium*, junio de 1998.
8. Yu-Chung Wang, y Kwei-Jay Lin. «Enhancing the Real-Time Capability of the Linux Kernel». *Proceedings of the RTCSA*, octubre de 1998.
9. Yu-Chung Wang, y Kwei-Jay Lin. «Providing Real-Time Support in the Linux Kernel». *Proceedings of the Real-Time Technology and Applications Symposium*, 1999.
10. Yu-Chung Wang, y Kwei-Jay Lin. «Some Discussion on the Low Latency Patch for Linux». *Real-Time Linux Workshop*, noviembre de 2000.
11. Victor Yodaiken. «The RTLinux Manifesto». Department of Computer Science, New Mexico Institute of Technology, Socorro, New Mexico.
12. Victor Yodaiken, Michael Barabanov. «RTLinux Version Two». VJY Associates LLC, 1999.

Referencias autorizadas

Las habituales referencias que desde 1999 nos ofrecen los coordinadores de las Secciones Técnicas de nuestra revista pueden consultarse en <<http://www.ati.es/novatica/lecturas.html>>

Sección técnica «Administración electrónica» (Gumersindo García Arribas, Francisco López Crespo)

Para inaugurar esta sección técnica informamos sobre el Congreso Internacional de Formación de Altos Funcionarios Públicos ha celebrado su XVII edición los días 11 a 13 de junio de 2003 en la sede de Madrid del Instituto Nacional de Administración Pública, bajo el lema «Reforma administrativa y Funcionarios Directivos» <http://www.inap.map.es/inapnew/cci/act_form/cifafp.htm>.

Uno de sus grupos de trabajo, presidido por **Jorge Barallat Quincoces**, funcionario del Ministerio de Administraciones Públicas, se ocupó de la Administración Electrónica. En sus conclusiones presentó un decálogo de factores esenciales para desarrollar con éxito una Administración electrónica, cuya versión traducida al castellano es la siguiente:

- 1) Es esencial y de importancia capital un fuerte liderazgo político que promueva y respalde los cambios.
- 2) El foco de la Administración electrónica son los usuarios, ciudadanos o empresas. Son necesarias medidas de estímulo para fortalecer la demanda, como por ejemplo proporcionar puntos públicos de acceso a Internet
- 3) Es preciso promover una revolución cultural en las administraciones públicas, para lo cual serán necesarias actuaciones formativas, que también habrán de considerar a los usuarios, ciudadanos o empresas.
- 4) Debe prestarse especial atención a la accesibilidad, integrando toda clase de plataformas en el servicio al ciudadano, como aspecto nuclear para evitar la marginación de grupos sociales.
- 5) La cooperación debe extenderse a todos los niveles de las Administraciones Públicas, Europea, Central, Regional o Local, con el fin de proporcionar servicios y procedimientos multiadministración integrados. En estas actuaciones deben estudiarse posibles sinergias con el sector privado.
- 6) Utilizar la tecnología para cambiar los procedimientos de trabajo a través de los diferentes niveles de la administración, con el fin de mejorar la eficiencia en la provisión de servicios.
- 7) Internet y las tecnologías relacionadas son herramientas poderosas para «repensar» y reformar las administraciones públicas. Es necesario revisar los procedimientos actuales desde la perspectiva electrónica.
- 8) Debe procurarse el equilibrio armónico entre las actuaciones tecnológicas internas (*back-office*) y las externas, (*front-office*).
- 9) Debe prestarse atención a los cuatro grandes cambios que provoca la Administración electrónica: a) estratégico, puesto que cambia la relación con los ciudadanos; b) funcionales, porque diseña nuevos procedimientos administrativos; c) organizativos, ya que provoca cambio cultural y precisa de nuevos perfiles profesionales en la carrera funcional y d) tecnológicos, ya que la estandarización y la interoperabilidad adquieren protagonismo principal.
- 10) La Administración Pública del siglo XXI o será electrónica o no será capaz de realizar sus cometidos.

Tras escuchar lo manifestado en el Congreso, cabe una reflexión personal de los editores de esta sección la progresiva automatización de los procesos administrativos va a continuar a ritmo creciente, de la mano de la buena aceptación social de la que goza. Para manejar la situación creada, las Administraciones Públicas han de reforzar la presencia de directivos de tecnología (en el sentido de *chief information officer*, según la denominación de la Administración Federal estadounidense), de profesionales bien preparados y de directivos y funcionarios que puedan liderar y aprovechar el cambio en beneficio de los ciudadanos.

Sección técnica «Auditoría SITIC» (Marina Touriño, Manuel Palao)

Tema: Libro

Reingeniería de la Auditoría Informática y su orientación a objetivos

estratégicos de negocio, Gustavo Adolfo Solís Montes (CISA), Grupo Cynthus S.A. 1999..

Dice Gustavo Adolfo Solís en su introducción al libro «*La lectura del presente libro requiere una actitud abierta a nuevas ideas y conceptos que, en última instancia, no buscan atacar los modelos existentes, sino fundamentar un proceso evolutivo que permita a la auditoría satisfacer la demanda de servicio que existe en nuestra cambiante sociedad*».

Su contenido se presenta de tal forma, en un orden lógico, coherente y de progresión gradual, que permite profundizar en la calidad de las tareas que desarrolla un auditor de sistemas de información. Tal como su autor anticipa no es una ruptura con el pasado sino que ahonda en el conocimiento de la materia, con la sólida base de una experiencia de años de ejercicio como auditor, que permite observar el grado de madurez que ha adquirido esta profesión.

En la parte uno, **Fundamentos**, se analiza sistemáticamente a las organizaciones y a los sistemas de información, definiendo a estos como «*el conjunto de datos, de elementos humanos, de procedimientos de trabajo y de tecnología que en forma coordinada y alineada a una estrategia institucional, proporciona soporte a la operación, a la toma de decisiones y al servicio de los clientes de una empresa*». A partir de aquí se indaga sobre el factor crítico que la tecnología de la información, representa para una organización, y la visión de la *auditoría de la tecnología de la información*, que una actitud proactiva, puede convertir en un factor de valor agregado para las organizaciones.

Dentro de la disquisición sobre el trabajo de una auditoría de tecnología de la información, se hace referencia a la naturaleza ecléctica de esta actividad: «*El desarrollo de un trabajo de auditoría en tecnología de información requiere de profundos conocimientos en distintas disciplinas, que en forma coordinada permitan obtener los resultados deseados. Normalmente, este tipo de auditorías son desarrolladas por equipos multidisciplinarios...*».

En esta parte también se abordan aspectos imprescindibles, hoy en día, sobre «governance», que define como el «*sistema mediante el cual las corporaciones son dirigidas y controladas para lograr sus objetivos de negocio*», sobre la visión estratégica del control interno, la relación entre este y el auditor, y la relación entre los objetivos de control y los objetivos de negocio, como los factores críticos de éxito de una organización. Acostumbrados a libros de auditoría de sistemas de información o informática que «descuidan» el carácter contributivo de la tecnología de la información, como una pieza más del total de elementos que utiliza una organización para el logro de sus fines, este libro aporta criterios sustanciales al auditor para que su trabajo redunde en beneficios reales para la organización.

Asimismo, en la primera parte se incluyen análisis de los procedimientos de control, y de control interno en Tecnología de la Información, que recapacita perspicazmente sobre la aplicación práctica de estos temas, y demuestra que los controles internos, no son elementos de «cortar y pegar». No se encuentran aquí «recetas magistrales», sino un desglose cualitativo y reflexivo. Estos argumentos y principios se completan con una análisis sumamente interesante de COBIT.

En la segunda parte, **Metodología**, el libro aporta distintos enfoques sobre este tema, y queda patente, luego de su lectura, que la auditoría de sistema de información no se «resuelve» con un mero cuestionario o *checklist* y unas cuantas entrevistas, y que no existen conclusiones *pret a porter*, frente a determinadas deficiencias en abstracto: «*el éxito de una auditoría no es el volumen de su informe ni la gravedad de sus hallazgos... ni el volumen es necesariamente representativo de la calidad del trabajo del auditor ni del nivel de su aportación a la empresa*».

En conclusión, la lectura de este libro es muy recomendable tanto para los auditores con muchos años de experiencia como para aquellos que comienzan su andadura en esta profesión, ya que contribuye con creces al «proceso evolutivo» de la auditoría de sistemas de información, con reflexiones profundas y sustentadas sobre esta actividad. Su lectura también puede ser de gran utilidad para otro tipo de profesionales de TI que quieran alinear su trabajo

con el control estratégico, y por ende contribuir de forma cualificada al logro de los objetivos de la organización. (Este libro se encuentra referenciado en la librería, *bookstore*, de la ISACA <http://www.isaca.org/bk_none.htm#2-spm>).

Tema: *Sistemas Expertos y ASITIC*

Los Sistemas Expertos (SE), aparentemente, están de baja forma, ante la pujanza de otras técnicas más «tecnológico-intensivas», como las redes neuronales y otras técnicas de *data-mining*. Carecemos de datos objetivos para probarlo (por eso decimos ‘aparentemente’), pero basta «estar en la página» para percibir la tendencia. (Esas modas raramente son ajenas a intereses comerciales). Sin embargo, los SE han desempeñado y están llamados a desempeñar un papel substancial en muchas áreas de los negocios y los servicios, y entre éstos en la ASITIC.

La ASITIC es un coste (justificado, si está bien planteada). Toda medida de seguridad, todo ‘control’, es --a corto plazo-- un extracoste, en el que se incurre, en la confianza de evitar otro mayor. Los costes de control son costes ‘hundidos’ (se incurre en ellos siempre); los costes que pretenden obviar son ‘contingentes’ (suceden si suceden: los siniestros).

