

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

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

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

Consejo Editorial

Joan Batlle Montserrat, Rafael Fernández Calvo, Luis Fernández Sanz, Javier López Muñoz, Alberto Lobel Ballori, Gabriel Martí Fuentes, Josep Moias i Bertran, José Onofre Montes Andrés, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramon Puigjaner Trepal, Miquel Sarries Griño, Adolfo Vázquez Rodríguez, Asunción Yturbe Herranz

Coordinación Editorial

Llorenç Pagés Casas <pages@ati.es>

Composición y autoodición

Jorge Llácer Gil de Ranales

Traducciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gt/lengua-informatica/>>

Administración

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

Secciones Técnicas - Coordinadores

Acceso y recuperación de la información

José María Gómez Hidalgo (Openet), <jmgomez@yahoo.es>

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

Administración Pública electrónica

Francisco López Crespo (MAE), <flc@ati.es>

Arquitecturas

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

Jordi Tubella Morgados (DAC-UPC), <jordit@ac.upc.edu>

Análisis STIC

Marina Touriño Troitino, <marinatourino@marinatourino.com>

Manuel Palao García-Suñto (ASIA), <manuel@palao.com>

Base de y tecnologías

Isabel Hernando Collazos (Fac. Derecho de Donostia, UPV), <isabel.hernando@ehu.es>

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

Escuela Universitaria de la Informática

Cristóbal Paraja Flores (DGSIC-UPV), <cparaja@si.upv.es>

J. Angel Velázquez Iruñe (DLSI, URJC), <angel.velazquez@urjc.es>

Entorno digital personal

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

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

Estándares Web

Encarna Quesada Ruiz (Alisys Software) <encarna.quesada@virat.com>

José Carlos del Arco Prieto (TCP Sistemas e Ingeniería), <jcarco@gmail.com>

Basión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young), <juan.baiget@ati.es>

Informática y Filosofía

José Angel Olivas Varela (Escuela Superior de Informática, UCLM) <joseangel.olivas@uclm.es>

Kenneth Gierah Martin (Kinnorad University), <kgherab@gmail.com>

Informática Jurídica

Miguel Chover Sellés (Universitat Jaume I de Castellón), <chover@lsi.uji.es>

Roberto Vivó Herrando (Eurographics, sección española), <rvivo@dstc.upv.es>

Legaloría del Software

Javier Dolado Cosin (DLSI-UPV), <dolado@si.upv.es>

Luis Fernández Sanz (Universidad de Alcalá), <luis.fernandez@uah.es>

Inteligencia Artificial

Vicente Botti Navarro, Vicente Julian Inglada (DSIC-UPV)

<vbotti_vinglada@dsic.upv.es>

Información Persona-Computador

Pedro M. Latore Andrés (Universidad de Zaragoza, AIPQ) <platore@unizar.es>

Francisco I. Gutierrez Vela (Universidad de Granada, AIPQ) <fgutier@ugr.es>

Lenguaje e Informática

M. del Carmen Ugarte García (BM), <cugarte@ati.es>

Lenguajes Informáticos

Oscar Geromte Ferrández (Univ. Jaime I de Castellón), <bellern@lsi.uji.es>

Inmaculada Coma Tatay (Univ. de Valencia), <inmaculada.coma@uv.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo), <xggo@uvigo.es>

Manuel Palomar (Univ. de Alicante), <mpalomar@dlsi.ua.es>

Mundo estudiantil y jóvenes profesionales

Federico G. Mon Trotti (RITSI) <gnu.fede@gmail.com>

Mikel Salazar Peña (Área de Jóvenes Profesionales, Junta de ATI Madrid), <mikelxbo_uni@yahoo.es>

Profesión Informática

Rafael Fernández Calvo (ATI), <rfdc@ati.es>

Miquel Sarries Griño (Ayto. de Barcelona), <msarries@ati.es>

Redes y servicios informáticos

José Luis Marzo Lázaro (Univ. de Girona), <joseluis.marzo@udg.es>

Juan Carlos López López (UCLM), <juncarlos@uclm.es>

Seguridad

Javier Arellano Bertolin (Univ. de Deusto), <jarellano@eside.deusto.es>

Javier López Muñoz (ETSI Informática-UMA), <jlm@icc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <galton@puente>

<galton@puente>

Software Libre

Jesus M. González Barahona (GSYC-URJC), <jgb@gsyc.es>

Israel Herráiz Tabernero (UAJ), <isra@herraz.org>

Tecnología de Objetos

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

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

Tecnologías para la Educación

Juan Manuel Doderio Beardo (UC3M), <doderio@inf.uc3m.es>

César Pablo Córcoles Brinco (UDC), <ccorcoles@uoc.edu>

Tecnologías y Empresa

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

Francisco Javier Cantais Sánchez (Indra Sistemas), <fjcantais@gmail.com>

Tendencias tecnológicas

Alonso Alvarez Garcia (TID), <aad@tid.es>

Gabriel Martí Fuentes (Interbits), <gabi@atinet.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)

<aguayo.guevara@icc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos.

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

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º, dcha., 28006 Madrid

Tfn. 914029391; fax. 913093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Av. del Reino de Valencia 23, 46005 Valencia

Tfn./fax 963330392 <secreval@ati.es>

Administración y Redacción ATI Cataluña

Via Laietania 46, ppal. 1º, 08003 Barcelona

Tfn. 934129235; fax. 934127713 <secregen@ati.es>

Redacción ATI Aragón

Lagasca 9, 3-B, 50006 Zaragoza.

Tfn./fax 976235181 <seccara@ati.es>

Redacción ATI Andalucía

<seccrand@ati.es>

Redacción ATI Galicia

<seccregal@ati.es>

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

Publicidad

Padilla 66, 3º, dcha., 28006 Madrid

Tfn. 914029391; fax. 913093685 <novatica@ati.es>

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

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

Portada: En la cresta - Goncha Ariza Pérez / © ATI

Diseño: Fernando Agresta / © ATI 2003

editorial

ATI protesta contra la marginación de las titulaciones informáticas > 02

documentos

Carta de la Junta Directiva General de ATI al Secretario de Estado de la SETSI > 02

en resumen

Casual Summer. Presentación del número de verano de 2010 > 03

Llorenç Pagés Casas

Informática en presente y futuro

Entrevista

Competitividad y desarrollo tecnológico en Europa: los próximos 10 años > 04

Entrevista a *Neelie Kroes*

Recursos humanos

Tecnologías de la información: ¿una profesión para hombres? > 08

Miren Idoia Alarcón Rodríguez, Luis Fernández Sanz

Software

Sistemas informáticos multilingües con lenguas españolas > 12

Ignacio Agulló Sousa

experiencias profesionales

Desarrollo de software

Una experiencia en transferencia de tecnología de Desarrollo de Software > 16

Dirigido por Modelos

Jesús J. García Molina, Jesús Sánchez Cuadrado, Joaquín Lasheras Velasco

Herramientas gráficas

Mapas tecnológicos > 21

Alonso Álvarez García

Inteligencia Artificial

Experiencias en el IIIA en investigación fundamental en Inteligencia Artificial > 25

Ramon López de Mántaras Badía

Experiencias en el IIIA en desarrollo de proyectos de Inteligencia Artificial > 31

Ramon López de Mántaras Badía

Protección de datos personales

Medidas de seguridad para ficheros de datos de carácter personal: Reflexiones y experiencias – La norma vs. la aplicación práctica > 35

Marina Touriño Troitino

Simuladores

Desarrollo de simuladores marítimos para FOREM-A > 38

Inmaculada Coma Tatay, Sergio Casas Yrurzum, Ausiàs Llorenç Palau, Jose Vicente Riera López

Tecnologías para la movilidad

Tecnologías móviles y ubicuas en un entorno docente universitario > 43

Pilar Moreno Díaz, Israel Herráiz Tabernero, Jesús Sánchez Allende, Santiago Portela García-Miguel

Universidad y empresa

Experiencias de investigación en la universidad y la empresa > 48

José María Gómez Hidalgo

Referencias con firma > 53

lecturas

Historia de la Computación

El Test de Turing: historia y significado > 63

Giuseppe O. Longo

Impacto económico y social de la Informática

Leones supercomputerizados > 75

José Luis Cárpatos

asuntos interiores

Socios Institucionales > 77

Los coordinadores de las Secciones Técnicas de nuestra revista nos ofrecen en este número de verano nuevas visiones, acerca de una amplia variedad de temas relacionados con la Informática, refrendadas con su prestigiosa firma.

Apple y el gallego

La presencia de la lengua gallega en los menús de los ordenadores y otros dispositivos tecnológicos propios de nuestra sociedad de la información (MP3, GPS, TDT, Nintendo, Wii...) es, con toda certeza, limitada, insuficiente y poco visible, mas no es inexistente. Mientras que el gallego está prácticamente ausente de las consolas, de los reproductores MP3 y de los navegadores GPS, en otros casos la oferta de productos en nuestra lengua es bastante más amplia, en particular, en el mercado de la telefonía móvil, de los sintonizadores TDT, de los sistemas operativos y de las aplicaciones informáticas.

Hablando de este último ámbito, el del mundo de los ordenadores, conviene recordar que la traducción de software libre al gallego fue una de las prioridades de la Consellería de Innovación e Industria del anterior gobierno gallego, de carácter progresista, una prioridad canalizada a través de Mancomún, el Centro de Referencia y Servicios de Software Libre, creado en 2005, que asumió la coordinación de las traducciones al gallego de Firefox, Thunderbird y OpenOffice, y llevó a cabo la publicación de Galinux, una distribución en gallego de Linux basada en Ubuntu.

