

# novática



CEPIS **UPGRADE**

Revista de la Asociación  
de Técnicos de Informática

Nº 199, mayo-junio 2009, año XXXV

Software libre para empresas

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

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

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

**Consejo Editorial**

Joan Batlle Montserrat, Rafael Fernández Calvo, Luis Fernández Sanz, Javier López Muñoz, Alberto Libel Batllori, Gabriel Martí Fuentes, Josep Molis i Bertran, José Onofre Montes Adame, Oleg Pallás Codina, Fernando Pira Gómez (Presidente del Consejo), Ramon Puigjaner Trepal, Miquel Sarries Grinó, Adolfo Vázquez Rodríguez, Asunción Yturbe Herranz

**Coordinación Editorial**

Llorenç Pagés Casas <pages@ati.es>

**Composición y autedición**

Jorge Liácer 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 (Opennet), <jmgomez@yahoo.es>

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

**Administración Pública electrónica**

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

**Arquitecturas**

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

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

**Análisis STIC**

Marina Tourño Troitillo, <marinatourino@marinatourino.com>

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

**Base de datos y tecnologías**

Isabel Hernández 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 Pareja Flores (DSIC-UPV), <cpareja@dsic.upv.es>

J. Angel Velázquez Irujibe (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**

Encarnación Quesada Ruiz (Pez de Babel) <equesada@pezdebabel.com>

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

**Gestión del conocimiento**

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

**Informática y Filosofía**

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

Karim Ghaleb Martin (Heriand University) <kgherab@gmail.com>

**Informática Gráfica**

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

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

**Inteligencia del Software**

Javier Dolado Cosin (DLSI-UPV), <dolado@lsi.ehu.es>

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

**Inteligencia Artificial**

Vicente Botti Navarro, Vicente Julián Inglada (DSIC-UPV) <vbotti,vinglada@dsic.upv.es>

**Información Persona-Computador**

Pedro M. Latorre Andrés (Universidad de Zaragoza, AIPO) <platorre@unizar.es>

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

**Lengua e Informática**

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

**Lenguajes Informáticos**

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

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

**Lingüística computacional**

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

Manuel Palomar (Univ. de Alicante), <mpalomar@cc.uja.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), <mikelxpa\_uni@yahoo.es>

**Profesión Informática**

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

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

**Redes y servicios telemáticos**

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

Germán Santos Boada (UPC), <german@ac.upc.es>

**Seguridad**

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

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

**Sistemas de Tiempo Real**

Alejandro Alonso Muñoz, Juan Antonio de la Puente Alfaro (DIT-UPM), <galtonso,puente@dit.upm.es>

**Software Libre**

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

Israel Herráiz Tabernero (UCM), <herrera@computer.org>

**Tecnología de Objetos**

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

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

**Tecnologías para la Educación**

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

César Pablo Corcoles Brinogo (UOC), <ccorcoles@uoc.edu>

**Tecnologías y Empresa**

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

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

**Tendencias tecnológicas**

Alonso Alvarez García (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@loc.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. 91 40 29 391; fax 91 30 93 685 <novatica@ati.es>

**Composición, Edición y Redacción ATI Valencia**  
 Av. del Remo de Valencia 23, 46105 Valencia  
 Tfn./fax 96 33 30 392 <secreval@ati.es>

**Administración y Redacción ATI Cataluña**  
 Via Laietana 46, ppal. 1º, 08003 Barcelona  
 Tfn. 93 41 29 235; fax 93 41 27 713 <secrecat@ati.es>

**Redacción ATI Aragón**  
 Lagasca 9, 3-B, 50006 Zaragoza.  
 Tfn./fax 97 62 35 181 <secreara@ati.es>

**Redacción ATI Andalucía** <secreand@ati.es>

**Redacción ATI Galicia** <secregal@ati.es>

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

**Publicación**  
 Padilla 66, 3º, dcha., 28006 Madrid  
 Tfn. 91 40 29 391; fax 91 30 93 685 <novatica@ati.es>

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

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

**Partida:** C掖erinto hacia arriba - Donde Arias Pérez / © ATI

**Diseño:** Fernando Agresta / © ATI 2003

**editorial**  
**Comunicado de Prensa de ATI sobre la creación del Consejo de Colegios de Ingeniería Informáticas (17/6/2009) en resumen** > 02  
**Un mundo dinámico donde lo importante es sumar** > 02  
*Llorenç Pagés Casas*  
**IFIP**  
**Reunión del TC6 (Communication Networks)** > 03  
*Ramón Puigjaner Trepal*  
**Reunión del TC1 (Foundations of Computer Science)** > 03  
*Michael Hinchey, Joaquim Gabarro Vallés*