La ASITIC no es una disciplina fácil de dominar. No sólo requiere un amplio conocimiento teórico, sino (como, acertadamente, exigen las especificaciones CISA <<http://www.isaca.org/cisa.htm>> una amplia experiencia), a la postre, la ASITIC se basa en ‘buenas prácticas generalmente aceptadas’ y en el criterio general del auditor.

Por ello, los SE pueden ser de gran utilidad en: i) la ayuda al desempeño de ASITIC con una amplia y variada carga de cometidos (p. ej., en grandes ‘conglomerados’ [agrupaciones no sectoriales] de empresas); ii) la formación/ ayuda al desempeño de ASITIC jóvenes / en formación; iii) las grandes firmas, con un número considerable de juniors y una alta rotación; y iv) los auditores / firmas que operan en grandes territorios de gran extensión y baja densidad de población (Canadá, Australia, Sudáfrica).

El Prof. Dr. Miguel Ángel Ramos (UCIII) fue precursor entre nosotros, hace años, con su tesis doctoral sobre SE aplicados a la ASITIC. Las siguientes URLs contienen información sobre el tema que nos ocupa:

- <<http://www.google.com/search?hl=en&lr=&ie=UTF-8&q=%22information+system+audit%22+expert+system%22&spell=1>>
 - <http://www.franz.com/success/customer_apps/finance/pricewaterhousecoopers.lhtml>
 - <<http://www.cisdept.csupomona.edu/gallegos/msbaprojects/msbap8892.html#20-1988>>

Tema: «*Texticulillos*» (redactado por Marina Touriño)

Según el DRAE, la relación encontrada más cercana a la palabra «Texticulillo», es: “*textual: Dicho de una persona: Que autoriza sus pensamientos y los prueba con lo literal de los textos, o expone un texto con otro*». Pero no es el origen o el significado de esta referencia, su aspecto clave, inventada o recreada por Manuel Palao para identificar y asociar, estos artículos de opinión sobre la auditoría de sistemas de información.

Sea cual fuese la idea de su autor en la creación de esta nomenclatura, lo importante es destacar lo oportuno de estos textos orientados a crear debate sobre temas relacionados con la auditoría de los sistemas de información, y con el tema, estrechamente relacionado con el anterior, de la gestión de la tecnología de la información y las comunicaciones.

Estos artículos, 49 hasta el momento, se pueden leer en la página <<http://www.auditoresdesistemas.com>>, y abarcan aspectos muy variados: desde el control interno de las TIC (*Juvenal y la Auditoría de Sistemas*), hasta los planes de continuidad del negocio (*Plan de continuidad de la Cadena Logística y la Auditoría de Sistemas*), pasando por sutiles reflexiones sobre la preparación de los profesionales (*Norbert Wiener y la Auditoría de Sistemas*).

Su mérito reside también en que no son textos para «sentar cátedra», sino para, desde el conocimiento y la experiencia, exponer aspectos de la auditoría de sistemas, induciendo a la reflexión y, en ocasiones, a provocar la polémica, como un medio creativo para la mejora de la profesión de la auditoría de sistemas y del servicio de las TIC.

Sección Técnica «Derecho y Tecnologías» (Isabel Davara Fernández de Marcos, Isabel Hernando Collazos)

Tema: *Plan Nacional de nombres de dominio bajo «.es»*

En cumplimiento de lo previsto en la Disposición Adicional sexta de la Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico, se ha publicado la Orden CTE/662/2003, de 18 de marzo, del Ministerio de Ciencia y Tecnología por la que se aprueba el Plan Nacional de nombres de dominio de Internet bajo el código de país correspondiente a España («.es»). El Plan Nacional de nombres de dominio, que sigue con los criterios de asignación que se habían introducido mediante la Orden de 21 de marzo de 2000, por la que se regula el sistema de asignación de nombres de dominio de Internet bajo el código de país correspondiente a España (.es), modificada por la Orden de 12 de julio de 2001, introduce algunas novedades que tienen por objeto facilitar el acceso a los nombres de dominio bajo «.es» sin renunciar por ello a la seguridad que debe garantizarse en este ámbito, como la reducción de las restricciones aplicables a la asignación de nombres de dominio bajo «.es», o la ampliación de la legitimación y el tipo de nombres de dominio que pueden registrarse bajo este código de país y, además, la creación de nombres de dominio de tercer nivel, que son «.com.es», «.nom.es», «.org.es», «.gob.es» y «.edu.es». La orden está disponible en la siguiente dirección: <<http://www.boe.es/boe/dias/2003-03-26/pdfs/A11917-11924.pdf>>.

Tema: *Reglamento por el que se regula el Registro de la Propiedad Intelectual*

Mediante el Real Decreto 281/2003, de 7 de marzo, por el que se aprueba el Reglamento del Registro General de la Propiedad Intelectual (publicado en el Boletín Oficial del Estado núm. 75, de 28 de marzo), se ha desarrollado lo dispuesto en los artículos 144 y 145 del Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, al mismo tiempo que se deroga el Real Decreto 733/1993, de 14 de mayo, por el que se aprobaba el anterior Reglamento del Registro General de la Propiedad Intelectual. Entre las novedades, destaca la referencia al impacto que tienen las Nuevas Tecnologías también en este sector, y así se prevé la posibilidad de inscribir en el Registro obras o producciones multimedia y páginas web, exigiéndose en el caso de los programas de ordenador el depósito de todo el código fuente. Por último, cabe referirse también a la implantación de las Tecnologías de la Información y las Comunicaciones en lo que se refiere a la gestión del Registro, siendo posible por ejemplo el acceso informático para la consulta de los asientos, con valor meramente informativo.

El Real Decreto puede consultarse en la siguiente dirección: <<http://www.boe.es/boe/dias/2003-03-28/pdfs/A12145-12153.pdf>>.

Tema: *Tramitación del Proyecto de Ley General de Telecomunicaciones*

Tras su remisión por el Consejo de Ministros el pasado día 7 de marzo a las Cortes Generales, se ha publicado el Proyecto de Ley General de Telecomunicaciones (Boletín Oficial de las Cortes Generales, Congreso de los Diputados, serie A, núm. 133-1, de 14 de marzo), que está siendo tramitado. La futura Ley General de Telecomunicaciones, que derogará a la vigente Ley 11/1998, de 24 de abril, General de Telecomunicaciones, tiene por objeto transponer al Ordenamiento jurídico español el marco regulador de las comunicaciones electrónicas aprobado en Europa en abril del año pasado, dando respuesta al mismo tiempo a la necesidad de consolidación de la libre competencia en el mercado de las telecomunicaciones que es resultado del proceso que se inició el 1 de diciembre de 1998 como consecuencia de las exigencias impuestas por la normativa europea.

Entre los objetivos de la regulación que se está tramitando ahora cabe resaltar algunos aspectos tales como los que se refieren a la simplificación de los trámites administrativos necesarios para que una entidad pueda actuar como operador, o la ampliación de algunos derechos de los usuarios ya regulados, como la posibilidad de elegir el medio de pago para el abono de los servicios, y la introducción de otros nuevos, como por ejemplo el derecho a impedir mediante un procedimiento sencillo y gratuito la identificación de llamada y el derecho a no recibir publicidad, mediante llamadas automáticas, mensajes de fax o correo electrónico, introduciéndose asimismo un régimen sancionador más estricto, previendo nuevas infracciones y sanciones.

El proyecto puede consultarse en la siguiente dirección: <http://www.congreso.es/public_oficiales/L7/CONG/BOCG/A/A_133-01.PDF>.

Tema: Informe de la Comisión Europea sobre los servicios electrónicos de pago

Tal como se prevé en la Directiva 98/84/CE del Parlamento Europeo y del Consejo, de 20 de noviembre, relativa a la protección jurídica de los servicios de acceso condicional o basados en dicho acceso, la Comisión Europea ha elaborado un informe de su aplicación dirigido al Consejo, al Parlamento Europeo y al Comité Económico y Social Europeo. El objeto de la Directiva 98/84/CE es establecer un nivel mínimo de protección jurídica de los servicios electrónicos de pago en la Unión Europea contra la piratería, protegiendo así del acceso mediante dispositivos ilícitos a los servicios de televisión de pago, radio e Internet. Cabe resaltar la consideración de la piratería de los servicios electrónicos de pago como un delito informático, debiendo tener además en cuenta el carácter transfronterizo de este tipo de acciones con lo que ello supone para su perseguibilidad al tener que atender a diferentes Ordenamientos jurídicos.

El informe puede consultarse en la siguiente dirección: <http://europa.eu.int/comm/internal_market/en/media/condac/functioning/com2003-198/com-2003-198_es.pdf>.

Tema: Informe sobre el Anteproyecto de Ley de firma electrónica

Tras la publicación del segundo borrador de Anteproyecto de Ley de firma electrónica el pasado mes de julio de 2002 por el Ministerio de Ciencia y Tecnología en su página web, el Consejo de Ministros recibió el pasado día 4 de abril un informe sobre dicho Anteproyecto de Ley, que dará lugar a la aprobación de la Ley de firma electrónica que sustituirá al vigente Real Decreto-ley 14/1999, de 17 de septiembre, sobre firma electrónica. La futura regulación de la firma electrónica persigue extender su uso en las transacciones comerciales que se realizan a través de Internet y, al mismo tiempo, facilitar la realización de trámites con las Administraciones Públicas, convirtiéndose así en un instrumento fundamental para la Administración electrónica. En cuanto a las novedades que se introducen en el Anteproyecto de Ley de firma electrónica cabe destacar la inclusión de la firma electrónica reconocida, que es aquella firma electrónica que cumple con los requisitos necesarios para ser equivalente a la firma manuscrita por ser una firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma. Otra de las novedades más importantes es la implantación del Documento Nacional de Identidad electrónico, que permitirá identificar en el entorno electrónico a todos los ciudadanos y servirá para firmar documentos electrónicos sin necesidad de la firma manuscrita, ya que la firma electrónica que se efectúe mediante el Documento de Nacional de Identidad electrónico y usando un dispositivo seguro de creación de firma tendrá efectos jurídicos equivalentes a los de la firma manuscrita. Por último, también se prevé que las personas jurídicas puedan hacer uso de la firma electrónica al poder solicitar certificados electrónicos, dando así respuesta a la demanda del sector de los servicios de certificación electrónica.