Sin embargo, aunque la presencia del gallego en el entorno de Linux tiene una cierta relevancia, sobre todo simbólica, la implantación real de este sistema operativo en la informática sigue siendo minoritaria. De acuerdo con los datos de la consultora Net Applications, sólo un 1% de los ordenadores funcionan con Linux, mientras que el 4,5% lo hace con Mac OS X de Apple, y más de un 90% se rinde a alguna de las tres últimas versiones del Windows de Microsoft (XP, Vista y 7). De ahí la enorme importancia de la traducción de Windows para la normalización lingüística de cualquier idioma.

La primera versión de Windows traducida al gallego fue la de Windows XP, que se presentó en junio de 2004 en un acto institucional con presencia de Manuel Fraga y de Rosa García, responsable de Microsoft Ibérica, y supuso también la traducción al gallego del Office 2003 de Microsoft. La presentación de la versión gallega del infausto Windows Vista y del Office 2007 tuvo lugar en abril de 2008, esta vez con presencia de Emilio Pérez Touriño. La iniciativa y el encargo de la traducción, en ambos dos casos, partió de Microsoft y fue llevada a cabo por la empresa compostelana Imaxin Software.

Ahora, la multinacional de Redmond, animada por las millonarias compras de software que cada día más le garantiza el actual gobierno conservador de la Xunta de Galicia, confirma la perdurabilidad de la versión gallega de Windows y Office con la reciente traducción de Windows 7 al gallego, ya disponible en la sección de descargas de su web. De modo anecdótico, o tal vez no, resulta interesante observar que la realización de las traducciones al gallego del Windows y su presentación pública siempre tiene lugar con un cierto retraso respecto de otras lenguas de la Península, como la catalana. Así, en el caso del Windows 7, la presentación del Windows 7 en catalán y en vasco ya tuvo lugar, respectivamente, en noviembre de 2009 y en junio de 2010, mientras que la versión gallega aún está pendiente de presentación en el momento de escribir este artículo (14 de julio de 2010).

Sea como fuere, la situación actual es que dos de los tres sistemas operativos de mayor implantación en los ordenadores, Linux y Microsoft, "falan galego" y, si no cambian mucho las cosas, su fidelidad lingüística a nuestro idioma está garantizada (si bien por razones muy distintas en cada caso). En este estado de la cuestión, resulta llamativo el desdén de Apple por el gallego, lengua que no

incorpora ni en el sistema operativo Mac OS X Snow Leopard para sus ordenadores personales, ni en el cada vez más presente iOS 4, el sistema operativo de sus reproductores iPod Touch, de sus móviles iPhone y de sus tablet iPad. El OS X ofrece versiones en inglés, japonés, francés, alemán, español, italiano, neerlandés, sueco, danés, noruego, finés, chino tradicional, chino simplificado, coreano, portugués, ruso y polaco. El iOS 4 aumenta estas 17 lenguas con otras 15: turco, ucraniano, húngaro, árabe, tailandés, checo, griego, hebreo, indonesio, malayo, rumano, eslovaco, croata, catalán y vietnamita.

Impresiona comprobar que la única lengua sin estado permitida por Apple en su flamante sistema operativo es la catalana. Aun así, hay que recordar que Apple ya había traducido al catalán algunas versiones de su sistema operativo en los años 1990 y que dejó de hacerlo en 1999, a partir del Mac OS 9, alegando las generosas subvenciones de la Generalitat de Catalunya a Microsoft para traducir el Windows.

Es probable que la inminente apertura de una Apple Store en Barcelona, las campañas populares llevadas a cabo en pro de la versión catalana y, sobre todo, las dimensiones y características propias del mercado catalán, hayan motivado a la empresa de la manzana a modificar su posicionamiento con respecto a la lengua catalana diez años más tarde. Acceder a los servicios y productos de las tecnologías de la información en la lengua propia es un derecho de toda comunidad lingüística. Esperemos que no haya que aguardar otros diez años para ver los productos de Apple en nuestro idioma.

Xavier Gómez Guinovart (Coordinador de la Sección Técnica "Lingüística Computacional")

Desarrollo basado en modelos de sistemas empotrados críticos

La importancia y difusión de los sistemas empotrados aumenta continuamente, a la vez que su complejidad. Estos sistemas se integran y a menudo controlan el comportamiento de productos como automóviles, aviones, teléfonos, dispositivos médicos, etc. En muchos casos se consideran sistemas críticos, pues su fallo puede causar importantes daños económicos o, incluso, pérdidas humanas. Por este motivo, es necesario que su comportamiento sea fiable y con tasas de error muy pequeñas.

La evolución de los sistemas empotrados críticos ha causado que algunas de las técnicas tradicionales no sean siempre útiles. En concreto, los sistemas actuales son más complejos, coexisten con subsistemas acríticos y se incluyen en dispositivos de electrónica de consumo. Una alternativa de gran interés y potencial para este tipo de sistemas es el desarrollo basado en modelos (*Model-Driven Development*, MDD).

MDD es una propuesta de desarrollo de aplicaciones en la que los modelos juegan el papel principal. En este contexto, un modelo representa los distintos elementos con que se describe el software. El uso de modelos pretende aumentar el nivel de abstracción con el que se describe el software.

Fundamentalmente, MDD distingue entre modelos independientes de la plataforma (PIM, *Platform Independent Models*) y dependientes de la plataforma de ejecución (PSM, *Platform Specific Models*). La plataforma de ejecución está compuesta por el hardware y los componentes software (sistema operativo, lenguaje de programación, etc.) que proporcionan servicios básicos para la ejecución de un sistema. MDD propone el uso de transformadores para transformaciones entre modelos en general. Un ejemplo es la generación de forma (semi) automática de los PSM, que pueden dar lugar finalmente al código ejecutable.

El uso de transformadores o generadores es una de las bases del enfoque MDD. Su función es analizar ciertos aspectos de los modelos y sintetizar varios tipos de notaciones de salida, que pueden ser código fuente, entradas para un simulador, descripciones XML, o representaciones alternativas del modelo, que pueden ser la entrada a herramientas de análisis. Los transformadores permiten generar código diferente, en función de los requisitos no funcionales de un determinado componente. Este enfoque proporciona varias ventajas: aumenta la productividad al trabajar con elementos de mayor nivel de abstracción, los transformadores pueden ser la base para generar PSM para distintas plataformas y la calidad del software aumenta, ya que si se prueba que los modelos PIM son correctos, el código final también lo será, si se emplean transformadores correctos.

Una tendencia actual consiste en enriquecer los modelos funcionales con información no funcional relativa a su comportamiento temporal, nivel de seguridad o requisitos de disponibilidad. Se pueden emplear herramientas de análisis para comprobar que un determinado modelo cumple las especificaciones no funcionales de un sistema. Igualmente, se pueden utilizar diversas transformaciones, dependiendo de los requisitos no funcionales de distintas partes de un modelo.

ASSERT es un proyecto del 6º Programa Marco de la Unión Europea, que tuvo por objetivo sentar las bases del desarrollo de sistemas de tiempo real en el dominio aerospacial a medio plazo. El consorcio estuvo formado por 28 socios representativos de la industria espacial europea, centros de investigación, empresas de software y fabricantes de herramientas, liderados por el centro de investigación de la Agencia Europea del Espacio (ESA/ESTEC). Entre ellos se encontraban el Instituto Europeo del Software (ESI), con sede en Bilbao, la empresa Software, de Vigo, y el grupo de sistemas de tiempo real e ingeniería de servicios telemáticos (STRAST) de la Universidad Politécnica de Madrid. El proyecto se llevó a cabo en los años 2004 a 2008.

Uno de los ejes principales de ASSERT es el empleo de un proceso de desarrollo basado en modelos para la producción de software de tiempo real embarcados. Una peculiaridad de estos sistemas es el uso de herramientas de ingeniería como Matlab/Simulink, SCADE, o SDL, para modelar el comportamiento funcional del sistema. El proceso de ASSERT permite utilizar estas herramientas, y también código funcional preexistente escrito en C/C++ o Ada, para modelar los aspectos funcionales del sistema, junto con herramientas basadas en UML. El modelo independiente de la plataforma comprende, además de la vista funcional, basada en estas notaciones, una vista de datos y una vista de interfaces. Esta última especifica la descomposición del sistema en subsistemas y las interfaces entre ellos. Los requisitos temporales y otros requisitos no funcionales se incorporan en esta vista como anotaciones formales en los puertos de interfaz. El lenguaje utilizado para ello es UML, aunque existe también una versión que emplea AADL. El modelo dependiente de la plataforma comprende una vista de despliegue, en la que se describe la plataforma mediante el lenguaje AADL, y una vista de concurrencia, en la que se modela el sistema como un conjunto de componentes primitivos de la plataforma subyacente: tareas periódicas y esporádicas, objetos de datos compartidos, objetos pasivos, y canales de comunicación.

La vista de concurrencia se genera automáticamente a partir del PIM y de la vista de despliegue, y el código final, en Ada 2005, se genera también de forma automática a partir del PSM.

Un aspecto fundamental del proceso de desarrollo de ASSERT es el empleo de un metamodelo común a todas las vistas del sistema, el modelo computacional de Ravenscar (*Ravenscar Computational Model*, RCM). Este modelo define un sistema de tiempo real como un conjunto estático de tareas concurrentes que se comunican mediante objetos de datos con acceso mutuamente exclusivo, plani-

ficadas con prioridades fijas y desalojo. Este modelo permite preservar las propiedades temporales desde el PIM hasta el código ejecutable, y verificarlas estáticamente utilizando técnicas de análisis de tiempo de respuesta. El código se ejecuta sobre una plataforma especializada, la máquina virtual de ASSERT, que asegura asimismo la preservación de las propiedades temporales durante la ejecución. En el proyecto se desarrolló también un conjunto de herramientas que soportan el proceso de desarrollo, y que están disponibles en la página web del proyecto.