**monografía**  
**Software libre para empresas**  
*(En colaboración con UPGRADE)*  
 Editores invitados: *Jesús M. González-Barahona, Teo Romera Otero y Björn Lundell*  
**Presentación. Software libre para empresas: icrea tu producto, alimenta tu comunidad y disfruta tu parte!** > 04  
*Jesús M. Gonzalez-Barahona, Teófilo Romera Otero, Björn Lundell*  
**El software libre en el mundo corporativo** > 08  
*Jesús M. Gonzalez-Barahona, Teófilo Romera Otero, Björn Lundell*  
**Buenas prácticas para la adopción del software libre** > 12  
*Carlo Daffara*  
**Construir y mantener una comunidad de práctica: método aplicado a proyectos de software libre** > 17  
*Stéphane Ribas, Michel Cezon*  
**Dinamización de comunidades en proyectos de software libre** > 21  
*Martin Michlmayr*  
**La Comunidad Morfeo: estrategias Open Source para la Open Innovation** > 25  
*Cristina Breaña, Andrés Leonardo Martínez Ortiz*  
**Aplicación de los principios del software libre en líneas de producto** > 29  
*Frank van der Linden*  
**Abordar las necesidades de la industria en Software Libre** > 36  
*Jan Henrik Ziesing*  
**SpagoWorld, la iniciativa de software libre de Engineering** > 39  
*Gabriele Ruffatti*  
**Una oportunidad para las empresas de software libre: mercado emergente en los países en vías de desarrollo** > 44  
*Susana Muñoz Hernández, Jesús Martínez Mateo*

**secciones técnicas**  
**Enseñanza Universitaria de la Informática**  
**Un juego de rol para la enseñanza de la profesión informática** > 47  
*Agustín Cernuda del Río, Manuel Quintela Pumares, Miguel Riesco Albizu*  
**Gestión del Conocimiento**  
**Datos, conocimiento, información... en este orden** > 51  
*Joan Baiget Solé*  
**Informática Gráfica**  
**Tratamiento de geoinformación a nivel peatonal: animación 3D a partir de metainformación Exif** > 53  
*Ricardo Navarro Moral, Francisco R. Feito Higuera, Rafael J. Segura Sánchez, Angel L. García Fernández*  
**Tecnologías para la Educación**  
**Influencia en el rendimiento académico de la interacción en línea de los alumnos: estudio y análisis comparativo entre diferentes modalidades de enseñanza** > 59  
*Ángel Hernández García, Santiago Iglesias Pradas, Julián Chaparro Peláez, Félix Pascual Miguel*  
**Referencias autorizadas** > 63

**sociedad de la información**  
**Prospectiva Tecnológica**  
**Un metaanálisis de la actividad prospectiva internacional en el campo de los Sistemas y Tecnologías de la Información** > 70  
*Víctor Amadeo Bañuls Silvera, José Luis Salmerón Silvera*  
**Personal y transferible** > 73  
**Una paradoja divina** > 73  
*Antonio Vaquero Sánchez*  
**Programar es crear**  
**Reconstrucción (CUPCAM 2007, problema G, solución)** > 75  
*Enrique Martín Martín, Cristóbal Pareja Flores*

## Comunicado de Prensa de ATI sobre la creación del Consejo de Colegios de Ingeniería Informática (17/6/2009)

El pasado 9 de junio se produjo la aprobación unánime por el Congreso de los Diputados de las proposiciones de ley de creación del Consejo General de Colegios Oficiales de Ingenierías Informáticas presentadas por el Grupo Socialista y el Grupo Popular.