El informe puede consultarse en la siguiente dirección: <http://www.setsi.mcyt.es/inic_legisla/firma_electr.htm>.

Sección Técnica «Enseñanza Universitaria de Informática»
(Joaquín Ezpeleta, Cristóbal Pareja Flores)

Tema: Desarrollo de programas correctos

John Barnes, High Integrity Software: The SPARK Approach to Safety and Security. Addison-Wesley, 2003. ISBN 0-321-13616-0. El libro está escrito por John Barnes (autor del «Programming in Ada 95» que todo el que haya programado alguna vez en Ada ha usado) por encargo de Praxis Critical Systems Limited, desarrolladores del entorno SPARK. Como se establece en el primer párrafo del prefacio, el libro «... is about programming in SPARK – a language highly suited for writing programs that need to be reliable». El «lenguaje SPARK» (como sus desarrolladores gustan denominar) está formado por un núcleo (un subconjunto de Ada 95) más un conjunto de anotaciones escritas como comentarios especiales. Estas anotaciones (tales como Pre y Post conditions de procedimientos y funciones, aserciones, como pueden ser invariantes de bucles o declaraciones de funciones para la verificación del programa) pueden ser utilizadas por las distintas herramientas del entorno (el «Examiner», el «SPADE Automatic Simplifier» y el «Proof

Checker») para verificar, a distintos niveles, la corrección del código (Ada) respecto a las especificaciones y anotaciones. El núcleo de Ada incorporado en SPARK es suficiente como para poder cubrir muchas de las necesidades en programación, pero no demasiado rico como para que resulte imposible hacer un análisis de la corrección de los programas escritos. Esto supone que el lenguaje no incorpora algunas herramientas fundamentales, como son punteros, tareas (aunque anuncian la inclusión de al menos un uso restringido de tareas para el futuro) o paquetes genéricos, además de imponer algunas restricciones en el uso de otros elementos del lenguaje (las cláusulas «use» no están permitidas, por ejemplo, pues pueden hacer muy complicado encontrar relaciones de dependencia entre datos).

El libro es, por un lado, una introducción detallada a SPARK; por otro, una descripción de las herramientas del entorno (que vienen en el CD que acompaña al libro).

La redacción del texto es clara, entretenida y cuidada, lo que hace que se trate de un libro ameno de leer. Adoptando la perspectiva de quien necesita (le gustan) los aspectos formales en el diseño y verificación de programas (ya sea por motivos docentes o profesionales), se trata de un libro de obligada lectura pues muestra un serio intento de plasmar en un entorno de desarrollo ciertos aspectos formales que, habitualmente, permanecen exclusivamente en el mundo académico.

Sección técnica «Informática y Filosofía»
(Josep Corco, Esperanza Marcos)

Tema: Libro

R. Hirschheim, H. K. Klein, K. Lyytinen. Information Systems Development and Data Modeling. Conceptual and Philosophical Foundations. Cambridge University Press, 1995. El texto realiza un estudio de los fundamentos conceptuales y filosóficos del modelado de datos y del desarrollo de Sistemas de Información (SI). Los autores proponen cuatro grandes paradigmas, así como la conexión filosófica a los mismos, a partir de los cuales se establece una clasificación de las metodologías para el desarrollo de SI. Se trata del paradigma funcionalista, conectado con posiciones filosóficas como el realismo y el positivismo, el relativismo social, relacionado con posiciones filosóficas posmodernas, segundo Wittgenstein y hermenéutica, el estructuralismo radical, vinculado al marxismo y lo que los autores llaman neohumanismo, basado en la filosofía de Habermas. Bajo estos cuatro taxones los autores agrupan distintas metodologías de desarrollo de SI. Los SI son entendidos en esta obra más como sistemas sociales basados en soportes técnicos, que como sistemas técnicos con incidencia social. Los autores conceden una importancia especial a la perspectiva filosófica (realista o relativista) cuando se trata de estudiar las metodologías de desarrollo de bases de datos, ya que éstas son entendidas como sistemas de representación del conocimiento.

Sección técnica «Ingeniería del Software»
(Javier Dolado Cosin, Luis Fernández Sanz)

Tema: Centro de recursos y referencias de Ingeniería del Software

1. *Webliography Software Engineering* <http://polaris.umuc.edu/~skerby/help/wbib_swe.htm> Este sitio es el clásico centro de recursos y enlaces dedicado a un tema que, en este caso, es el de la Ingeniería del Software. Tiene diversos apartados correspondientes a las principales áreas de la disciplina (Arquitectura, Estimación, Mantenimiento, Gestión de Configuración, Herramientas, etc.) así como referencias a otros centros de recursos de ingeniería de software en la Web. Aunque no todos los enlaces están actualizados y, en ocasiones, hay que hacer algunas investigaciones, la cantidad de información referenciada aconseja visitar esta página.

2. SEIR (*Software Engineering Information Repository*) <<http://seir.sei.cmu.edu/seir/seir-home.html>> Este anexo a la web del *Software Engineering Institute* proporciona a través de un sistema gratuito de registro el acceso a un conjunto variado de recursos que no son exactamente los mismos que están en la web principal del SEI, <<http://www.sei.cmu.edu>>. También permite contribuir al web aportando recursos.

Sección técnica «Interacción Persona-Computador»
(Julián Abascal González, Jesús Lorés Vidal)

Tema: Libros

Carolyn Snyder. Paper Prototyping: Fast and Simple Techniques for Designing and Refining the User Interface. Morgan Kaufmann;

(Abril 2003) ISBN: 1558608702. El prototipado es una de las actividades más importantes a la hora de realizar evaluaciones de la usabilidad dentro del desarrollo de sistemas interactivos siguiendo un modelo de proceso centrado en el usuario. El libro de Carolyn Snyder, de reciente aparición, trata en profundidad el prototipado sobre papel cómo una de las técnicas más baratas, rápidas y eficientes. El prototipado sobre papel ofrece, además, la ventaja de que, dada su sencillez, el usuario no se siente incómodo al expresar sus opiniones sobre el diseño. Coincidimos, pues, con Nielsen <<http://www.useit.com/alertbox/20030414.html>> en el interés por el libro ya que cubre un tema importante y poco tratado en la bibliografía.

Brinck Tom et al. *Usability for the web*. Morgan and Kaufmann. ISBN 1-55-860-658-0. Este libro está escrito por expertos en usabilidad con una amplia experiencia en el ámbito industrial en el desarrollo de sitios web. Está organizado alrededor de un modelo de proceso centrado en el usuario orientado al desarrollo de sitios web y distribuye los capítulos en las diferentes etapas en que se divide dicho modelo de proceso. En todos los capítulos se encuentran numerosos ejemplos que validan las propuestas. La lectura de cada fase/capítulo del modelo permite apreciar diferencias entre el desarrollo de una aplicación interactiva y el de un sitio web, tales como el análisis de la audiencia, las necesidades de los usuarios, la arquitectura de la información, etc. Recomendamos el libro como material básico de consulta.

Sección técnica «Internet» (Alonso Álvarez, Llorenç Pagés)

Tema: *El acceso a Internet está en el aire*

Esta no es una metáfora. Al igual que nosotros respiramos el oxígeno del aire, nuestros portátiles y dispositivos manuales se tornan, cada vez más, capaces de «alimentarse» a través de las microondas aéreas. No se trata de la lenta tecnología de comunicación GSM ni tampoco de la prometida 3ª Generación celular, sino de una tecnología basada en conjunto de protocolos (802.11x) para cuya mención conjunta se ha acuñado el término Wi-Fi (del inglés *Wireless Fidelity* o «fidelidad inalámbrica»), que permite velocidades de conexión superiores a las de la popular conexión telefónica ADSL (hoy en día, 11Mbps en el protocolo más usual) <<http://www.pcworld.com.pa/Articulos/mar2003/28-29-30.htm>>.

Este es el año de Wi-Fi <<http://www.baquia.com/com/20030117/bre00004.html>>. No solamente porque todo el mundo habla de ello, sino porque cada vez hay más personas que lo han probado y experimentado. Simplemente instalando una tarjeta Wi-Fi en su equipo (portátil) y localizando un 'punto de acceso' para conectarse. <http://www.noticiasdot.com/servicios/vocabulario/servicios_vocabulario_wifi.htm>.

En el mes de marzo, el Ayuntamiento de Zamora realizó una presentación en el Senado de su experiencia pionera en la instalación de una red inalámbrica de acceso a Internet en toda la ciudad <<http://es.news.yahoo.com/030320/4/2m920.html>>. Esto es fruto de una iniciativa privada de un grupo de empresas encabezada por Intel, habiendo sido Zamora la elegida precisamente por su baja implantación del acceso convencional a Internet <<http://www.afitel.com>>.

La principal conclusión de la experiencia de Zamora es que Wi-Fi es hoy en día una tecnología que puede funcionar muy bien para comunidades vecinales pero que su extensión a toda una ciudad presenta aún bastantes problemas <<http://www.afitel.com/espanol/prensa/wallsj.htm>>. Inconvenientes éstos que se vislumbran solucionables bajo el impulso de algunos de los grandes de la industria <<http://www.noticiasdot.com/publicaciones/2003/0503/0505/noticias050503/noticias050503-16.htm>>. El impacto en nuestro país va más allá, con todo, de iniciativas más o menos públicas ya que algunos operadores están ofreciendo como parte de su oferta de ADSL estaciones base con las que crear nuestra propia red en casa.