En conjunto, los resultados de ASSERT han demostrado la aplicabilidad de MDD a los sistemas empotrados con requisitos de tiempo real críticos. El proyecto CHESSE, que forma parte de la iniciativa ARTEMISIA del 7º Programa Marco, pretende avanzar en esta línea, extendiéndola a otros dominios de aplicación, como el transporte ferroviario o las telecomunicaciones, e incorporando el perfil MARTE de UML como notación básica.

Alejandro Alonso Muñoz y Juan Antonio de la Puente Alfaro (Coordinadores de la Sección Técnica "*Sistemas de Tiempo Real*")

El software libre amenazado: código binario sin fuentes en el kernel Linux

En sus últimas conferencias, además de los temas habituales relacionados con conceptos fundamentales sobre el software libre, Richard M. Stallman ha mencionado en varias ocasiones que "Linux no es software libre". La última vez que asistimos a una de estas conferencias, y oímos tal afirmación, pensamos que estaba exagerando, y que no era más que una muestra de la tradicional rivalidad que mantiene con el autor principal de Linux, Linus Torvalds.

Días después de aquella conferencia, el diario Público publicaba una entrevista a Stallman con el titular "*Linux no es software libre*" [1]. En Barrapunto [2], los comentarios sobre la noticia criticaban al periódico y esgrimían las ya habituales quejas sobre la manipulación periodística de las palabras de Stallman.

Sin embargo, nosotros oímos esas palabras en la conferencia, así que el artículo del diario Público no contenía ningún sesgo. Intrigados por el tema, decidimos investigar un poco más sobre este asunto, y ver si Stallman estaba en lo correcto o si, por el contrario, estaba siendo excesivamente celoso al aplicar la definición de software libre a Linux.

¿Hay código no libre dentro de Linux?

En nuestra búsqueda para dar respuesta a esta pregunta, una de las primeras referencias que encontramos fue LinuxLibre, una versión de Linux difundida por la *Free Software Foundation* (FSF), que se autodenomina "el Linux libre" [3]. Además, la FSF mantiene una lista de distribuciones que incluirían una versión no libre del kernel Linux [4], y que, por tanto, no podrían ser consideradas como software libre.

Sorprendentemente, en dicha lista aparece la distribución Debian GNU/Linux, conocida por el rigor que aplica para rechazar software que no se adhiera estrictamente a los principios del software libre. Debian cuenta con un Contrato Social que incluye unas directrices para decidir si un software en particular es libre o no [5]. Así, cualquier programa que quiera ser incluido en Debian tiene que cumplir esas directrices que incluyen, entre otros requisitos, todos los necesarios para que un programa sea libre (las habituales cuatro libertades del software). De hecho, la definición de software libre de la *Open Source Initiative* (OSI) está basada en este mismo texto [6]. Entonces, ¿cómo es posible que la FSF diga que Debian no es software libre, si sólo incluye software que cumple con las directrices anteriores? ¿Está Debian distribuyendo software privativo de manera inadvertida?

La distribución LinuxLibre de la FSF incluye algunos *scripts* que se encargan de borrar las partes ofensivas del kernel de Linux [7]. El resultado es un kernel Linux totalmente libre, al menos si hacemos caso a la documentación de LinuxLibre. Para la versión 2.6.34 de Linux (la última disponible en el momento de escribir estas palabras), el *script* elimina, al menos, 32 archivos de la distribución original de Linux.

Si escogemos al azar uno cualquiera de estos ficheros, por ejemplo, `drivers/net/ixp2000/ixp2400_tx.ucode`, podemos examinar su contenido para intentar averiguar por qué LinuxLibre lo elimina. A continuación reproducimos algunas líneas del mencionado fichero de código fuente (el código completo, que incluye la lista de todos los ficheros eliminados, puede consultarse en [8]):

```
.insns = (u8[]) {
0xf0, 0x00, 0x00, 0x07, 0x00,
0xd8, 0x03, 0x00, 0x00, 0x11,
(continúan 70 líneas similares)
```

¿Qué función tiene ese fragmento de código? Parece el *firmware* de una tarjeta de red, aunque podría tratarse de cualquier otra cosa. En cualquier caso, lo importante es que forma parte del kernel Linux, y como ese ejemplo, muchos otros, detallados en el *script* de LinuxLibre.

A estos fragmentos de código binario, incrustados dentro de archivos de código fuente, se les suele denominar *blobs*. Al tratarse de código binario, el estudio de su funcionalidad, o su modificación, resultan imposibles. Es decir, no cumplen uno de los requisitos básicos del software libre. Además, estos fragmentos también suponen un riesgo para la seguridad del sistema, porque al ser binarios es imposible certificar que no tendrán ningún efecto colateral no deseado.

¿Cuál es la postura de Debian?

¿Cómo puede ser que se haya añadido código binario al kernel Linux?
¿Cómo es que Debian, la distribución libre por antonomasia, no se haya dado cuenta de este problema?

La respuesta es obviamente que sí conocen el problema. De hecho, ha sido objeto de una fuerte controversia dentro de Debian, hasta el punto de que la postura oficial al respecto tuvo que ser votada por parte de los miembros del proyecto [9].

De todas las opciones de la votación, la postura ganadora fue la de que los *blobs* satisfacen los requisitos de las directrices de Debian, y de la licencia GPL del kernel Linux (que consiste básicamente en las cuatro libertades del software de Stallman), a menos que se demuestre lo contrario. Es curioso que se añadiera la coletilla "a menos que se demuestre lo contrario", porque es muy sencillo "demostrar lo contrario": simplemente hay que intentar decir qué hace ese trozo de código, o modificar su funcionalidad. Si no se puede hacer alguna de estas dos cosas, entonces no es software libre.

Tras la votación, el secretario de Debian dimitió, acusado de manipular el recuento de votos. El hasta entonces secretario de Debian niega haber manipulado esa votación, y esgrime como motivos de su dimisión la pérdida de confianza de algunos de sus compañeros en el proyecto.

Entre las opciones que se votaron, se observan dos posturas enfrentadas: aceptar los *blobs* de manera que los usuarios de Debian tengan una distribución con mayor funcionalidad, o eliminar de la distribución todas las partes no libres.

El Contrato Social de Debian dice textualmente que "*prometen mantener el sistema con todos sus componentes completamente*

libres", usando las directrices como guía para dilucidar si un programa es libre. El Manifiesto de Debian [11] también dice "*[Debian GNU/Linux] debe hacerse por parte de organizaciones en situación de avanzar con éxito y abogar por el software libre sin la presión de los beneficios o los ingresos*".

Sin embargo, en la votación la opción ganadora fue la de echar la vista para otro lado con los *blobs*, y presionar para que Debian incluya todas las novedades y funcionalidades, incluso si es en forma de *blobs*, en vez de promover el avance del software libre.

Resumiendo, **Linux incluye fragmentos de código binario como parte de su código fuente**, y este código binario se incluye en la inmensa mayoría de distribuciones de GNU/Linux. **Debian, la distribución que nació para promover y hacer avanzar el software libre**, y que tanto ha ayudado en el pasado a tener una distribución completamente libre, está cambiando sus criterios y **acepta tener código binario que no cumple sus propias directrices de software libre**, con el fin de proporcionar mayor funcionalidad a sus usuarios.

¿Qué opinan los desarrolladores de Linux?

Algunos desarrolladores de Linux ya se han pronunciado sobre el tema. Por ejemplo, Theodore Ts'o se muestra a favor de aceptar dentro de Linux código binario sin sus correspondientes fuentes, a pesar de su licencia GPL. Theodore reaviva el antiguo debate sobre idealismo y pragmatismo, defendiendo que los *blobs* son necesarios para satisfacer la demanda de los usuarios, y que defender el software libre por el mismo concepto de software libre, supone anteponer las ideas a las personas.

La publicación de una entrada de contenido parecido al de este artículo en un blog de uno de sus autores [13] levantó bastante polvareda en algunos foros de la comunidad. Una muestra de esta polvareda es el informe de fallo abierto en la distribución Ubuntu [14] proporcionado por un usuario tras leer la mencionada entrada y que **avisa de que el paquete del kernel Linux de Ubuntu contiene violaciones de la licencia GPL**.

El informe contiene detalles sobre los archivos binarios distribuidos con Linux, y el propio Mark Shuttleworth comenta en ese informe que quizás esos archivos binarios deberían moverse al repositorio *restricted* de Ubuntu [15], que es donde se incluyen las partes no libres que se distribuyen junto con Ubuntu.

El software libre bajo amenaza

Es sorprendente que estemos viviendo de nuevo este debate de pragmatismo frente a idealismo. **Es precisamente este conflicto el que originó el contrato social de Debian**, y también el que está detrás de la idiosincrasia del software libre. Durante muchos años, la comunidad de software libre ha tenido que dedicar muchos esfuerzos para que la experiencia de los usuarios fuera la misma independientemente del tipo de software (privativo o libre) que eligieran. Actualmente contamos con distribuciones preparadas para ser usadas por cualquier persona que no haya tocado un ordenador nunca antes en su vida. ¿Qué hubiera sido del software libre si en aquel momento se hubiera enarbolado la bandera del pragmatismo para justificar la violación de los principios del software libre?