La Asociación de Técnicos de Informática (ATI), creada en 1967, quiere expresar su opinión ante este hecho, como entidad de ámbito estatal de la que forman parte cerca de 4.000 profesionales informáticos de todo el país, casi la mitad de ellos titulados en Ingeniería Informática Superior y Media o Licenciados en Informática, y los demás titulados en otras materias y de otros niveles como Formación Profesional, así como también no titulados y estudiantes de Informática en sus diversas vertientes.

1. En primer lugar ATI considera que la creación del Consejo de Colegios de Ingenierías Informáticas equipara a éstas con las demás Ingenierías, que cuentan con este tipo de órganos desde hace años.

2. ATI declara su disposición a colaborar con el Consejo una vez que se haya formado, para defender a los Ingenieros Informáticos contra toda injusta discriminación y para contribuir a la mejora del conjunto de la profesión informática y al desarrollo armónico de la Sociedad de la Información. Esta disposición a colaborar se extiende a los Colegios que formen parte del citado Consejo, tal como ATI les ha expresado ya en diversas ocasiones.

3. ATI recuerda que los intereses de los profesionales informáticos que no son Ingenieros Informáticos deben ser también objeto de reconocimiento y tutela, tanto por las empresas como por las Administraciones Públicas, pues es un hecho evidente que la profesión informática es multidisciplinar y abarca también a multitud de profesionales que no son Ingenieros Informáticos o Licenciados en Informática (en concreto, titulados de otras carreras con especialidades informáticas, graduados de la rama de Informática de Formación Profesional, poseedores de certificados otorgados por diversas empresas y organizaciones, e incluso personas sin titulación alguna).

De hecho, en el momento presente el número de profesionales informáticos en activo que no tienen título de Ingeniería Informática es muy superior al de éstos.

4. En correspondencia con el punto anterior es obvio que el Consejo de Colegios será representativo de los intereses profesionales colegiales de los Ingenieros Informáticos del país pero no podrá serlo de la totalidad de los profesionales informáticos, pues por su propia naturaleza corporativa no puede representar a quienes no son titulados en Ingeniería Informática, teniendo en cuenta además que la colegiación no es obligatoria y que no en todas las Comunidades Autónomas existen Colegios.

En este sentido ATI, creada en 1967, al ser la

mayor y más antigua de las asociaciones de profesionales informáticos de nuestro país, también se considera representativa del conjunto de los profesionales informáticos en el ámbito asociativo y no reservado por ley exclusivamente a los Colegios.

5. Es preciso subrayar que, a nivel internacional, ATI es desde hace años es el único representante oficial de los profesionales informáticos españoles (en Europa a través de CEPIS, <www.cepis.org>, y en el mundo a través de IFIP, <www.ifip.org>). ATI continuará ejerciendo plenamente esa representación pues ninguna ley española puede decidir sobre el funcionamiento de organizaciones supranacionales como las citadas.

6. Por último debemos resaltar que en el ámbito profesional y laboral ha existido siempre una convivencia armoniosa entre los profesionales informáticos que cuentan con títulos de Ingeniería Informática y quienes no los poseen, convivencia que debe ser preservada frente a posibles visiones exclusivistas y discriminatorias.

Como conclusión, todo lo expuesto anteriormente hace más evidente si cabe la necesidad de colaboración, sin exclusiones, entre las distintas organizaciones que representamos a los profesionales informáticos españoles, así como la interlocución de todas ellas, también sin exclusiones, con las Administraciones Públicas implicadas en estas materias.

## en resumen Un mundo dinámico donde lo importante es sumar

Llorenç Pagés Casas

Coordinación Editorial de *Novática*

A estas alturas empieza a resultar evidente que estamos en un "nuevo" mundo, cada vez más dinámico, donde las interrelaciones, tanto entre personas como entre entidades, son cada vez más plurales, cambiantes e incluso imprevisibles, y donde la mejor forma de reducir toda la incertidumbre generada consiste en ser positivos, cooperar y sumar.