Pero es en EEUU donde, tanto porque la penetración de las tecnologías celulares es menor como por la fascinación de la gente hacia las nuevas tecnologías, Wi-Fi está pegando más fuerte. La cadena de cafeterías Starbucks ofrece ya conexión inalámbrica en más de 2.100 de sus establecimientos <<http://www.starbucks.com/retail/wireless.asp>> a cambio de una cuota mensual, mientras que McDonald's planea algo parecido aunque ofreciéndolo como servicio de un solo uso bajo una módica cuota horaria <<http://www.mcdonalds.com/countries/usa/whatsnew/pressrelease/2003/03112003/index.html>> (no deja de ser irónico fomentar la permanencia en locales diseñados precisamente para reducirla).

La fascinación por Wi-Fi se extiende en la Red y se cuentan historias como la de Mike Smart que ha hecho de la 'cafetería de la esquina' una extensión virtual de su oficina donde recibir a sus clientes <http://www.businessweek.com/magazine/content/03_17/b3830609.htm>

Pero en este entorno idílico, surgen también los motivos de controversia. Desde el momento, en que resulta tecnológicamente sencillo y no demasiado oneroso instalar antenas vecinales y 'colgarse' gratis de Internet, surgen movimientos reivindicativos apostando por una Internet rápida y gratuita en nombre de la innovación y del bien común. <<http://www.cnn.com/2002/TECH/11/21/yourtech.wifis/index.html>>

Y en relación a la dicotomía negocio de las empresas vs. universalidad del acceso a la Red, surge la preocupación de los empresarios que están invirtiendo en tecnologías celulares avanzadas (GPRS, UMTS) de que un firme y positivo avance en tecnologías Wi-Fi rivalice con los servicios que tienen planeado ofrecer. La discusión a este respecto está todavía muy abierta entre quienes piensan que el futuro está exclusivamente en Wi-Fi <<http://www.wired.com/wired/archive/10.10/wireless.html>> y quienes opinan que los servicios Wi-Fi no se van a superponer a los servicios celulares avanzados <http://www.businessweek.com/magazine/content/03_17/b3830612.htm>. Sin duda, el desenlace promete ser de lo más apasionante.

No se puede terminar esta breve reseña sin hacer mención a una de las leyendas más difundidas en torno a la tecnología: el uso de botes vacíos de Pringles como antenas direccionales <<http://www.oreillynet.com/cs/weblog/view/wlg/448>>. Este símbolo de la cultura nacida en torno a Wi-Fi es también un reflejo de que la tecnología no tiene que estar obligada a pasar por la sofisticación: perseverancia, imaginación y mucho ingenio son las verdaderas bases sobre las que se asienta su desarrollo.

Sección técnica «Lengua e Informática» (M. del Carmen Ugarte)

Miscelánea primaveral

Tema: *RAE. Diccionario de la lengua española (CD-ROM)*

Sin duda, la novedad más importante que nos ha traído la primavera en nuestra área ha sido la publicación en CD-ROM del *Diccionario de la lengua española* de la RAE (Madrid, Espasa Calpe, 21.ª edición, ISBN: 84-670-0317-0). La primera característica que conviene señalar es la de que pueda instalarse además de en entorno Windows, en Mac, Linux, Unix y en general en cualquier sistema operativo que disponga de entorno de ejecución de Java 1.1.8 o posterior. Este carácter multiplataforma, que no nos atrevemos a decir sea del todo novedoso, es sin duda el camino a seguir por las publicaciones electrónicas que normalmente venían restringiendo su instalación a Windows y como mucho a Mac. La integración con las aplicaciones ofimáticas de esos entornos es también posible por lo que permite su consulta dentro de esas herramientas.

En cuanto al manejo interno la interfaz permite búsqueda simple y avanzada, consultas dentro de los lemas, diccionario inverso, semejanza fonético-ortográfica y otra serie de utilidades que facilitan la búsqueda. Entre las facilidades se encuentra la introducción de notas personales, copia al portapapeles de la información, exportación para su incorporación a otros documentos, etcétera.

Tras su instalación, para su utilización no se requiere del CD-ROM original. En definitiva, a pesar de su precio (46 EUR) creemos que es un producto interesante que merece la pena considerar.

Tema: *Nuevo Corpus del español en la Web*

Desarrollado por **Mark Davies**, profesor asociado de lingüística hispánica en la Universidad del Estado de Illinois, cuenta con 100 millones de palabras. Se ha trabajado sobre 9314 obras y documentos, que cubren desde el año 1200 hasta nuestros días. Las búsquedas permiten varias posibilidades y combinaciones incluidas la de antónimos, sinónimos además de poder seleccionar categorías gramaticales.

El corpus puede consultarse en <<http://www.corpusdelespanol.org>>.

Tema: *Nuestra eñe*

Finalmente, y llegada desde Canarias, queremos recoger esta

iniciativa de publicación en la Web de forma totalmente gratuita, además de la edición convencional en papel, del libro Nuestra eñe, del joven estudiante de traducción **Ovidio Cordero**. Cordero recoge en él 51 artículos publicados en el semanario canario *Lancelot* desde el 2001, acerca de dudas sobre la lengua española muy al estilo de los *Dardos* de Lázaro Carreter. El material es variado y no siempre atiende a un interés del hombre de la calle sino del profesional de las letras, pero conviene echarle un vistazo por lo menos al índice pues puede resolverse algunas de esas «dudas tontas» que ni tan siquiera sabemos que las tenemos. El libro de 166 páginas puede descargarse en formato PDF desde <<http://www.lawebdelanzarote.com/usuarios/traduccion/archivos/deslibro.htm>>.

Sección técnica «Libertades e Informática» (Alfonso Escolano)

Tema: *los derechos de los pasajeros de líneas aéreas*

Desde comienzos del mes de marzo de 2003, Estados Unidos han estado teniendo acceso a la mayoría de las bases de datos de pasajeros de las líneas aéreas europeas. Un acuerdo entre la Comisión Europea y las aduanas de Estados Unidos permite la consulta en línea al PNR (Registro del nombre del pasajero) de todas las compañías europeas que vuelan a Estados Unidos. La UE y EEUU están negociando sobre la forma en que son procesados estos datos. En este entorno, España y Reino Unido han propuesto (ver <<http://www.statewatch.org/news/2003/mar/spainpassengers.pdf>>) una nueva directiva europea, que entre otras cosas imponen a las compañías aéreas la obligación de recoger datos de los pasajeros y enviarlas a las autoridades de los países destinos de los vuelos, y pueden ser obligadas a investigar a todos los extranjeros que no hayan salido en los vuelos que tenían reservados tanto a la ida como a la vuelta. Una amplia información se puede encontrar en <http://www.epic.org/privacy/intl/passenger_data.html>.

Sección técnica «Redes y Servicios Telemáticos» (Josep Solé i Pareta y Luis Guijarro Coloma)

Tema: *Presentación de la sección técnica «Redes y Servicios Telemáticos»*

Esta nueva sección que se inaugura en este número pretende ser una sección de carácter tecnológico, cuyo ámbito es el de las redes de comunicaciones, por un lado, y el de los servicios telemáticos que se prestan sobre ellas, por otro lado. Ambos aspectos, esto es, tanto el de las redes, como el de los servicios, encuentran en la actualidad su paradigma de investigación, experimentación y aplicación en Internet. Así pues, esta sección tratará principalmente sobre Internet y su entorno, intentando no solaparse con la sección sobre Internet ya existente. Aquí Internet se abordará tanto desde los aspectos de red, que son los que posibilitan su carácter ubicuo, como desde los aspectos de las aplicaciones que, aprovechando esta ubicuidad, proporcionan servicios útiles a los internautas, sean estos ciudadanos, empresas o instituciones. Sin ánimo de ser exhaustivos, a continuación se incluye una lista de los temas que se pretenden cubrir en esta sección:

- Tecnologías de redes de acceso a Internet: xDSL, HFC, WiFi, FTTH, etc.
- Tecnologías de redes de transporte para el soporte de Internet pública: IP/ATM, IP/SDH, IP/DWDM, GMPLS, 1-10 GbEthernet, Resilience Packet Ring (IEEE 802.17), etc.
- Evolución de los nuevos protocolos de Internet: IPv6, MPLS, RSVP-TE, OSPF-TE, etc.
- Movilidad en Internet: localización, itinerancia, traspaso, etc.
- Calidad de servicio en Internet y soporte de aplicaciones multimedia
- Aplicaciones distribuidas sobre Internet: comercio electrónico, tele-educación, administración electrónica, etc.
- Experiencias en España: I+D, colaboración universitaria, etc.
- Reglamentación en España y en la Unión Europea: nombres de dominio, ISP, etc.

Tema: *Página web* <<http://www.nagios.org>>

(Esta referencia nos ha sido proporcionada por Carles Kishimoto y Abel Navarro, CCABA-UPC). Cuando nuestra red empieza a crecer, el control del estado de todas nuestras máquinas se complica, es entonces cuando se hace necesario el uso de aplicaciones o programas de *script* capaces de monitorizar todos los servicios que ofrecen los nodos de la red y de enviar una notificación que permita solucionar posibles problemas lo más rápidamente posible. Esta

referencia concierne a un programa de libre distribución, bajo licencia GNU-GPL2, que, entre otras muchas cosas, proporciona esta funcionalidad: se trata del programa Nagios.

El sistema de monitorización de Nagios corre en plataformas UNIX y consulta la disponibilidad de los servicios mediante plugins instalados en los sistemas a vigilar. Los datos que va recogiendo, como por ejemplo capacidad de disco, tiempo de respuesta o caída de una máquina o servicio, se almacenan en ficheros de texto o bases de datos para la creación de informes históricos. Permite agrupar las máquinas y servicios para poder notificar por correo electrónico selectivamente a los responsables sobre el estado de los equipos. Todos los datos que recoge se pueden consultar fácilmente mediante su interfaz web y además se crea dinámicamente mapas en 2D y 3D de la topología de la red.