Las palabras de Stallman advirtiendo de los peligros del software privativo están más vigentes que nunca. **¿Cuánto tiempo pasará hasta que el primer binario malicioso forme parte de Linux y sea distribuido a millones de usuarios por parte de terceros como Debian?** Pero no se trata únicamente de un problema de seguridad informática, el riesgo es mucho mayor: al haber aceptado binarios dentro del *kernel*, se han propagado a

distribuciones como Debian, bastión clásico del movimiento por el software libre, provocando que no cumpla sus propios principios, su *constitución* y principal motivo de su existencia. En la película "Conspiración en la red", el argumento era que una malévola multinacional del software eliminaba a desarrolladores clave para acabar con el software libre. Parece que hemos encontrado una estrategia mejor.

- [1] <<http://www.publico.es/ciencias/206550/stallman/linux/software/libre>>.
- [2] <<http://softlibre.barrapunto.com/article.pl?sid=09/03/05/0818255>>.
- [3] <<http://www.fsfla.org/svnwiki/selibre/linux-libre>>.
- [4] <<http://www.gnu.org/philosophy/common-distros.html>>.
- [5] <http://www.debian.org/social_contract.es.html#guidelines>.
- [6] <<http://www.opensource.org/osd.html>>.
- [7] <<http://www.fsfla.org/svn/fsfla/software/linux-libre/scripts/deblob-2.6.34>>.
- [8] <http://git.kernel.org/?p=linux/kernel/git/stable/linux-2.6.34.y.git;a=blob;f=drivers/net/ixp2000/ixp2400_tx.ucode>
- [9] <http://www.debian.org/vote/2008/vote_003>.
- [10] <<http://www.mail-archive.com/debian-vote@lists.debian.org/msg22837.html>>.
- [11] <<http://www.debian.org/doc/manuals/project-history/ap-manifesto.es.html>>.
- [12] <<http://think.org/tytso/blog/2008/12/28/debian-philosophy-and-people/>>.
- [13] <<http://libresoft.es/Members/herraiz/blog/linux-is-not-free-software>>.
- [14] <<https://bugs.launchpad.net/ubuntu/+source/linux/+bug/575518>>.
- [15] <<https://bugs.launchpad.net/ubuntu/+source/linux/+bug/575518/comments/27>>.

Israel Herráiz Tabernero (Coordinador de la Sección Técnica "Sistemas de Tiempo Real") y **Antonio J. Reinoso Peinado** (Universidad Alfonso X el Sabio).

Interrupciones

Cuando una CPU necesita comunicarse con un componente, como por ejemplo el teclado o el modem, utiliza uno de estos 2 sistemas: por encuesta o por interrupción. Por encuesta la CPU, cuando está libre de trabajo pregunta a los componentes si tienen algo para ella, y los componentes le dicen que si hay teclas pulsadas o si hay datos que lleguen por la línea de teléfono. Cuando es por interrupción el periférico al tener algo que decir, manda una señal a la CPU, y ésta para lo que está haciendo para atender al teclado o al modem.

Antes de los móviles contactaba con mis amigos "vía encuesta". Iba a sus casas a picar al timbre para saber si estaban, o llamaba por teléfono fijo para saber si querían hacer algo. Si no querían hablar conmigo pedían sus padres o hermanos que dijeran que no estaban, yo les creía y nadie se enfadaba. Con la llegada de los móviles el mundo se ha vuelto un "sistema por interrupciones". Todo el mundo espera que pares lo que estás haciendo para contestar la llamada. No importa el día, la hora o el lugar. Siempre hay que estar disponible. Existen además dos daños colaterales: los tonos con los sonidos más molestos que se te puedan ocurrir; y lo idiota que parece la gente cuando van andando por la calle hablando solos con el manos libres.

Mi tío-abuelo Lucho hablaba solo y le llamaban loco. Ahora sería algo normal. La gente, al verlo liando sus cigarrillos con los restos de las colillas de los celtas sin boquillas, y hablando con sus amigos invisibles, no lo encontrarían extraño. Lo de las colillas puede que si fuera un poco raro. Añoro los tiempos en que nadie se molestaba si

no cogías el teléfono, cuando se quedaba a una hora y se llegaba puntual porque no se podía avisar en el mismo momento para decir que se llegaba tarde. Cuando nadie mandaba un SMS con un mensaje "llego 30 minutos tarde" cuando tú ya estás en el sitio 10 minutos antes esperando.

Todos nos sabíamos más de 20 números de memoria, y los horarios en los que se encontraba cada uno en su casa. Sabíamos la hora en la que llegaban y la hora en la que se iban a dormir. Sabíamos a qué hora cenaban y sabíamos a qué hora estaba vetado llamar porque todos escuchaban la conversación.

Llevo 2 meses sin móvil, sin necesidad de contestar ninguna llamada. Tengo una agenda con los teléfonos de todos mis amigos. Sé a qué hora pueden estar en casa. He aprendido de memoria, por fin, el teléfono de la casa de mi hermana. Y hace 3 años que tiene el mismo número.

Añoro los tiempos de la paciencia y de la puntualidad. Añoro los tiempos de la comunicación por encuesta.

Alejandro López Martín, por recomendación de Joan Baiget Solé (Coordinador de la Sección Técnica "Gestión del conocimiento")

Las humanidades en los nuevos planes de estudio de Informática

Como es sobradamente conocido, estamos inmersos en un proceso de profundos cambios en la orientación, estructura y contenidos de nuestros estudios universitarios, en el marco de un "espacio común europeo". Las sociedades están en permanente cambio, aunque no necesariamente se debería hablar de evolución o mejora, simplemente de cambio. La sociedad europea ha cambiado mucho, pero la española puede que haya cambiado incluso más debido a las evidentes mejoras en nuestra situación socio-económica de los últimos años. Esto ha provocado que los modelos sociales sean diferentes a los de las generaciones que estudiaban en las universidades en los 70, 80 e incluso 90 del siglo pasado. Los estudios universitarios de "humanidades" se han devaluado socialmente, el número de alumnos de estas titulaciones desciende vertiginosamente, y los profesores, departamentos y centros que las imparten cada vez tienen más dificultades y menos apoyo institucional para promover y llevar a cabo investigaciones en estos campos.

Por otra parte, los profesores que impartimos docencia en titulaciones de Ingeniería (tanto en 1er. y 2º ciclo como en posgrado y doctorado), en muchísimos casos observamos atónitos como nuestros alumnos (y muchos profesionales del ámbito) cometen infinidad de faltas de ortografía, son incapaces de redactar párrafos coherentes y su práctica más usual es el "cut and paste" de trabajos realizados por otros. Si se compara un trabajo de un alumno de humanidades con uno de ingeniería, observaremos en la mayoría de los casos la mayor calidad expresiva y de contenidos del primero. Pero la sociedad valorará mucho más (económicamente y en cuanto a prestigio social) al segundo. Esta paradoja es consecuencia de que se haya fomentado durante las décadas más recientes el saber superficial y monotemático, los medios de comunicación frívolos e ignorantes, ensalzando el éxito sin esfuerzo, disciplina ni rigor. Por otro lado, en la educación en los niveles anteriores a la universidad todo ha sido un cúmulo de despropósitos en los últimos años en el mismo sentido: ensalzando el éxito sin esfuerzo, disciplina ni rigor.

Y así estamos donde estamos. Nuestros universitarios cada vez escriben peor, razonan peor, son menos críticos y más ignorantes en cuanto a conocimientos fundamentales de la ciencia y de la tecnología. Por ejemplo, pocos estudiantes de ingeniería (de cualquier curso)

tienen una idea clara de cómo nació el cálculo diferencial e integral, la teoría de conjuntos o la estructura *if...then* en programación. ¿Es esto importante?, ¿Es más importante que conozca con profundidad el Java Developer X.X?. Creo que son cuestiones sobre las que deberíamos reflexionar. En algunos planes de estudios de ingeniería anteriores, había asignaturas como "historia de la ciencia" o dos cursos de "lógica" donde los alumnos podían aprender algo sobre Leibnitz, Cantor, Russell, Turing o los megáricos y estoicos, lo que entre otras cosas podría motivarlos a indagar más sobre los fundamentos del conocimiento que están aprendiendo y propiciaría probablemente que escribieran y pensarán mejor.

Por desgracia, en los nuevos planes de estudios la tendencia es la contraria: total desaparición de los contenidos humanísticos en los planes de ingeniería. Soy licenciado en Filosofía (Lógica), máster en Inteligencia Artificial y doctor Ingeniero en Informática. Soy profesor de Lógica en una escuela superior de Informática y desde mi experiencia (y la de algunos compañeros) creo que debería potenciarse el valor de los estudios de humanidades, fomentar y propiciar la vinculación entre los estudios de "letras" y los de "ciencias" y valorar más y premiar la disciplina y el rigor de los estudiantes en todas sus realizaciones.

José Ángel Olivas Varela (Coordinador de la Sección Técnica "Informática y Filosofía")

La privacidad de la información base para la construcción de la seguridad

La privacidad de la información es una propiedad de la seguridad y un elemento clave en su construcción. A nivel individual posibilita conseguir la libertad: contra la intrusión, la generación de perfiles y la manipulación. Permite proteger contra delitos como el robo de identidad, proporciona flexibilidad a la hora de acceder y utilizar contenidos y servicios, y posibilita controlar la información de uno mismo. Actualmente, la seguridad de la información y la privacidad pueden contemplarse como dos procesos no enfrentados sino complementarios, que permiten sumar esfuerzos en la ardua tarea de proteger contra la creciente escalada de riesgos de todo tipo. Se puede afirmar en relación a privacidad y seguridad que "la unión hace la fuerza". El estado de la ciencia y la tecnología de hoy en día posibilita abordar con rigor esta simbiosis que hace años parecía impensable. La privacidad y la seguridad pueden parecer, en principio, dos caras de una misma moneda, esencial en nuestra sociedad de la información y del conocimiento ya que poseen muchos mecanismos comunes como la criptografía y esteganografía, pero la privacidad va un poco por delante.