Es este un mundo apasionante donde los grandes avances de esta época, tanto tecnológicos (redes) como sociales (mayor igualdad de oportunidades) y filosóficos (dar antes de recibir), están marcando el camino.

Si tuviéramos que buscar una muestra de todo ello en el ámbito de la Informática, muy

probablemente deberíamos escoger los llamados "ecosistemas" de software libre. Puesto que es en el campo del software libre donde se dan mejor esas grandes paradojas, signo de los nuevos tiempos, como son la cooperación con los competidores o el beneficiarnos de la aportación gratuita de (parte de) nuestros productos.

Es por esto que la monografía que presentamos en este número y que han coordinado **Jesús M. Gonzalez-Barahona** y **Teófilo Romera Otero** (Universidad Rey Juan Carlos), y **Björn Lundell** (Universidad de Skövde, Suecia) nos trae una excelente noticia: empresas europeas de todos los tamaños y de los más variados sectores están apostando decididamente por este modelo, que escapa ya de ser un mero modelo de gestión del software para integrarse en una novedosa

corriente de hacer negocio, y en definitiva de afrontar la vida.

La constatación (hay mucho que leer sobre ello en la monografía) de la cooperación de esas empresas en comunidades de software libre, asumiendo los consecuentes retos e incertidumbres en la expectativa de importantes beneficios "comunitarios", es una buena muestra de que vamos camino de desterrar ese mundo demasiado compartimentalizado, tanto en lo económico como en lo ideológico, que heredamos de nuestros padres y abuelos.



# Reunión del TC6 (Communication Networks)

Ramón Puigjaner Trepas

Delegado Permanente de ATI en IFIP y en el TC6, Vocal de la Junta Directiva General de ATI

<putxi@uib.es>

Fue una de las reuniones del TC6 a la que asistió un mayor número de miembros: 31 más Eduard Dundler, del secretariado de la IFIP en Laxenburg (Austria). Tuvo lugar en Aachen (Alemania) los días 14 y 15 de mayo a continuación de *Networking 2009*, organizada en la misma ciudad por el Prof. Otto Spaniol, tal vez la edición mejor organizada desde que se instauró esta conferencia. Como actividad notable y singular de esta reunión fue la conferencia que impartió James Roberts de *France Telecom* sobre las nuevas tendencias y retos en Internet.

Aparte de una serie de asuntos de trámite sobre países que dejarán de ser miembros de IFIP por falta de pago de la cuota, se comentó la participación del TC6 en el próximo *World Computer Congress WCC 2010* que se celebrará en septiembre del próximo año en Brisbane (Australia) coincidiendo con la celebración del jubileo de IFIP con motivo de

cumplir 50 años. Esta participación se materializará en el *3rd. IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, WCITD'2010*, a propuesta del grupo de trabajo WG6.9 *Communication Networks for Developing Countries*, contando con la participación del WG6-11 *E-commerce* en una conferencia sobre *E-Governmenty*, como actividad principal, con la conferencia sobre *Future Internet*.

Hay que recordar que este WCC es el último que se piensa organizar con el formato actual de diversas conferencias desarrollándose en paralelo. Está por ver si habrá WCC en 2012 y, en caso de que lo haya, qué estructura tendrá.

Se discutió la planificación de las actividades (conferencias, *workshops*, *symposia*, etc.) de los diferentes WG realizadas en los meses

pasados y a realizar por ellos en los meses a venir. Entre las primeras, aparte de las habituales conferencias y *workshops*, conviene resaltar la *2nd Summer School on Networking*, dedicada en el año 2008 a las comunicaciones inalámbricas y celebrada en Pretoria (Sudáfrica) en los primeros días de octubre con una notable participación española entre el cuerpo docente (tres de los cinco profesores eran españoles).

Otro punto importante que fue tratado es el de las características de calidad que deben seguir las distintas actividades organizadas bajo el patrocinio del TC6 y los resultados que cabe esperar de ellas.

Se planificaron las próximas reuniones probablemente en París en diciembre de este año coincidiendo con los *IFIP Wireless Days* y en Chennai (antes Madrás, India) en mayo de 2010 coincidiendo con *Networking 2010*.