Este sistema se está usando con éxito desde hace más de un año en la red experimental i2CAT (Internet 2 en Catalunya).

Tema: *Libros*

Luis Guijarro Coloma, *Redes ATM. Principios de interconexión y su aplicación*, Madrid: Ra-Ma, 2000. ISBN 84-7897-395-8

La Internet basa su funcionamiento en la existencia de diversas tecnologías que han ido apareciendo desde principios de los años ochenta. Ahora bien el componente fundamental que permite un funcionamiento consistente y armonioso de toda la amalgama existente de redes es el protocolo IP. Dentro de la aparición gradual de nuevas tecnologías de red, a principios de los años noventa vio la luz una tecnología muy prometedora. Se trató de las redes ATM (*Asynchronous Transfer Mode*, Modo de Transferencia Asíncrono), que permitieron velocidades de transmisión de 34 Mbit/s y de 155 Mbit/s y el transporte datos, voz y vídeo.

Este libro describe las bases técnicas de la Internet y de las redes ATM y constata el impacto que la aparición de las redes ATM ha tenido sobre el aumento de la capacidad de transmisión y del número de usuarios de la Internet. Aborda asimismo los fundamentos de la inclusión de redes ATM en la Internet. Todo ello asumiendo en el lector únicamente conocimientos generales sobre redes de área local.

S. Keshav, *An engineering approach to computer networking: ATM networks, the Internet, and the telephone network*, Reading: Addison-Wesley, 1997 (Addison-Wesley professional computing series) ISBN 0201634422. Este libro es una introducción a las redes telemáticas desde una perspectiva ingenieril que no sólo describe cómo funcionan las redes, sino también se adentra en los principios de su diseño, a diferencia de otros libros clásicos en el mundo de la Telemática, como 'el Tanenbaum' o 'el Stallings'. El libro aborda en paralelo tres de las principales tecnologías de red existentes: las redes ATM, la Internet y la red telefónica y, además, cubre los aspectos principales de las redes, tales como el encaminamiento, el control de errores, el control de flujo, la gestión de tráfico, la estructuración en capas, el acceso múltiple, la conmutación, el direccionamiento, los nombres y el *scheduling*.

Joan Serrat Fernandez, Alex Galis (Editors), *Deploying and Managing IP over WDM Networks*, Artech House Publishers, May 2003, ISBN: 1-58053-501-1. Este libro estudia la evolución de la arquitectura de redes de datos hacia IP sobre WDM. Aunque se prevé que en futuro las tecnologías IP y WDM serán las tecnologías de red dominantes, su introducción se hará de forma gradual complementando y reemplazando las actuales soluciones de red basadas en las tecnologías ATM y SDH. Este libro estudia esta evolución, dando una extensa visión general del problema y exponiendo algunos resultados de investigación para la solución de IP directamente sobre WDM. En particular, el libro describe los requerimientos necesarios tanto nivel de arquitectura como de gestión de red para la integración entre la capa IP y la capa WDM. Los conceptos de interoperabilidad y de gestión se han demostrado en un campo de prueba constituido por encaminadores IP y anillos WDM dentro del proyecto de investigación financiado por la Unión Europea WINMAN. El libro concluye con un análisis sobre la evolución de la gestión de redes y el despliegue de redes ópticas.

Sección técnica «Seguridad» (Javier Areitio Bertolín, Javier López Muñoz)

Tema: *Libros*

C. Andrews, D. Litchfield, *SQL Server Security*. McGraw-Hill Osborne Media. 2003.

M. Arnold, S.D. Wolthusen and M. Schmucker, *Techniques and Applications of Digital Watermarking and Content Protection*. Artech House. 2003.

C. Cobb, *Cryptography for Dummies*. Hungry Minds, Inc. (ISBN 0764541889). 2003.

D. Cougias, K. Koop and E.L. Heiberger, *Backup Book, The: Disaster Recovery from Desktop to Data Center*. Schaser-Vartan Books. Third Edition. ISBN 0972903909. 2003.

J. Edney, W.A. Arbaugh, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley Professional. 2003.

N. Galbreath, *Cryptography for Internet and Database Applications: Developing Secret and Public Key Techniques with Java*. John Wiley & Sons Ltd. 2003.

M.G. Graff, K.R. Van Wyk and D. Russell, *Secure Coding: Principles and Practices*. O'Reilly & Associates. 2003.

P.B. Janeczko, *Keeping Secrets: A Field Guide to Codes, Ciphers and Secret Writing*. Candlewick Press. 2004.

T. Lammle, E. Quinn, J. Menga and T. Lancaster, *CCSP*. Sybex. ISBN 0782142338. 2003.

R. Latham, *Bombs and Bandwidth: The Emerging Relationship Between IT and Security*. New Press (ISBN 1565848624). 2003.

W. Mao, *Modern Cryptography: Theory and Practice*. Prentice-Hall. PTR. 2003.

J. Nathan, D. Ruiui, and J. Haile, *Snort 2.0: The Complete Guide to Intrusion Detection*. John Wiley & Sons Ltd. 2003.

T.R. Peltier, *Information Security Risk Analysis*. Auerbach Publications. 2001.

J. Pieprzyk, T. Hardjono and J. Seberry, *Fundamentals of Computer Security*. Springer Verlag. Berlin. 2003.

A.D. Ross, K. Feltman, *Network Security Essentials*. John Wiley & Sons Ltd. 2003.

Sección técnica «Tecnología de Objetos» (Esperanza Marcos, Gustavo Rossi)

Tema: *Análisis y Diseño Orientado a Objetos*

Craig Larman, *UML y Patrones: Una introducción al análisis y diseño orientado a objetos*, Segunda Edición, Prentice-Hall, 2002. Desde noviembre pasado está disponible la traducción al español de la segunda edición del libro de Craig Larman de introducción al análisis y diseño orientado a objetos (A/DOO), un excelente texto que ha tenido muy buena acogida tanto en las universidades como en el mundo profesional. Esta nueva edición mantiene el enfoque de la primera pero introduce mejoras significativas al refinar y extender el tratamiento de algunos temas como son: casos de uso, requisitos no funcionales, patrones de diseño GoF y arquitectura del software; además, ahora, se enmarca cada modelo descrito en el contexto del Proceso Unificado (UP).

Este libro es de gran utilidad para todas aquellas personas que conociendo los principios de la programación orientada a objetos (OO) y con experiencia en un lenguaje de programación OO, desean adquirir una visión práctica sobre un proceso software OO basado en UML. Resulta muy apropiado como libro de texto para una asignatura de A/DOO. A través de un proceso simple se describe de forma muy clara cómo llegar desde los requisitos a la implementación, planteándose de un modo realista la construcción de software OO: las técnicas presentadas son perfectamente aplicables en proyectos reales. Además, el proceso es compatible con UP y posee las propiedades que deben caracterizar a todo proceso basado en UML: dirigido por casos de uso, iterativo, incremental y centrado en la arquitectura.

Sin duda, se trata del mejor texto publicado para enseñar los principios del A/DOO, ya que combina un enfoque práctico (todos los modelos y técnicas se aplican sobre un único caso de estudio) con un tratamiento muy riguroso de las cuestiones que van surgiendo en las diferentes etapas del proceso, además de estar escrito de un modo muy didáctico. El proceso involucra las etapas de Modelado de Requisitos (Modelo de Casos de Uso y Requisitos no Funcionales), Modelado del Dominio, Modelado del Diseño (Colaboraciones de Objetos) y Modelo de Implementación. Los modelos se describen con detalle y se aplican al caso de estudio de un sistema de terminal de punto de venta, con lo que se consigue una gran claridad en las explicaciones.

La notación UML se va presentando gradualmente conforme se van explicando los diferentes modelos. En el capítulo 6 se presenta un caso de uso escrito de forma detallada que sirve como base para discutir los diferentes modelos. Se explica con claridad y minuciosidad cómo obtener las colaboraciones para ese caso de uso utilizando los patrones básicos de reparto de responsabilidades (GRASP) y los patrones de diseño GoF y se muestra cómo obtener la

implementación en Java de esas colaboraciones. El modelado se organiza en tres iteraciones en las que se extienden los requisitos iniciales y se refinan los modelos construidos. También hay que destacar la presentación y aplicación de los patrones de diseño GoF más utilizados, principalmente en el capítulo 23. El capítulo 30 es una buena introducción al tema de la arquitectura del software y el capítulo 32 trata de uno de los aspectos clave en el desarrollo de software: la influencia de los requisitos no funcionales en el diseño de la arquitectura del sistema, lo que se ilustra a través del caso de estudio considerado.

En definitiva un excelente libro sobre construcción de software orientado a objetos. Su lectura es un placer. (En <<http://www.craiglarman.com/>> se puede encontrar más información sobre el autor y el libro).

Sección Técnica «TIC para la Sanidad» (Valentín Masero Vargas)

Tema: *Congresos*

- SCAR 2003 20th Symposium on Computer Applications in Radiology. 7-10 Junio, 2003. Hynes Convention Center. Boston, Massachusetts. EE.UU. <<http://www.scarnet.org/>>.
- The Emerging Technologies & Healthcare Innovations Congress. Septiembre 22-24, 2003. Washington, DC. EE. UU. <<http://www.tethic.com/>>.
- Medical Image Computing and Computer Assisted Intervention. 1-4 Noviembre, 2003. Toronto, Ontario, Canada. <<http://www.miccai2003.org/>>.
- IV Congreso Nacional de Informática y Farmacia. Infarma' 2003. Noviembre 2003. Logroño. España.
- TEPR 2004. 17-21 Mayo, 2004. Fort Lauderdale, Florida. <<http://www.medrecinst.com/conferencias/tepr/index.asp>>.
- SCAR 2004 Annual Meeting. 21th Symposium on Computer Applications in Radiology. 20-23 Mayo, 2004. Vancouver Convention and Exhibition Centre. Vancouver, British Columbia. Canada. <<http://www.scarnet.org/>>.