La seguridad de la información permite proteger los datos y recursos, tanto física como tecnológicamente, en confidencialidad (acceso limitado), integridad/autenticación (que sean auténticos, no modificados y completos), disponibilidad (accesibles a entidades legales), no repudio (impidiendo la negación de las acciones realizadas), autorización (control de acceso/gestión de identidad), auditoría/responsabilidad (*logging*, test de intrusión/*hacking* ético, revisión de código de aplicaciones con métodos formales, monitorización de actividades, etc.).

La privacidad de la información persigue proteger los datos en relación a cómo se definen y utilizan. Para ello se van usando cada día más tecnologías y políticas que definen y clasifican los datos y su uso, regulaciones y leyes (LOPD-RMS, HIPAA, GLBA, FERPA, FCRA, COPPA, etc.). La privacidad de la información se relaciona con la protección de datos y posibilita la libertad de elección, el control personal (mantener las esferas de soledad, reserva, control, autonomía, intimidad, etc.), la auto-determinación informacional (la privacidad implica el derecho a controlar la información personal y la

capacidad de determinar si la información de uno se recoge y se utiliza y de qué forma). Permite controlar la recogida, uso y revelación de cualquier información registrada acerca de una persona. Privacidad no es lo mismo que seguridad, sin embargo ésta es vital para la privacidad. Sin embargo, la privacidad integra un conjunto mucho más amplio de protecciones que sólo la seguridad.

La privacidad de la información es el derecho de individuos y organizaciones para denegar o restringir la recogida y utilización de información/conocimiento sobre ellos. Es difícil de mantener actualmente debido a que los datos se almacenan *online*. Nuestros datos personales son capturados no sólo a través del uso que hacemos de los computadores sino desde un número creciente de puntos de captura analógico-digitales.

La libertad se evaporaría sin la privacidad. La privacidad no sólo consiste en tener algo oculto sino que preserva esferas de reserva, autonomía, control personal, soledad, etc. Los tres aspectos clave de la privacidad son el verse libre de la intrusión (ser dejados solos), el control de la información sobre uno mismo y el verse libre de todo tipo de vigilancia (seguimiento, monitorización, persecución...). Algunos problemas de privacidad ocurren cuando la información personal apropiada para un contexto es compartida de forma inapropiada en otro (integridad contextual), el usuario debe poder juzgar el contexto. Es un imperativo actual el proporcionar un control mayor sobre la información personal compartida (es crítico saber qué información se trata y con quién es compartida).

Existe un número creciente de propiedades técnicas de la privacidad: anonimato/pseudo-anonimato (el proveedor del servicio puede observar el acceso al servicio pero no puede observar la identidad del usuario), no-correlación (entre emisor y receptor), no observabilidad (la presencia no es visible), transferencia trascordada (el proveedor del servicio puede identificar al usuario pero no puede observar detalles del acceso al servicio como a qué registros de datos accedió, qué palabras clave utilizó, qué datos descargó, etc.), denegación plausible, privacidad de la localización, resistencia al censo (uso de pseudónimos), *wizards* de privacidad (por ej. en redes sociales), gestión de *cookies*, bloqueo de anuncios, gestión de *malware*, filtrado Web/XML, etc.

Las principales preocupaciones en torno a la privacidad son:

1) La *recogida invisible de información*, es decir información personal sobre alguien sin el conocimiento de la persona (física o jurídica). La PII (*Personally Identifiable Information*) es cualquier información que mantiene una entidad que identifica o describe a un individuo. La *PII sensible* es el nombre asociado con: un DNI, datos financieros, datos médicos, biometría, permiso de conducir, pasaporte, tarjeta de compras, propiedades inmobiliarias, etc. Los *datos sensibles* son contraseñas, pronósticos económicos, información de propiedad intelectual, informes financieros, etc. La masa de información recogida de forma sigilosa es cada día mayor y abarca un número creciente de áreas: vigilancia (cámaras de TV/CCTV y webcams en parkings, comercios/supermercados, bancos, hospitales, aeropuertos utilizando rayos X y ondas milimétricas para ver las personas por debajo de su ropa, calles, carreteras, diversos edificios, etc.), *cookies* de trazabilidad al navegar por Web, tarjetas bancarias y de compras, identificación del que llama en telefonía, comunicaciones P2P, monitorización y trazabilidad basada en GPS/Galileo/Glonass, cajas negras de automóviles, Wi-Fi-WLAN, RFID/IoT, Bluetooth/Zigbee-WPAN, WiMAX-WMAN, teléfonos móviles, etc. Toda esta información da lugar a bases de datos de individuos, consumidores, etc. en áreas como registros de compras, actividad de navegación Web, formularios de cambio de dirección, listado de morosos, personas con hipotecas, personas con vivienda, personas con alguna multa de tráfico, etc. Los "medidores de energía inteligentes" permiten enviar la información del consumo de los contadores en tiempo real al provee-

de servicios lo cual le permite inferir información privada de las personas, como cuándo están en casa o no, rutinas de cuándo duermen, comen, ven la TV, usan la lavadora, y los cambios de rutina como la llegada de visitas.

2) *La utilización secundaria de información personal* para un propósito diferente para el que fue proporcionada. En este sentido se pueden identificar: a) Minería de datos. Consiste en buscar y analizar grandes masas de datos para encontrar patrones y desarrollar nueva información y/o conocimiento. El problema de la inferencia en bases de datos surge cuando la información confidencial puede ser obtenida de datos liberados por usuarios que no lo autorizaron. El objetivo de la PPDM (*Privacy-Preserving Data Mining*) consiste en desarrollar algoritmos que modifiquen los datos originales de forma que los datos privados y el conocimiento privado permanezcan privados incluso después del proceso de minería de datos. b) *Matching* por computador. Consiste en combinar y comparar información de diferentes bases de datos (por ej. usando DNIs, fotografías, números de la Seguridad Social para identificar registros). c) Generación de perfiles por computador. Consiste en analizar datos de ficheros para determinar las características de personas más probablemente implicadas en un cierto comportamiento. En el ámbito de los negocios la generación de perfiles permite determinar la propensión de un consumidor hacia un producto o servicio. En el ámbito policial permite crear descripciones de posibles delincuentes/terroristas. En el ámbito de la sanidad la tendencia a contraer enfermedades debido a la genética, el entorno, etc. El uso secundario de la información del consumidor es cada día mayor, observando un número creciente de áreas. En relación a qué fue antes el huevo o la gallina se puede afirmar que a partir de la privacidad de la información se puede construir una seguridad de la información más evolucionada.

La relación antagonista entre privacidad y seguridad que afirmaba que si se aumenta la seguridad disminuye la privacidad y recíprocamente, si se aumenta la privacidad se disminuye la seguridad, en la actualidad se ha superado ya que podemos concebir seguridad y privacidad como procesos auto-cooperantes, acumulativos y no opuestos. Aunque es evidente la necesidad de un compromiso entre privacidad y seguridad en el sentido de que la vigilancia mal dimensionada puede desencadenar en abuso.

La privacidad ayuda a la seguridad dando una vuelta de tuerca más. Por ejemplo la tecnología VPN (*Virtual Private Network*) no tiene nada de privada, protocolos de seguridad como SSH y SSL/TLS no proporcionan privacidad. La privacidad ayuda implantando mecanismos de anonimato para defenderse de ataques como el análisis de tráfico.

Dos paradigmas extremos de la privacidad son:

1) *Privacidad rigurosa*. El sujeto proporciona la mínima cantidad de datos posible y reduce en lo posible la necesidad de confiar en otras entidades. El sujeto es un usuario de seguridad activo. El objetivo de protección de datos es la minimización de los datos, protegerse contra la vigilancia, interrogación, agregación e identificación. Para ello se pueden utilizar por ejemplo herramientas PET. El adversario puede localizarse en el proveedor de comunicaciones o en la entidad que sostiene los datos.

2) *Privacidad basada en confianza*. Enfoque poco riguroso, el sujeto proporciona sus datos y existe una entidad controladora responsable de su protección. El sujeto de los datos ha perdido control de sus datos y en la práctica es muy difícil que pueda verificar cómo se recogen y procesan sus datos. El objetivo de protección de datos gira bajo los ejes del propósito, consentimiento y seguridad de datos. Se necesita confiar en los controladores de los datos en el sentido de su honestidad-competencia y se espera lo mejor de ellos.

Según Daniel Solove una sociedad sin protección de la privacidad sería sofocante y además injusta por violar un derecho. E incluso poco racional, con un bajo nivel de evolución y transgresora de una norma social clave. Según Diffie y Landau, la comunicación privada es fundamental para la seguridad nacional y la democracia. Un informe de la AAAS (*American Association for the Advancement of Science*) señala que la comunicación anónima *online* es una tecnología moralmente neutral. En EEUU la comunicación anónima se considera como un derecho constitucional (Segunda Enmienda) y por tanto un derecho humano sólido. Según la Unión Europea, la seguridad mal entendida puede erosionar los valores de la sociedad democrática como son la justicia, la libertad de elección de poder hacer cosas, la igualdad, y la libertad de ir cuando y donde uno desee. Para poder determinar un equilibrio entre privacidad perfecta y la total pérdida de privacidad, surgen dos conceptos: a) La noción de individuo. Qué se les debería permitir hacer a los individuos para que lleguen a estar totalmente realizados. b) La noción de sociedad. Para una elevada individualidad necesitamos una sociedad liberal y plural.