## Reunión del TC1 (Foundations of Computer Science)

El TC1 se reunió en la Universidad de York (Reino Unido) el 22 de marzo de 2009. Se trataron los siguientes puntos:

- Se discutió la situación de los miembros en suspenso (recordemos que los miembros de IFIP son países, no personas) que son los que no han pagado la suscripción a IFIP durante los últimos tres años. Los representantes nacionales de los miembros suspendidos ya no forman parte de la Asamblea General (GA) y no son elegibles como miembros de categoría A de los TC (Comités Técnicos). Pueden ser miembros de las categorías B o C de los TC. También pueden ser miembros de los WG (Grupos de Trabajo). Como los miembros de las categorías B y C tienen que ser menos que los miembros de la categoría A, la pérdida de miembros de categoría A reduce el número de miembros de categorías B y C que los TC pueden tener.

- Es preocupante la situación de Francia. Los miembros franceses son muy activos en

IFIP, pero con la suspensión de Francia parece que no hay en aquel país ninguna sociedad que pueda ser sociedad nacional.

- Se discutió el modelo de IFIP como sponsor. Este modelo se está discutiendo también dentro de la GA (General Assembly) y las TA (Technical Activities) de IFIP.

- Se recordó que los fondos de IFIP para el desarrollo igualan las contribuciones para una de las conferencias de los TC por año en un país en desarrollo. Se está discutiendo un nuevo modelo de financiación en la GA y en los comités correspondientes.

- La edición de la colección de libros IFIP se ha movido a Heidelberg, aunque Springer continuará editando la colección. El nuevo título de la serie es "*IFIP Advances in Information and Communication Technology*". El nuevo comité editorial incluirá inicialmente un representante de cada TC. El TC1 acuerda que su representante sea el *chair*.

- Se discutieron algunos planes para el WWC 2010 y en particular para el TCS 2010 en Brisbane. Los profesores James Harland (Melbourne) y Barry Jay (Sydney) fueron propuestos como *co-chairs* de TCS 2010. Vladimiro Sassone (Southampton) y Cristian Calude (Auckland) fueron propuestos como *co-chairs* del comité de programa. Posteriormente, los cuatro aceptaron las propuestas.

Finalmente resaltar que York es una ciudad que merece ser visitada. Además alberga el NRM, National Railway Museum. Dicho museo guarda una réplica de la máquina de los Stephenson.

**Michael Hinchey<sup>1</sup>,  
Joaquim Gabarro Vallés<sup>2</sup>**

<sup>1</sup>Lero, the Irish Software Engineering Research Center, Chairman del TC1  
<sup>2</sup>Universitat Politècnica de Catalunya, Representante de ATI en el TC1

<mike.hinchey@usa.net>,  
<gabarro@lsi.upc.edu>

Jesús M. Gonzalez-Barahona<sup>1</sup>, Teófilo Romera Otero<sup>1</sup>, Björn Lundell<sup>2</sup>

<sup>1</sup>Universidad Rey Juan Carlos, Madrid;

<sup>2</sup>Centro de Investigación en Informática de la Universidad de Skövde, Suecia

<{jgb,teo}@gsyc.es>,  
<bjorn.lundell@his.se>

En los últimos años la relevancia que el software libre ha ido ganando en diversos ámbitos y niveles no ha hecho más que crecer. Está presente en los mercados, en las administraciones públicas y es objeto de estudio y valoración por parte de académicos, empresas e instituciones de gobierno. Se modelan procesos de desarrollo de software siguiendo pautas nacidas en las comunidades de software libre, se diseñan modelos de negocio para empresas a partir de actividades típicas de los proyectos de software libre, los partidos políticos comienzan a preparar acciones al respecto del software libre en sus propuestas electorales y los gobiernos diseñan leyes respecto a su uso. E incluso se estudian modelos económicos alternativos en base a las características que han hecho posible una comunidad fuerte con capacidad para coordinar profesionales, empresas, voluntarios e instituciones públicas alrededor de productos de calidad y disponibles para cualquiera.