Tema: *Cursos*

- 14th Annual Computed Body Tomography for the Technologist 2003. Flamingo. Las Vegas, Nevada Octubre 23-26, 2003.
- Introduction to Medical Informatics. Curso On-line sobre Informática Médica gratuito: (<http://www.cpmc.columbia.edu/edu/textbook/index.html>)

Sección técnica: «TIC y Turismo» (Andrés Aguayo Maldonado, Antonio Guevara Plaza)

Tema: *IFITT*

La *International Federation for Information Technology and Travel & Tourism* (IFITT), es una organización internacional independiente para la discusión, intercambio y desarrollo de conocimiento acerca del uso e impacto de las nuevas tecnologías de la información y las comunicaciones en la industria de los viajes y el turismo. Constituye la principal referencia en el campo de las tecnologías aplicadas al turismo.

Dicha organización posee miembros en la comunidad científica internacional. Desarrolla distintos proyectos en colaboración con la Organización Mundial del Turismo (OMT) y es responsable de la organización del principal congreso internacional en la materia (ENTER) y de la publicación de la revista «Information Technology and Tourism» (ITT).

El congreso ENTER se celebra anualmente, las conclusiones del congreso son publicadas en la editorial Springer Computer Science. La última edición se celebró en la ciudad de Helsinki en el mes de enero: Andrew J. Frew, Martin Hitz, Peter O'Connor (ed.), Springer Verlag, 2003, ISBN: 3-211-83910-0.

La revista ITT, es la primera revista científica que trata el tema de las relaciones entre las tecnologías de la información y el turismo. Sus líneas de interés son entre otras: sistemas de reservas, sistemas multimedia distribuidos, mercados electrónicos, etc.

Para mayor información acerca de la organización puede consultar la página <<http://www.ifitt.org>>.

Personal y transferible

Rafael Fernández Calvo
Socio senior de ATI

<rfcalvo@ati.es>

Ada Byron: una condesa analógica y digital

No sólo las máquinas de computación pueden ser analógicas y digitales; algunos psicólogos dicen que existen también habilidades personales digitales y analógicas. Entre las primeras se hallarían la capacidad de observación y experimentación, así como la objetividad; entre las segundas, la imaginación, la capacidad de visualización de conceptos y la aptitud para la metáfora.

«Si no me concedes poesía filosófica, déjame invertir el orden: ¿podrías concederme al menos filosofía poética, ciencia poética?»

Quien escribía esta frase en una carta a su madre era la persona que mejor podía combinar ambas facetas: Augusta Ada Byron, hija de un poeta de la talla de Lord Byron, nacido George Gordon, máximo representante del romanticismo tanto en sus obras como en su propia vida, y de una dama de la alta aristocracia inglesa, Anne Isabella Milbanke, conocida como Lady Byron tras su matrimonio con el poeta, del que se separó sólo unos meses después de la boda.

Parece evidente que Ada Byron, nacida en 1815 y conocida también como Lady Lovelace tras su matrimonio con el conde de Lovelace, recibió las gracias analógicas de su padre, al que, dada la vida viajera y aventurera de éste, no llegó a conocer pese a que ya tenía nueve años cuando murió de la peste en Grecia, luchando como voluntario contra los turcos. Moribundo, Lord Byron se lamenta amargamente de ello a su asistente: *«¡Oh, mi pobre y querida niña! ... mi querida Ada. ¡Dios mío, ojalá hubiese podido verla!»*.

Las gracias digitales le llegaban directamente de su madre, Lady Byron, mujer de carácter profundamente calculador y dominante, y muy aficionada a las ciencias en general y a la matemáticas en particular («princesa de los paralelogramos») la llama Lord Byron para intentar conquistarla). Durante toda su vida trató de evitar, inútilmente, que su hija Ada cultivara su sensibilidad poética, «para impedir» --decía-- «que siga el mal camino de su padre», al que odiaba profundamente y acusaba incluso de incesto con su hermana.

Locos por los ordenadores (II): Ada Byron y Charles Babbage, o la bella y la bestia

© 2000, Grupo Santillana de Ediciones, S.A. Ediciones El País, S.A. Segunda parte del capítulo titulado «Locos por los ordenadores del libro» «Esto es imposible (científicos visionarios a quien nadie creyó pero que cambiaron el mundo)», de Editorial Aguilar (pp 167-190). Javier Gregori, Coordinador. Madrid, julio de 2000, ISBN 84-03-09206-7. En próximos números de *Novática* aparecerá el resto de dicho capítulo. Se publica con los correspondientes permisos.

En aquellas primeras décadas del siglo XIX, la máquina de vapor moderna, inventada a mediados del XVIII por James Watt, estaba desarrollando todas sus potencialidades y la revolución industrial vivía una etapa de apogeo, cambiando modos de vida y trastocando el orden social y económico. Las plácidas campiñas británicas se despo-laban en beneficio de las grandes ciudades industriales, en las que crecían suburbios insalubres donde se apiñaban en condiciones miserables tan vívidamente descritas por Federico Engels, compañero y mecenas de Carlos Marx, los miembros de la nueva clase obrera. (Lord Byron estaba entre los que se oponían a estos cambios no tanto por razones sociales sino porque estaban destruyendo la Inglaterra tradicional; no así un tal Charles Babbage, del que hablaremos en las páginas que siguen, el cual tenía una clara comprensión de los nuevos fenómenos y los abordaba de forma innovadora).

Nacía una nueva religión de carácter positivista, el Progreso, cuyo nuevo Dios era la ciencia, a la que, como ocurre también en nuestro días, se contemplaba como capaz de resolver todos los problemas de la humanidad. Los avances científicos y técnicos eran en verdad impresionantes en todos los campos y entre ellos se encontraba uno aparentemente muy modesto, la invención de la ficha perforada, que iba a tener sin embargo consecuencias revolucionarias no sólo en el campo de la industria textil, para la que fue creada en 1801 por el francés Joseph Marie Jacquard, sino en el campo del cálculo automático y de lo que más de un siglo y medio después se llamaría informática o computación.

Ada y Babbage, o la bella y la bestia

--¿El señor Babbage, supongo?

--¿Lady Ada Byron, verdad?

Ada Byron, la bella sensible y a la vez analítica de nuestra historia, se encuentra por primera vez con la bestia, encarnada en el sabio Charles Babbage, durante una fiesta en el año 1833, cuando ella no ha cumplido aún dieciocho años y él es ya un viudo de cuarenta y dos, reverenciado como uno de los grandes genios de su tiempo y con acceso por dicha razón a los más selectos círculos intelectuales y aristocráticos. Es además una persona de múltiples intereses y saberes, y de ideas avanzadas en los campos social,

económico y político. Pero a la vez tiene fama, bien fundada, de hombre áspero en el trato, aunque también, de forma paradójica, amante de la vida social y muy dado a organizar reuniones y fiestas en su casa; es testarudo e incapaz no sólo de ceder sino ni siquiera de llegar a soluciones de compromiso («genio irascible» le ha llamado una biógrafa). Ada se acerca a Babbage en búsqueda de un profesor de matemáticas.

--Me han dicho que está teniendo algunos problemas con la música últimamente --dice la joven sonriente.

A Ada le habían contado esa misma mañana una anécdota que describía de modo fiel la personalidad de Babbage: irritado porque los músicos callejeros turbaban el silencio de su gabinete de trabajo, consideró imprescindible denunciarlos ante las autoridades. La respuesta de los denunciados fue muy sutil: decenas de ellos se concentraron durante varias horas ante su casa cantando y haciendo sonar sus instrumentos sin cesar.

--En labios de otra persona habría considerado esa pregunta como una provocación intolerable, pero es usted

una joven demasiado inteligente y demasiado bella para darle una respuesta airada que me cierre para siempre las puertas de su amistad.

En ese primer encuentro Babbage explicó sus ideas a la joven Ada y entre ambos surgió una fuerte atracción, predominantemente intelectual en el caso de la muchacha pero además sentimental, aunque siempre platónica, en el caso del sabio. Desde 1822 éste, con una subvención del gobierno de Su Majestad, había venido desarrollando parte de su «máquina de diferencias» o «motor diferencial» (*differential engine*), una calculadora mecánica muy avanzada y compleja, de tipo analógico, destinada a una tarea específica, en este caso la resolución de tablas matemáticas, especialmente de navegación, por el método de las diferencias con una precisión de seis decimales, y estaba a punto de dar un salto cualitativo de gran alcance.

(Continuará)

CIBSI'03

Segundo Congreso Iberoamericano de Seguridad Informática

28/10/2003 - 31/10/2003

Escuela Superior de Cómputo, Instituto Politécnico Nacional, México DF

<<http://www.escom.ipn.mx/cibsi/>>

Los objetivos de este congreso son, entre otros: integrar proyectos, sumar esfuerzos, intercambiar ideas, evaluar avances, contar con puntos de reunión e intercambiar conocimiento entre especialistas, instituciones, empresas y público en general interesado en las diferentes áreas que involucran la Seguridad Informática.

ATI es entidad colaboradora de este evento

Coordinación editorial

Novedades en las Secciones Técnicas de *Novática*

Se han producido novedades en las Secciones Técnicas de *Novática*. Son las siguientes:

- «**Administración Pública Electrónica**»: se ha creado esta nueva sección, que coordinarán **Gumersindo García Arribas** <gumersindo.garcia@map.es> y **Francisco López Crespo** <flc@ati.es>, ambos empleados públicos en el Ministerio de Administraciones Públicas y que coordinaron la monografía del pasado núm. 162 sobre este mismo tema.
- «**Arquitecturas**»: se incorpora **Víctor Viñals Yúfera** <victor@unizar.es>, profesor en el depto. de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. Acompaña al actual coordinador, **Jordi Tubella** <jordit@ac.upc.es>, profesor en el DAC de la Universitat Politècnica de Catalunya.
- «**Calidad del Software**»: por motivos de carga de trabajo, que le impiden seguir al frente de esta sección, cesa a petición propia como coordinador de la misma **Juan Carlos Granja Álvarez** <jcgranja@goliat.ugr.es>, después de varios años de ejercer dicha función, habiendo coordinado varias monografías y publicado numerosos artículos. Le transmitimos nuestro más sincero agradecimiento por su valiosa labor. Por el momento esta sección queda sin titular.
- «**Interacción Persona-Computador**»: se incorpora **Jesús Lorés Vidal** <jesus@eup.udl.es>, profesor en el departamento de Informática de la Universitat de Lleida. Acompaña al actual coordinador, **Julio Abascal González** <julio@si.ehu.es>, profesor de la Facultad de Informática de la Universidad del País Vasco.