La "privacidad por diseño" está relacionada con determinar qué datos son necesarios para la provisión de un servicio bajo el principio de minimización de datos. Se han definido diferentes principios de la privacidad por diseño: a) Que sea proactiva, no reactiva y preventiva; b) Debería embeberse en el diseño; c) Debería existir por defecto; d) Deben existir visibilidad y transparencia; e) Siempre debe estar presente el respeto de la privacidad del usuario; f) Debe existir protección del ciclo de vida extremo a extremo; g) Debe incluir funcionalidad completa de modo que los resultados sean positivos, no neutros. La privacidad-seguridad por diseño es una meta pendiente en nuestra sociedad que debe potenciarse a todos los niveles (cultural-concienciación, a través de leyes, utilizando tecnología-gestión, etc.) ya que reduciría en gran proporción problemas de difícil solución actualmente existentes.

Tres posibles políticas de privacidad de cara al usuario son: a) *Opt-in*. Permite utilizar la información personal dentro de una organización pero requiere el consentimiento explícito del usuario antes de que la información personal pueda revelarse a terceras partes fuera de la organización.

La Directiva de Protección de Datos de la UE de 1995 enfatiza este modelo. b) *Opt-out*. Esta política da al usuario la opción de prohibir la compartición de su información personal no pública con terceras partes. Una organización que se propone compartir información no pública de un usuario con terceras partes debe darle la oportunidad para denegar el permiso de uso de su información personal. Se ha comprobado que un porcentaje inferior al 8% de usuarios se suele borrar de listas de e-mail. c) *Blanket Opt-in*. Está relacionada con el anonimato.

Una política de privacidad adecuada debería responder a preguntas tales como: ¿qué tipo de información se está recogiendo (direcciones de e-mail, direcciones postales, información del servidor/cookie, etc.)?; ¿existe más de un método de recogida de datos?; ¿qué otros métodos se van a emplear? Incluso se debe ser muy específico: ¿de qué forma se utilizará esa información recogida?; ¿con quién se compartirá esa información?; ¿dónde se almacenará la información recogida y qué nivel de seguridad se utilizará para su protección?; ¿cual será la política concreta si alguien desea des-inscribirse?

Además de especificar claramente lo que se debería hacer si un usuario no desea recibir ningún tipo de futuras comunicaciones y si un usuario no desea que su información personal se incluya en una base de datos.

Javier Areitio Bertolín (Coordinador de la Sección Técnica "Seguridad")

Minería de datos para la Ingeniería del Software

Estamos acostumbrados a escuchar los términos "data mining", "knowledge mining", "minería de datos" referidos principalmente a las aplicaciones que se realizan en los ámbitos de investigación médica o de entornos financieros. Sin embargo, desde hace ya mucho tiempo se ha trabajado con datos numéricos en el ámbito de la ingeniería del software, especialmente en los aspectos de gestión y planificación. También sabemos que la limitación en la disponibilidad de datos fiables es uno de los principales escollos para la utilización de datos cuantitativos. La realidad actual es que todavía seguimos sin utilizar datos cuantitativos de manera sistemática en muchos aspectos de gestión del software. Mencionamos a continuación algunas referencias importantes a tener en cuenta para quien desee profundizar en esta línea.

Herramientas para la minería de datos

■ **Weka** es la herramienta de minería de datos en código abierto más popular. Se compone de tres herramientas: a) Explorer para "probar" rápidamente diferentes algoritmos, preprocesado, selección de atributos y visualización; b) Experimenter para ejecutar múltiples algoritmos y análisis de bases de datos; y c) KnowledgeFlow para el diseño visual de experimentos.

■ **Rapidminer** es otra herramienta abierta de minería con un intuitivo y potente interfaz gráfico. Ha sido desarrollada en Java e incluye además Weka.

■ **Orange** es otra herramienta de minería de datos, pero en este caso está implementada en Python.

■ **R** es principalmente una herramienta estadística pero existen numerosos "plug-ins" de minería de datos incluyendo uno para Weka (RWeka). Además existe un entorno gráfico llamado Rattle que facilita la aplicación de ciertos algoritmos de minería.

■ Dentro del ámbito comercial nos encontramos con **Clementine**. También dentro del ámbito comercial podemos utilizar las librerías que se pueden encontrar en las herramientas clásicas como **Matlab**, **Mathematica** para algunos aspectos del análisis de datos.

■ **KNIME** es otra herramienta abierta de minería de datos basada en la plataforma Eclipse.

■ Además, existen diversas de herramientas para la extracción de datos de repositorios software como paso previo a la aplicación de minería de datos. Una relación de las mismas la podemos encontrar en <http://tools.libresoft.es/>.

Bases de datos disponibles

■ **PROMISE** (*Predictive Models in Software Engineering*, <http://promisedata.org/>). Este repositorio contiene múltiples bases de datos clasificadas según su propósito (estimación de defectos, estimación de costes, etc.) junto con los artículos que las han usado. Está relacionado con la conferencia que lleva el mismo nombre.

■ **ISBSG**: Es una base de datos clásica, de pago. La experiencia nos ha demostrado que es muy difícil aplicar algoritmos debido a la heterogeneidad de sus datos y a los diversos enfoques de "puntos de función" utilizados, lo que ha resultado en una práctica inutilidad en lo referente a inferencia sobre los datos <http://www.isbsg.org/>.

■ **FLOSS** (*Free/libre Open Source Software*) Metrics. <http://www.flossmetrics.org/>. El objetivo de este repositorio es disponer de una gran base de datos con información y métricas de cientos de proyectos software "Open Source".

■ **Ultimate Debian Database**. Esta base de datos recoge información sobre todos los aspectos de desarrollo de Debian, incluyendo los fuentes de Ubuntu, defectos, migraciones, etc. <http://udd.debian.org/>.

Congresos relacionados

La lista de eventos donde se pueden encontrar ponencias sobre este tema es muy extensa. Mencionamos a continuación algunos específicamente orientados a este tema:

■ **PROMISE**. *Predictive Models in Software Engineering*. Conferencia dedicada exclusivamente a los modelos de predicción y análisis de datos <http://promisedata.org/2010/>.

■ **MSR** (*Mining Software Repositories*). El mismo título de la conferencia indica que abarca la minería sobre cualquier tipo de datos relacionados con el software. <http://www.msrconf.org/>.

■ **SCAM** (*Source Code Analysis and Manipulation*). Aunque esta conferencia se centra en la manipulación de código, también incluye entre sus tópicos la minería de datos <http://www2010.ieee-scsm.org/>.

■ **ESEM** (*Empirical Software Engineering*). Se centra en los estudios empíricos y métricas del software (anteriormente se denominaba *Software Metrics*) pero también se presentan artículos de minería de datos aplicados a proyectos software <http://esem2010.case.unibz.it/>.

Revistas Especializadas

Existe un buen conjunto de revistas internacionales donde se publican resultados de minería de datos en Ingeniería de Software, como son *Empirical Software Engineering Journal*, *Journal of Systems and Software*, *Information and Software Technology*, *IEEE Trans. on Software Engineering*.

Daniel Rodríguez García (Universidad de Alcalá, futuro colaborador estable de *Novática*), **Luis Fernández Sanz y Javier Dolado Cosín** (Coordinadores de la Sección Técnica "Ingeniería del Software")

Presunción de inocencia

Hace unos 39 siglos se esculpió el "Código de Hammurabi" (conservado en el Louvre), que es considerado como la primera codificación conocida de derechos y obligaciones humanos. Como referencia, quizá más familiar, Moisés bajó del Sinaí, con las Tablas de la Ley, 5 ó 6 siglos después.

Hace unos 75 años, Isaac Asimov acuñó las "Tres Leyes de la Robótica" (luego evolucionadas a la Roboética) codificando derechos y obligaciones de robots, demonios y otro software de Inteligencia Artificial.

En 2002, Rodney Brooks, director del *MIT Artificial Intelligence Laboratory* pro-nosticaba que (igual que históricamente se han ido reconociendo ciertos derechos a muchos animales y, sobre todo a las mascotas), es plausible que surjan corrientes de reconocimiento de derechos a algunas de esas máquinas, sobre todo a las más antropomórfas y a las que "convivan" en nuestros hogares (robots domésticos).

Aunque los profesionales de los SI (sistemas de información) y las TIC (tecnologías de la información y las comunicaciones) estamos sujetos (por convicción y/o adhesión a un código ético profesional) a ciertos compromisos respecto a las SITIC, eso no obliga en general a otras personas. Y eso permite un pernicioso, generalizado y continuado linchamiento impune de los SITIC, e indirectamente de sus profesionales. Pernicioso, porque nos impone un sambenito a los profesionales; Pero sobre todo porque puede desembocar en un fallo sistémico. Me explicaré.

Pero antes, descendamos a lo concreto. Creo que las siguientes referencias son autoexplicativas y se aceptarán como meros botones de muestra de algo mucho más generalizado (las negritas en la cita han sido introducidas ad-hoc para una mejor exposición).

"*La juez sustituye la fianza de un millón de euros al alcalde de Seseña por otra de 10.000. Emite un auto de rectificación para aclarar un error informático debido a introducir más dígitos de los que procedían*"¹.

"La patronal fotovoltaica Asif pidió un análisis a una empresa, que concluyó que podía tratarse de que el contador considerara que las 12 de la mañana eran las 12 de la noche o **fallos informáticos**"².

"La lista de premiados (Premios Max de la SGAE entregados el 3 Mayo de 2010) publicada en la página web de la SGAE por un **error informático**"³.