Más allá del software, la cultura libre influye sin duda y a velocidad de vértigo en diversas facetas de las tecnologías de la información, el arte, la cultura y en definitiva en nuestras vidas. Elementos como la Wikipedia, la innovación abierta, la meritocracia o las licencias alternativas para el arte y la cultura basadas en los fundamentos introducidos por el software libre (como las licencias Creative Commons), las comunidades para compartir y crear en Internet como Flickr, y tantas otras novedades han sido absorbidas por nuestra sociedad líquida (que diría Zygmunt Bauman) con pasmosa naturalidad y a un ritmo imparables.

Los profesionales de las tecnologías de la información y la comunicación, así como las empresas del sector no pueden pasar por alto estas nuevas tendencias y tanto es así que resulta difícil encontrar una sola de estas empresas que no tenga ya una estrategia clara respecto al software libre, bien sea para competir con él, para adoptarlo en sus modelos de negocio o para dedicarse en exclusiva a su creación y explotación.

Este número monográfico especial ha sido confeccionado contactando con diferentes expertos profesionales e investigadores del mundo de la empresa, del software libre y de las relaciones entre empresa y comunidades de software libre. No pretendemos que sea una recopilación exhaustiva pero sí represen-

# Presentación Software libre para empresas: icrea tu producto, alimenta tu comunidad y disfruta tu parte!

## Editores invitados

**Jesús M. Gonzalez-Barahona** es profesor e investigador en la Universidad Rey Juan Carlos de Móstoles (Madrid). Comenzó a involucrarse en software libre en 1991. Desde entonces ha colaborado en numerosos grupos de trabajo, ha desarrollado algunas líneas de investigación y ha comenzado programas de formación diversos en la materia. También colabora con varios proyectos y asociaciones de software libre, escribe en diversos medios sobre temas relacionados con el software libre y lleva a cabo consultoría para compañías y administraciones públicas sobre software libre, todo ello desde el marco del grupo de investigación GSyC/LibreSoft, <<http://libresoft.es>>.

**Teófilo Romera Otero** es Ingeniero en Informática por la Universidad Rey Juan Carlos de Madrid, donde está actualmente llevando a cabo tareas de investigación para la consecución de su doctorado. Sus intereses académicos son el software de fuentes abiertas y cómo las empresas se relacionan con el. También está interesado en el desarrollo de software global aplicado al desarrollo de software de fuentes abiertas y la habilitación tecnológica a través de software de fuentes abiertas. Trabaja para el grupo de investigación GSyC/LibreSoft como coordinador de proyectos de investigación nacionales y europeos, algunos de ellos: Calibre, Edukalibre, FLOSSWorld, Morfeo, Vulcano, QualiPSO, FLOSSInclude o Tree. Otras tareas que lleva a cabo en el grupo son consultorías técnicas sobre software de fuentes abiertas y la coordinación docente del *Master on Libre Software* organizado por GSyC/LibreSoft. Es miembro del *Qualipso Network Board* que gestiona la red de centros de competencia Qualipso y del grupo de trabajo sobre Software Libre de la plataforma NESSI. Ha realizado estancias de investigación en la universidad de Leeds en el Reino Unido y en el *Irish Software Engineering Research Centre* (Lero) de la universidad de Limerick (Irlanda).

**Björn Lundell** es investigador en el Centro de Investigación en Informática de la Universidad de Skövde. Sus intereses académicos incluyen el software de fuentes abiertas, su evaluación y el soporte de métodos. Fue gestor técnico del proyecto de investigación COSI (*Codevelopment Using Inner & Open Source in Software Intensive Products*). Es miembro fundador del grupo de trabajo IFIP 2.13 sobre software libre y preside la asociación industrial *Open Source Sweeden*. Lundell es doctor en Ciencias de la Computación por la Universidad de Exeter.

tativa del panorama actual en Europa en lo que se refiere a casos descriptivos de cómo las empresas usan y crean software libre, y especialmente cómo se relacionan con otros actores (otras empresas, voluntarios y profesionales) dentro del mismo escenario.