Noticias de *Upgrade*

- En la reunión que el pasado 26 de abril celebró en Dublín el consejo de CEPIS (*Council of European Professional Informatics Societies*) se aprobó la reorganización de las funciones editoriales de la revista digital *Upgrade*, quedando éstas como sigue:
 - Director: **Rafael Fernández Calvo** (hasta ahora codirector de *Upgrade* y actualmente director de *Novática*). <rfcalvo@ati.es>.
 - Directores adjuntos: **François Louis Nicolet**, Suiza (hasta ahora codirector de *Upgrade* y antiguo director de *Informatik/Informatique*), <nicolet@acm.org>, y **Roberto Carniel**, Italia (fundador y director de la edición italiana de *Upgrade*), <carniel@dgt.uniud.it>.
- Recordamos la existencia de una lista cuyo objetivo es distribuir noticias sobre *Upgrade*. Información sobre sus características, y sobre cómo darse de alta en ella, en <<http://www.upgrade-cepis.org/pages/editinfo.html#newsletter>>.

Programación de *Novática*

Próximas monografías

La próxima monografía que publicarán *Novática* y *Upgrade* será la siguiente, salvo causas de fuerza mayor:

Julio-agosto (nº 164)

Monografía: «**Ingeniería del Software: estado de un arte Software – Software Engineering: State of an Art**».

Fecha de publicación: primera quincena de septiembre

Editor invitado: **Luis Fernández Sanz** (Universidad Europea de Madrid).

El resto de monografías de *Novática* y *Upgrade* para el año 2003 será anunciado en breve.

Apellidos Nombre
 Empresa/Organismo CIF/NIF
 Domicilio
 Ciudad Provincia
 Código Postal País
 Teléfono Fax Correo elec.

Nota: Rellenar los siguientes datos solamente si la dirección de envío es diferente de la anterior.

Domicilio para envíos
 Ciudad Provincia
 Código Postal País

Deseo suscribirme a Novática (6 numeros al año) en las siguientes condiciones (marcar con X la opción deseada y, en su caso, la cantidad de suscripciones solicitadas):

*** España**

- 1 suscripción: 54,10 € (9.000 pts.) (+4% IVA)
 _ suscripciones: 49,60 € (8.250 pts.) cada una (+4% IVA)

*** Otros países de la Unión Europea y Marruecos**

- 1 suscripción: 66,11 € (11.000pts.)
 _ suscripciones: 63,10 € (10.500 pts.) cada una

*** Resto del mundo**

- 1 suscripción: 95 dólares USA
 _ suscripciones: 90 dólares USA cada una

Abonaré el importe:

- Con domiciliación de cobro por entidad bancaria (deberá rellenar los datos bancarios abajo solicitados)
 Talón adjunto
 Transferencia bancaria a la cta. 3025-0004-30-1500001500, Caja de Ingenieros, Calle Buen Pastor 5, 08018, Barcelona, (España)

Fecha Firma

Estos datos se automatizarán en un fichero informático a los solos efectos de prestar a los subscriptores el servicio de suscripción a Novática. Los derechos de acceso, rectificación y cancelación previstos en la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal (LOPD) pueden ejercerse en la sede de Novática en Barcelona arriba citada.

DATOS BANCARIOS PARA DOMICILIACION

Banco/Caja.....

CÓDIGO CUENTA CLIENTE			
ENTIDAD	OFICINA	D.C.	NÚMERO DE CUENTA

nv 163

>-----

AUTORIZACIÓN DE COBRO

Le rogamos escriba otra vez los datos bancarios. ATI se encarga de su envío al Banco o Caja.

Banco/Caja.....

CÓDIGO CUENTA CLIENTE			
ENTIDAD	OFICINA	D.C.	NÚMERO DE CUENTA

Ruego a Uds. se sirvan tomar nota de que, hasta nueva orden mía en contra, deberán adeudar en mi cuenta arriba indicada los recibos que a nombre de D./Dª..... le sean presentados por la Asociación de Técnicos de Informática (ATI), en concepto de suscripción a la revista Novática.

Fecha..... Firma

Normas de publicación para autores

Febrero 2003

Novática agradece su contribución desinteresada a los miles de autores que han elegido y elegirán sus páginas para presentar sus aportaciones al avance profesional y tecnológico de la Informática.

Periodicidad: **Novática** tiene periodicidad bimestral y aparece los meses de febrero, abril, junio, septiembre, octubre y diciembre, salvo retrasos debidos a causas de fuerza mayor. El cierre de la edición es habitualmente un mes antes de la fecha de distribución (dos meses para los artículos del bloque monográfico).

Normas de revisión: todos los artículos serán sometidos a un proceso de «revisión por iguales» (*peer review*), o revisión por personas especializadas en la materia objeto del artículo, excepto los expresamente solicitados por **Novática** a sus autores. En el caso de las monografías, serán los editores invitados y su equipo los que realicen la revisión y decidan sobre su publicación o no. Excepto en el caso de las monografías, los artículos deberán ser enviados a la oficina de Coordinación Editorial (**Novática-ATI**, Calle Padilla 66, 3ª dcha., 28006 Madrid, <novatica@ati.es> (ver «Soportes» más abajo). Una vez aprobados por el revisor(es), serán publicados tan pronto como sea posible, si bien la publicación no está garantizada pues razones de exceso de material pueden hacerla imposible. Los autores serán informados del resultado de la revisión y de la publicación o no de los artículos remitidos.

Tamaño y formato de los artículos: los artículos deberán tener un máximo de 4.500 palabras, lo que equivale a entre 8 y 10 páginas DIN A4 a doble espacio (fuente Times New Roman, tamaño 12), incluyendo resumen, palabras clave, figuras, bibliografía y notas. Sólo en casos excepcionales se aceptarán artículos superiores a dicho tamaño. Salvo excepciones, los artículos no deberán incluir más de cinco ecuaciones ni más de doce referencias bibliográficas o notas, y deberán incorporar título, resumen (máximo 20 líneas), palabras clave (un máximo de 10), nombre y afiliación del autor/a (es/as), así como su dirección postal y electrónica, y números de teléfono y fax. **Nota importante:** título, resumen y palabras clave deberán enviarse en español e inglés.

Soportes: los artículos deberán ser enviados a **Novática** en formato digital, preferentemente mediante correo electrónico o, si no se tiene acceso a éste, mediante disquete a través de correo postal. En caso de envío por correo electrónico, si el fichero tiene un tamaño superior a los 250KB, es preciso enviar el fichero comprimido con ZIP e indicando qué procesador de texto entre los citados a continuación se ha utilizado. En ambos casos (correo electrónico o disquete) el artículo debe ser enviado en formato *Word*, *RTF* o *HTML*. En todos los casos el artículo habrá de enviarse también en formato *PDF* para asegurar la fidelidad al original en el proceso de edición.

En todos los casos es preciso además enviar las figuras por separado, con la mayor resolución posible (mínimo 600 ppi), teniendo en cuenta que solamente se publicarán en blanco y negro.

Lengua: aunque **Novática** admite artículos escritos en todas las lenguas reconocidas por la Constitución española y los Estatutos de las diferentes Comunidades Autónomas, dado que el ámbito de difusión de la revista conlleva su publicación en castellano, como lengua oficial común, los autores deberán presentar sus artículos en castellano y, si así lo desean, en otra lengua oficial de su elección. **Novática** enviará a los socios y suscriptores que lo soliciten una copia de la versión original de aquellos artículos que hayan sido escritos en una lengua oficial que no sea el castellano.

Copyright: **Novática** da por supuesto que un autor acepta las presentes normas al enviar su original y que, en caso de que esté destinado a ser publicado en otro medio ajeno a ATI (o ya haya sido publicado) debe de aportar la autorización del editor del mismo para su reproducción por **Novática** (incluida la autorización para realizar traducciones). **Novática** por tanto no asume ninguna responsabilidad sobre derechos de propiedad intelectual si un texto se ha publicado en otro medio de comunicación, sea inadvertidamente o no, por parte del autor. Todo autor que publique un artículo en **Novática** debe saber que autoriza su reproducción, citando la procedencia, salvo que el autor declare explícitamente que desea proteger sus derechos con © o *copyright*. Asimismo, se entiende que el autor acepta que, además de en **Novática**, su artículo podrá ser también publicado y distribuido electrónicamente, mediante los medios habituales de difusión de ATI (servidor WWW, listas de distribución Internet, etc.) en su totalidad o parcialmente.

Estilo: si bien **Novática** respeta totalmente el estilo y contenido de cada artículo, da por supuesta la autorización del autor para retocar su ortografía, léxico, sintaxis, titulación y paginación, a fin de facilitar su comprensión por el lector y de subsanar posibles errores. Cualquier cambio que afecte al contenido será consultado con el autor.

SOCIOS INSTITUCIONALES DE ATI

Según los Estatutos de ATI, pueden ser socios institucionales de nuestra asociación «las personas jurídicas, públicas y privadas, que lo soliciten a la Junta Directiva General y sean aceptados como tales por la misma».