"EEUU busca el ordenador fantasma. Los reguladores bursátiles descartan el factor humano como causa del pánico que sacudió a Wall Street" (Sandro Pozzi | Nueva York 07/05/2010)⁴.

"No podemos permitir que un **error tecnológico** espante a los mercados y provoque pánico", señaló el congresista demócrata Paul Kanjorski. El propio presidente, Barack Obama, pidió aclarar el desplome [bursátil] para "evitar que algo así vuelva a suceder"⁵.

"Un senador aprovechó ayer para reclamar que se ponga coto a las máquinas en las operaciones bursátiles. El demócrata Ted Kaufman denunció que se había puesto de manifiesto una vez más 'el potencial de los ordenadores gigantes de alta capacidad para alterar el mercado y crear el caos' en lo que denominó 'la batalla de los algoritmos'. Kaufman presentará una enmienda a la ley de reforma financiera para que se endurezca la regulación".

Como no parece probable que ni Hammurabi ni Moisés vayan, a estas alturas, a ocuparse de este tema, nos incumbe a los profesionales (sobre todo a través de nuestras asociaciones) intentar reencauzarlo. Para ello parecen obligadas ciertas reflexiones, actitudes y actuaciones.

En primer lugar, reconocer que todo error es humano.

Lucius Annaeus Seneca (cordobés, coetáneo de Jesucristo, tutor y consejero de Nerón) ya afirmó que "errare humanum est perseverare diabolicum" [errar es humano y perseverar diabólico]. Lo que fue parafraseado (y convertido en eslogan) por Alexander Pope (1688-1744) [más papista que el papa], en su famoso verso: "To err is human, to forgive divine" [errar es humano, perdonar divino]⁶.

El famoso HAL 9000⁷ no era tan perverso como parecía. Simplemente fue programado así ("heurístico-algorítmico" [¡signifique esto lo que signifique!]). Pero eso era ciencia ficción: intentemos mantenernos en la realidad.

En segundo lugar, aceptar que:

"... sin duda es un abuso del lenguaje... y si... [el]... error lo hubiesen tenido hace 20 años, sería un error 'mecanográfico' o hace 100 años, un error 'plumográfico'... la cuestión es tipificar el error sobre el instrumento y no sobre la persona que lo comete... pues al final, como todos los errores, es humano(gráfico)"⁸

En tercer lugar, aceptar que, pese a todo, hay / ha habido muchos [¡demasiados!] "errores informáticos / tecnológicos", esto es, errores humanos en materias informáticas / tecnológicas. En algunos de los cuales he/(¿quizá hemos?) incurrido por omisión o acción.

La tabla que presentamos en la columna siguiente⁹ extracta alguno de los más señalados, e indica su coste.

El excelente reciente artículo de Darren Dalcher en *Novática* 200 cita algunos importantes errores informáticos que efectivamente fueron tales. Como caso más relevante se dice que el fracaso en la introducción de un sistema informático en la Oficina de Recaudación del Reino

Año	Empresa	Coste (MUSD)	Proyecto
2004	Ford Motor Co.	400	Sistema de compras
2004	J. Sainsbury PLC	527	Sistema de SCM (Supply-Chain Management)
2004	Hewlett-Packard Co.	160	Problemas en el sistema de ERP
2002	Mc Donald's Corp.	170	Sistema de gestión de compras cancelado
2002	CIGNA Corp.	445	Problemas con sistema CRM
1999	Hershey Foods Co.	151	Problemas con el sistema de ERP
1996	Araïne Space	350	Explosión Ariane 5 por errores de especificación y diseño
1993	Bolsa de Londres	600	Sistema Taurus

Unido causó que no se enviasen recordatorios a los asalariados sobre que tenían que actualizar sus contribuciones al sistema nacional de seguros. Y que ahora ¡¡10 millones de personas sufren recortes en sus pensiones debido a eso!!¹⁰

Otra cita de Darren por el impacto de los errores en la fase de producción: "un sistema de envío de ambulancias que fue entregado a los usuarios (al tercer intento) y falló posteriormente en producción dando lugar a potenciales pérdidas de vidas humanas".

Dice Darren: "La práctica contemporánea de desarrollo de software se caracteriza sistemáticamente por proyectos descontrolados, entregas retrasadas, presupuestos excedidos, funcionalidad recortada, y una calidad cuestionable, lo que a menudo se traduce en cancelaciones, reducción del alcance, y un ciclo significativo de revisiones"¹¹.

Y continúa: "El resultado neto de ello es el típico despilfarro medido en términos financieros. Por ejemplo, en 1995, los proyectos fallidos en EEUU costaron... un total de 140.000 M USD... en 1996, 100.000 M USD... en 1998 75.000 M USD".

A mí, en este artículo, me interesa resaltar un aspecto que el artículo citado de Darren (y toda la excelente monografía de la que forma parte) trata, pero sólo de forma secundaria: el de las pérdidas y molestias ocasionadas por el uso de sistemas una vez que fueron dados de alta para su explotación.

Incluso, cuando no hay evidencia de defectos o fallos en el proyecto, pueden evidentemente darse fallos en la explotación.

La tabla presentada más arriba reseña unos cuantos fiascos, pero los presenta desde la óptica del promotor del proyecto (pérdidas o costes incurridos por el fracaso), no desde la óptica de los costes de los perjuicios ocasionados a usuarios por la operación de sistemas defectuosos o por la operación defectuosa de sistemas.

En 2007 el Ayuntamiento de Torreldones (Madrid) (por lo que parecía tratarse de un simple error administrativo) emitió dos veces recibos (distintos) pro-forma para el cobro de la tasa de recogida de basura.

La empleada / funcionaria encargada del tema me contesta al teléfono (a la tercera llamada; en las anteriores "estaba desayunando") y reconoce que sí, que lo que me acontece es un error ("parte de un error masivo") y que "... nada, está arreglado". Cuando entonces le pido confirmación fehaciente de que lo mío (ya que no lo de todos) ha sido corregido, como error masivo del Ayuntamiento, que es de su responsabilidad, me dice imperturbable que "¡para eso, presente un recurso!", y que entonces se revisará mi duplicado¹².

O sea que unos miles de recibos erróneos se corrigen con unos miles de recursos administrativos. Mientras, yo sigo teniendo dos recibos

pro-forma enviados por el Ayuntamiento: el correcto y el erróneo. ¿O serán los dos erróneos? ¿Y los demás masivos? Afortunadamente, pagado que fue uno de los recibos, el silencio administrativo condonó (espero) el otro.

¡Séneca tenía razón: perseverar en el error es diabólico!

En cuarto lugar, está la cuestión del huevo o la gallina (o más académicamente, de la recursividad). Me parece evidente, a estas alturas de este breve e informal artículo, que todos debemos aceptar que en la raíz de todo error "informático" hay un error humano.

La cuestión ahora es de trazabilidad y, sobre todo, de imputabilidad (*chargeability*). ¿En qué estrato(s) organizativo(s) radica la responsabilidad por el error?

En quinto lugar, está la cuestión del temido y temible 'fallo sistémico'. Parece mentira que los denunciantes de 'fallos informáticos' (políticos o periodistas, muchos de ellos, por lo que no es tan sorprendente; profesionales, en otras ocasiones, más difícil de comprender) no entiendan o ignoren el principio básico de que **NUNCA se debe culpar al sistema**. (¡Aunque fuese verdad!).

Culpar al sistema es concitar un 'fallo sistémico': la profecía que se autocumple.

Los protocolos de actuación de portavoces de células de emergencia, entidades en crisis, y (a mucha menor escala) los de los "chaquetas rojas" (o "verdes" o "azules") en aeropuertos u otros servicios públicos coinciden: siempre hay que decir: a) que las causas se desconocen y se están investigando; y b) que se ha debido probablemente a un fallo humano: NUNCA a un FALLO DEL SISTEMA. Reconocer la causa humana es atenerse a la verdad, pero además es políticamente interesante: ¿Si falla el sistema, qué nos queda?

De hecho, ciertos manuales de gestión de incidentes recomiendan acusar en público (incluso en falso, o sin suficiente información) a algún empleado y luego disculparse ante los vejados y compensarles de algún modo.

En sexto lugar, reconocer que "un sistema informático (por el hecho de estar construido por humanos y/o especificado o solicitado por humanos) es imposible que sea infalible"¹⁰.

En séptimo lugar, recordar que, por una parte, **los errores** (humanos) **pueden no ser indeseables** sino tolerados e incluso buscados (como en ciertas fases del proceso científico o técnico: prueba y error); y por otra que (como se sabe en gestión de la calidad) no todo error acarrea necesariamente una disfunción o un defecto.

Los defectos o fallos son discrepancias entre producto / proceso y especificaciones / estándares (o características deseables, incluso no especificadas).

"Un 'defecto' [/fallo] es una instanciación individual de alguna no conformidad con algún requisito, mientras que un producto/proceso / servicio defectuoso [defective], contiene uno o más 'defectos'¹¹."

Los defectos se clasifican generalmente en tres grandes grupos. Estos grupos son¹³: "1) defectos *inherentes* (resultantes de la fabricación o de las materias primas); 2) defectos *fabricados* (resultantes de la transformación de las materias primas en una pieza, producto o servicio terminado); y 3) defectos *inducidos por el servicio* (generados durante el funcionamiento de algún componente)".

Los defectos pueden deberse a una materia prima / componente

defectuoso, o a una pieza / componente defectuoso (procedente de un proceso previo), a una puesta a punto ('setup') / ensamblaje o mantenimiento defectuoso de la maquinaria / material, a métodos erróneos, o a errores humanos¹⁴.