En un primer artículo introductorio, *El software libre en el mundo corporativo*, los editores invitados del presente número, **Jesús M. González Barahona**, **Teófilo Romera Otero** y **Björn Lundell**, ofrecen su visión particular de la importancia de que el mundo empresarial tenga estrategias de acción respecto del software libre. Al mismo tiempo, sirve de introducción a los nuevos términos y conceptos a los que se hace referencia en el resto del número, intentando ofrecer una mirada a vista de pájaro de los motivos y los medios que aplican en este nuevo mundo que es el software libre para empresas.

En su artículo, *Buenas prácticas para la*

*adopción del software libre*, **Carlo Daffara**, reconocido experto en modelos de negocio sostenibles con software libre, recoge un compendio de consejos y directrices que aseguran una migración al software libre exitosa para entornos empresariales. Este artículo abre el monográfico porque resulta muy representativo de los problemas que las empresas suelen encontrar a la hora de incluir el software libre en sus procesos o en sus modelos de negocio y al mismo tiempo puede servir al lector no iniciado como una extensión de la introducción, puesto que maneja aspectos técnicos que resultan familiares al profesional del sector.

Seguidamente, el artículo *Construir y mantener una comunidad de práctica: método aplicado sobre proyectos de software libre* muestra un método para la creación y gestión de comunidades que puede ser aplicado a la creación y gestión de comunidades de software libre y especialmente a aquellas que

como OW2 o Morfeo implican relaciones entre voluntarios, profesionales y empresas. Los autores, **Stéphane Ribas** y **Michel Cezon**, conocen bien tanto la comunidad del software libre como el mundo empresarial y tras su paso por diversas empresas informáticas han recalado recientemente en el mundo académico, para desde el instituto INRIA seguir ayudando a construir buenas relaciones entre comunidades de desarrollo y empresas.

En línea con el artículo anterior, **Martin Michlmyer** nos aporta su trabajo *Dinamización de comunidades en proyectos de software libre*. Se trata de un artículo que complementa muy bien tanto al artículo anterior, sobre la construcción de comunidades, como al que le sigue. Sin duda, se trata de un tema muy interesante y novedoso, puesto que aporta una descripción formal de la figura del dinamizador de comunidades en las grandes empresas. Un fenómeno reciente de importancia creciente y relativamente poco conocido hasta el momento. No en vano, Martin ha sido en parte su propio objeto de estudio y observación puesto que su trayectoria como líder de diversos proyectos de software libre de gran envergadura e importancia (incluyendo Debian) le ha llevado a ser contratado por HP para dinamizar y gestionar las relaciones entre la empresa y las comunidades de software libre.

Con una fuerte relación con los dos artículos anteriores, **Andrés Leonardo Martínez Ortíz** y **Cristina Breña**, co-fundador y responsable de comunicación respectivamente de la comunidad Morfeo liderada por Telefónica I+D, presentan *La Comunidad Morfeo: estrategias Open Source para la Open Innovation*. Morfeo es una comunidad de software en la que participan actores de muy diversa índole y con intereses y capacidades dispares, y aun así exitosa. Se trata pues de un caso de estudio descriptivo de la historia y peculiaridades de la comunidad Morfeo.

Una vez cubierto el tan interesante tema de las

relaciones entre comunidades de desarrollo de software libre y empresas, el número continúa con algunos otros trabajos menos orientados a la creación y mantenimiento de comunidades pero mucho más enfocados a la descripción de casos de éxito, oportunidades y estrategias que las empresas siguen ya o tienen a su disposición para la adecuada explotación de nuevos modelos de negocio basados en software libre.

Desde su amplia experiencia en Philips, **Frank van der Linden**, reconocido experto en software libre y líneas de producto, escribe su artículo *Aplicación de los principios del software libre en líneas de producto* en el que investiga las diversas opciones en las que el software libre y las metodologías de desarrollo asociadas pueden utilizarse para reducir los problemas del desarrollo de software global (GSD o desarrollo distribuido) así como para aumentar la calidad del software que se desarrolla.