Mediante esta figura, todos los profesionales y directivos informáticos de los socios institucionales pueden gozar de los beneficios de participar en las actividades de ATI, en especial congresos, jornadas, cursos, conferencias, charlas, etc. Asimismo los socios institucionales pueden acceder en condiciones especiales a servicios ofrecidos por la asociación tales como Bolsa de Trabajo, cursos a medida, *mailings*, publicidad en **Novática**, servicio ATInet, etc.

Para más información dirigirse a <info@ati.es> o a cualquiera de las sedes de ATI. En la actualidad son socios institucionales de ATI las siguientes empresas y entidades:

ABADÍA, SISTEMAS DE INFORMACIÓN
 AGROSEGURO
 AIGÜES DEL TER LLOBREGAT
 AIS - Aplicaciones de Inteligencia Artificial
 AJUNTAMENT DE L'HOSPITALET DE LLOBREGAT
 AYUNTAMIENTO DE TERRASSA
 ALMIRALL PRODESFARMA, S.A.
 ATOS ODS, S.A.
 BARCELONESA DE DROGAS DE PRODUCTOS QUÍMICOS
 Barcelonesa de Gestión Administrativa, S.L.
 BITWARE, S. L.
 BBR INGENIERIA DE SERVICIOS, S.L.
 BT TELECOMUNICACIONES, S.A.
 BURKE FORMACIÓN, S.A.
 CÁLCULO, S.A.
 CARGILL España, S.A.
 CCS PROFESIONALES, S.L.
 CENTRO ESTUDIOS VELÁZQUEZ, S.A. (C.E. ADAMS)
 CENTRO DE FORMACIÓN LA CAIXA/IBM
 CESISA
 CLASE 10 SISTEMAS, S.L.
 CLINICA PLATÓ FUNDACION PRIVADA
 COOPERS & LYBRAND AUDITORÍA Y CONSULTORÍA
 CONSULTORES SAYMA, S.A.
 Departament d'Ensenyament de la Generalitat de Catalunya
 DEPTO. INFORMÁTICA - ESC. POLITÉCNICA (CÁCERES)
 DOXA CONSULTORES, S.L.
 EDITORIAL BELLADONA S.L.
 EPISER, S.L.
 ESPECIALIDADES ELÉCTRICAS S.A. (ESPELSA)
 ESTEVE QUÍMICA, S.A.
 FINCONSUM - Financiación al Consumo
 FUNDACIÓN SAN VALERO
 GRUPO CORPORATIVO GFI INFORMÁTICA, S.A.
 GRUPO INFORMÁTICO ITEM, S.A.
 GS y C, Gabinete Sistemas y Consultoría, S.L.
 INFORMATION BUILDERS IBÉRICA, S.A.
 INQA TEST, S.L.
 INSERT SISTEMAS, S. A.
 INSTITUT D'ESTUDIS CATALANS
 INVERAMA, S.A.
 IN2 Ingeniería de la Información
 KRITER, S.A.
 LABORATORIOS SERONO, S.A.
 LEUTER, S.A.
 METASINCRO, S.L.
 OCCIDENTAL HOTELES MANAGEMENT, S.A.
 PAUTA FORMACIO S.L.
 RÁPIDA SISTEMAS INTEGRALES, S.A.
 RD SISTEMAS, S.A.
 RENAULT FINANCIACIÓN
 SADIÉL, S.A.
 SARA LEE DE ESPAÑA, S.A.
 SERES ESPAÑA, S.A.
 SERVEIS INFORMÀTICS
 SISTEMAS TÉCNICOS LOTERÍAS DEL ESTADO (STL)
 SOCIEDAD DE REDES ELECTRÓNICAS Y SERVICIOS, S.A.
 SOLUCIONES INFORMÁTICAS PARA EL COMERCIO, S.L.
 STRATESYS CONSULTING ADP&M, S.L.
 SYSDATA, S.L.
 T-SYSTEMS
 TATUM SISTEMAS
 TCP SISTEMAS DE INGENIERÍA, S.L.
 TRANSBAIX LLOBREGAT, S. A. (Grupo Secur)
 TRW ISCS, S.L.
 UMANIS
 UNIVERSIDAD DE EXTREMADURA (Dpto. de Informática)
 UNIVERSITAT OBERTA DE CATALUNYA
 WAPETON NUEVAS TECNOLOGÍAS, S.A.

Una Asociación abierta a todos los informáticos

Una Asociación útil a sus socios, útil a la Sociedad

¿Qué es ATI?

✓ ATI es una asociación *abierta a todos los técnicos y profesionales informáticos y que está implantada en todo el país a través de los Capítulos Territoriales existentes en diversas Comunidades Autónomas*. Creada en 1967, es en la actualidad la asociación más dinámica y más numerosa (5.000 socios a finales de 2000) de las existentes en el Sector Informático español, con sedes en Barcelona (sede general), Madrid, Sevilla, Silleda (Pontevedra), Valencia y Zaragoza.

✓ ATI es miembro de CEPIS (*Council for European Professional Informatic Societies*) y tiene un acuerdo de colaboración con ACM (*Association for Computing Machinery*). En el plano interno tiene establecidos acuerdos de colaboración o vinculación con Ada Spain, All y ASTIC.

¿Cuales son los objetivos de ATI?

Se resumen en uno esencialmente:

SER UTIL A SUS SOCIOS Y A LA SOCIEDAD

Más concretamente, ATI se propone:

- ✓ Defender, promover y mejorar el desarrollo de la actividad de quienes ejercen como profesionales y técnicos en el campo de las Tecnologías de la Información.
- ✓ Facilitar a sus socios el intercambio de experiencias, la formación y la información sobre dichas tecnologías.
- ✓ Contribuir a la promoción y desarrollo de las Tecnologías de la Información.
- ✓ Mantener relaciones con el entorno social y económico en que la Asociación se mueve.
- ✓ Fomentar la difusión de las Tecnologías de la Información y estudiar su impacto sobre la sociedad y sobre los ciudadanos.
- ✓ Colaborar con otras entidades profesionales informáticas implantadas tanto en nuestro país como fuera de él, especialmente en Europa y en la América Latina.

¿Cómo está organizada ATI?

- ✓ La *Asamblea General* de socios y la *Junta Directiva General* son los órganos máximos de dirección para el conjunto de la asociación.
- ✓ Los *Capítulos Territoriales*, con sus *Asambleas Territoriales* y sus *Juntas Directivas Territoriales*, estructuran la asociación en las Comunidades Autónomas mediante una organización de orientación federal.
- ✓ Las *Secciones Técnicas* y los *Grupos de Trabajo* sobre diversos temas facilitan la participación de los socios en las actividades de la Asociación.

¿Qué ofrece ATI?

Mediante el pago de una cuota anual, los socios de ATI pueden disfrutar de la siguiente gama de servicios:

- ✓ *Formación Permanente*
 - Cursos, Jornadas Técnicas, Mesas Redondas, Seminarios, Conferencias, Congresos
 - Secciones Técnicas y Grupos de Trabajo sobre diversos temas
 - Intercambios con Asociaciones Profesionales de todo el mundo

- ✓ *Servicios Profesionales*
 - Asesoramiento profesional y legal
 - Peritajes, diagnósticos y certificaciones
 - Bolsa de Trabajo
 - Emisión en España del ECDL (*European Computer Driving License*)
- ✓ *Servicios de Información*
 - Revista Novática (bimestral)
 - Boletín Informativo ATI Informa
 - Servidor Web
 - Red asociativa ATInet (acceso básico gratuito a Internet, correo electrónico, IntraATInet, listas de distribución generales y especializadas, dirección permanente)
 - Biblioteca
- ✓ *Actividades Sociales*
 - Promociones y ofertas comerciales
 - Intercambios internacionales

Juntas Directivas

Junta Directiva General

Presidente: Josep Molas i Bertrán.

Vicepresidente Primero: Fernando Piera Gómez.

Vicepresidente Segundo: Celestino Martín Alonso.

Secretario: Miquel Sàrries Griñó.

Interventor-Tesorero: Antoni Carbonell Nogueras.

Vocales: Asunción Yturbe Herranz, Julián Marcelo Cocho, Roberto Moya Quiles, César Pérez Chirinos.

Suplentes: Francisco López Crespo, Mario Piattini Velthuis.

Capítulos Territoriales (Presidentes)

Andalucía (Fernando Sanjuán de la Rocha); Aragón (Manuel Solans); Catalunya (Pere Lluís Barbarà Butifull); Galicia (José Gómez); Madrid (Fidel Tapia Castro); Valencia (Manuel Ortí)

¿Dónde está ATI?

Servidor Web: <http://www.ati.es>

Sede General y Capítulo de Catalunya

Via Laietana 41, 1º, 1ª, 08003 Barcelona
Tlf. 93 4125235; fax 93 4127713 / <secregen@ati.es>

Capítulo de Andalucía

Isaac Newton, s/n, Ed. Sadiel (Isla Cartuja), 41092 Sevilla
Tlf./fax 95 4460779 / <secreand@ati.es>

Capítulo de Aragón

Lagasca 9, 3-B, 50006 Zaragoza
Tlf./fax 976 235181 / <secreara@ati.es>

Capítulo de Galicia

Recinto Ferial s/n, 36540 Silleda (Pontevedra)
Tlf.986 581413; fax 986 580162 / <secregal@ati.es>

Capítulo de Madrid

Padilla 66, 3º, dcha., 28006 Madrid
Tlf. 91 4029391; fax. 91 3093685 / <secremdr@ati.es>

Capítulo de Valencia

Reino de Valencia 14, 46005 Valencia
Tlf./fax 96 3330392 / <secreval@ati.es>

Grupo Promotor Asturias-Cantabria <gp-astucant@ati.es>

Grupo Promotor Castilla-La Mancha <gp-clmancha@ati.es>

Revista Novática

Padilla 66, 3º, dcha., 28006 Madrid
Tlf.91 4029391; fax. 91 3093685 / <novatica@ati.es>