A la postre, todos los defectos se deben a errores humanos. Esta es la razón por la que la lucha contra los defectos es fundamentalmente una cuestión de formación y motivación del personal.

Los errores pueden clasificarse según 10 grandes criterios¹⁵: 1) omisiones; 2) errores debidos a falta de comprensión; 3) errores de identificación; 4) errores debidos a falta de experiencia; 5) errores voluntarios (consentidos); 6) errores inadvertidos; 7) errores debidos a lentitud; 8) errores debidos a falta de normas o estándares; 9) errores por sorpresa; y 10) errores intencionales.

No todos los errores causan defectos / fallos. Sin embargo, se han estudiado y publicado correlaciones fuertes entre errores y defectos¹⁶. Por ello se conocen diversas grandes estrategias de prevención de defectos.

Por otra parte, muchos defectos / "defectuosos" [*defective*] se pueden prevenir-evitar con técnicas de poka-yoke¹⁷ (si bien éstas pueden resultar más fáciles en el mundo "real" (industria manufacturera, por ej.) que en el "virtual" (aplicaciones web, por ej.).

Frente a errores, defectos y fallos sólo hay una panacea: calidad. Pero calidad integral.

En octavo lugar, aceptar que "[L]os **usuarios** (y sobre todo los **responsables últimos** del trabajo de los usuarios) deben saber que un sistema informático... es imposible que sea infalible. Y por lo tanto **son tanto, o yo diría que aún más responsables, ellos de las consecuencias de no verificar** que el sistema funciona como los propios informáticos".

"Una vez en cierta empresa donde trabajé [Llorenç Pagés] hace ya muchos años nos asignaron un supervisor del trabajo informático una de cuyas frases favoritas era: '*Me gusta la Informática porque es una ciencia exacta, es como las Matemáticas o el ajedrez, exacta*'. Claro, con esta mentalidad ¡cualquier error en un sistema informático es culpa del informático!!"¹⁰.

Para concluir: los sistemas informáticos son frecuentemente imputados de haber causado errores, en ocasiones graves. Ello es falso (por cuanto el origen de todo error es humano) e imprudente (por cuanto culpar al sistema degrada su credibilidad).

La presunción de inocencia es ciertamente un derecho. También de los sistemas y de quienes los desarrollan. Pero la historia demuestra que los derechos se conquistan y que, hasta que son ampliamente reconocidos y respetados, hay que invertir mucho activismo y tiempo en ello.

Notas

¹ <http://www.elpais.com/articulo/espana/juez/sustituye/fianza/millon/euros/alcalde/Sesena/10000/elpepuesp/20100114elpepunac_13/Tes>.

² <http://www.elpais.com/articulo/sociedad/Francia/ofrece/Espana/intercambio/residuos/nucleares/elpepusoc/20100505elpepisoc_3/Tes>.

³ Telediaro 1 TV1, 4 mayo 2010.

⁴ <http://www.elpais.com/articulo/economia/EE/UU/busca/ordenador/fantasma/elpepueco/20100507elpepueco_13/Tes#>>.

⁵ <http://www.elpais.com/articulo/economia/EE/UU/replantea/negociacion/electronica/derrumbe/bursatil/elpepueco/20100508elpepieco_7/Tes>.

⁶ <http://www.answerbag.com/q_view/666955>.

⁷ [IBM, desplazando cada carácter al precedente, según algunos] <http://en.wikipedia.org/wiki/HAL_9000>.

⁸ El autor de la feliz frase, que reproduzco con permiso, es **Didac López**, en correo cruzado con motivo de la preparación de este artículo.

⁹ Fuente: **Eduardo Palao**. Del Caos al Buen Gobierno: Paradigmas y Tendencias en las Tecnologías de la Información y las Comunicaciones y su relación con el Buen Gobierno. Trabajo Fin de Máster. MAGTIC 2. Usado con autorización.

¹⁰ Contribución de **Llorenç Pagés Casas**, con motivo de la preparación de este artículo.

¹¹ **Darren Dalcher**. "El éxito de los proyectos de software: yendo más allá del fracaso". *Novática 200*. julio-agosto 2009, p 45.

¹² Donde resido y tengo mi despacho. Por si alguien manifestara más interés, tengo un expediente completo y detallado. Naturalmente, sin que el incidente se haya resuelto satisfactoriamente aún (junio de 2010).

¹³ **J.M. Juran**, Ed. *Juran Quality Handbook*, 1979.

¹⁴ **G.P. Hayward**. *Introduction to Nondestructive Testing*, ASQC, Wisconsin, 1978, p. 2.

¹⁵ **Hiroyuki Hirano**. *Kaizen*, 1988.

¹⁶ **J.M. Juran**, Ed. *Juran Quality Handbook*, 1979, pp. 18-22.

¹⁷ **Hiroyuki Hirano**, **Nikkan Kogyo Shimbun**. *Poka-yoke*, Ernst & Young, 1991. Ver también <<http://es.wikipedia.org/wiki/Poka-yoke>>.

Manuel Palao García-Suelto (Coordinador de la Sección Técnica "Auditoría SITIC")

Qué hacer cuando el entorno digital falla y no morir en el intento

Estos días de julio debido a motivos profesionales he estado visitando Liverpool y Londres. Cual no sería mi sorpresa cuando comprobé que en el hotel en el que estaba alojado en Liverpool tenía estropeada la conexión a Internet gratuita en recepción. Ni siquiera pagando una tarifa existía posibilidad de servicio. Por supuesto, al viaje me había llevado una serie de *gadgets* que suelo utilizar a diario, el teléfono móvil que tiene posibilidades de e-mail y navegación, una PDA donde además de disponer de GPS puede conectarse a Internet mediante Wifi, y por supuesto el portátil para trabajar en cosas más pesadas.

Pasé tres días haciendo reclamaciones en recepción a ver qué pasaba con la conexión y obteniendo la llamada por respuesta: que si el operador no da servicio, que si depende de los servicios centrales de la cadena hotelera... En resumen, al final tuve que ir a una plaza en el centro de Liverpool donde había un conocido local de comida rápida y zamparme una buena hamburguesa para poder bajar el correo, leerlo, y al menos responder a lo más urgente, entre ello a esta convocatoria de referencias de *Novática*.

En Londres, dado que el hotel era de la misma cadena, tampoco había conexión gratuita pero aquí pagando una tarifa relativamente alta (mejor dicho abusiva) ya tuve conexión. El operador era distinto y pude continuar con mi trabajo sin necesidad de tomar más riesgos con respecto a mi colesterol.

Diego Gachet Páez (Coordinador de la Sección Técnica "Entorno digital personal")

Redes sociales: ¿Dónde está mi privacidad?

"Redes sociales" es un término que oímos por todas partes. Está de moda como muchos otros que aparecen y desaparecen en el mundo de Internet después de un boom, dejando un buen o mal sabor de boca.

En las redes sociales encontramos herramientas que han tenido una gran aceptación y uso pero de las que, al día de hoy, desconocemos muchos aspectos que generan incertidumbre, tales como la privacidad de los contenidos que publicamos: ¿dónde queda nuestra privacidad? ¿somos conscientes del alcance de los contenidos que publicamos?

Por este, y otros motivos, el W3C celebró en enero del año pasado un taller al que dio el nombre de "*W3C Workshop on the Future of Social Networking*", cuyo objetivo era reunir a expertos en el área para hablar

sobre temas relacionados con arquitecturas de redes sociales distribuidas y menos centralizadas, la portabilidad de datos, la experiencia de usuario, así como la privacidad y la confianza. Y es que la web social tiene que enfrentarse, entre otros, a un gran reto relacionado con la privacidad y la confianza del usuario en estos entornos.

La posibilidad de que el usuario pueda fragmentar su identidad a través de varios perfiles puede significar un avance en cuanto a la protección de datos publicados en estas redes sociales, pero el problema va mucho más allá y no parece resolver ni mucho menos la laguna de privacidad detectada... queda trabajo por hacer. Toda esta temática, que parece interesar a expertos de diferentes empresas, es lo que ha llevado a W3C a crear el *Social Web Incubator*, que tratará estos y muchos otros temas, relacionados con la web social.

Y es que los contenidos que publicamos quedan visibles a todas aquellas personas vinculadas a nuestra red de contactos, tanto si son amigos, como si son sólo conocidos o compañeros de trabajo, ya que no se distingue a la hora de visualizar esta información. El problema es que a veces no se es consciente de este aspecto y seguramente se publique contenido que se desea compartir con una parte de las personas de nuestra red social y no con otros, atendiendo a muchos motivos; quizá los mismos por los que en la vida real hablamos con unas personas y no con otras para contar algo de nuestra vida privada, enseñar una foto, o un video, etc.

Uno de los problemas es que es difícil trasladar el esquema de privacidad de la vida real a la vida *online*, y lo que queremos compartir sobre nuestra vida o nuestro trabajo depende mucho de quien es cada una de las personas "amigas" o "conocidas", y de la vinculación o nivel de relación que tengamos con ellas.

Probablemente deseemos compartir fotos de nuestra última fiesta con los amigos y amigas, pero no con compañeros de trabajo u otras personas vinculadas a nosotros, con las que no tengamos una relación muy estrecha.

Por ahora, si es que no sale nada nuevo en este terreno de arenas movedizas, al usuario le queda el buen sentido común y tener precaución con aquellos contenidos que quiere compartir en sus redes sociales, porque, ya se sabe, luego hay que ir al trabajo...

Encarna Quesada Ruiz (Coordinadora de la Sección Técnica "Estándares web")