**Jan Henrik Ziesing**, científico investigador del Instituto Fraunhofer en sistemas de comunicación abiertos en Berlín, aporta un trabajo sobre las necesidades de la industria en el ámbito de software libre y la participación del Instituto Fraunhofer FOKUS en el proyecto Qualipso, el mayor proyecto del programa marco de la Comisión Europea de los que tratan exclusivamente con el software libre. El artículo lleva por título *Abordar las necesidades de la industria en Software Libre*. Muestra la estrategia que Fraunhofer FOKUS tiene preparada para llevar sus modelos y servicios de interoperabilidad tradicionales al ámbito del software libre, mediante la creación de un nuevo centro de competencia del software libre que ofrezca servicios al tejido empresarial de la región de Berlín.

En el siguiente artículo, *SpagoWorld, la iniciativa de software libre de Engineering*, **Gabriele Ruffati**, director de la unidad de Arquitecturas y Consultoría de la división de I+D en Engineering, muestra un caso de estudio basado en la experiencia de la compa-

ñía. Se describen las razones, la estrategia y las relaciones de Engineering con las comunidades, así como los resultados obtenidos por la empresa en sus productos libres.

Por último y para cerrar el número, **Susana Muñoz Hernández** y **Jesús Martínez Mateo**, profesora y estudiante de doctorado respectivamente, de la Universidad Politécnica de Madrid (UPM) e investigadores ambos del grupo de cooperación TEDECO (TEcnología para el DEsarrollo y la COoperación), presentan el artículo *Una oportunidad para las empresas de software libre: mercado emergente en los países en vías de desarrollo*. En él se describen oportunidades y mecanismos mediante los cuales el software libre puede ayudar a la creación del tejido empresarial en los países en vías de desarrollo.

En definitiva, el presente número especial *Software libre para empresas* pretende mostrar un panorama actual de las iniciativas, estrategias y acciones que las empresas europeas están llevando a cabo respecto al software libre. Hemos intentado recabar un conjunto de artículos equilibrado y de fuentes y procedencias diversas, incluyendo Italia, Francia, Holanda, Alemania y España, con autores de renombre que se sitúan en la franja, cada vez más amplia, en la que se solapan las comunidades de software libre y el mundo empresarial, y ofreciendo puntos de vista por parte de universidades, grandes empresas, individuos y organismos públicos. Esperamos que el resultado sea del agrado del lector y les emplazamos a seguir las evoluciones futuras de la presente publicación.

### Agradecimientos

Los editores invitados de este número especial de *Novática* quieren agradecer al equipo de ATI y en especial a Llorenç Pagés, por su ayuda y buen hacer. También agradecer a Miguel Vidal su ayuda como coordinador de las traducciones y a los traductores del grupo GSyC/LibreSoft por su magnífico trabajo.

### Referencias útiles sobre "Software Libre en la empresa"

Las referencias que se citan a continuación, junto con las proporcionadas en cada uno de los artículos, tienen como objetivo ayudar a los lectores a profundizar en los temas tratados en esta monografía permitiendo contrastar ideas y obtener información actualizada.

#### Libros

■ **Dan Woods, Gautam Guliani.** *Open Source for the enterprise*. O'Reilly Media, Inc. (2005). ISBN-10: 0596101198. Este es un gran libro que explica el software libre desde

el punto de vista empresarial. Ofrece ideas brillantes sobre la gestión de proyectos y la paquetización de software libre.

■ **Jan Sandred.** *Managing Open Source Projects*. John Wiley & Sons, 2001. Un manual sobre cómo abordar los principios y ventajas de la programación de software libre.

■ **Jesús González Barahona, Joaquín Seoane Pascual, Gregorio Robles.** *Introducción al Software Libre*. Es un libro de texto usado en el tema de Software Libre del programa de doctorado de la Universidad Rey

Juan Carlos y en algunos otros programas de la *Universitat Oberta de Catalunya*. Cubre prácticamente todos los temas relacionados con el software libre. Este es un libro imprescindible. <<http://ocw.uoc.edu/computer-science-technology-and-multimedia/introduction-to-free-software/materials/>>.

■ **Karl Fogel.** *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media, Inc. (2005). ISBN-10: 0596007590. Este es un libro sobre el aspecto humano del desarrollo de software

libre. Describe cómo operan los proyectos exitosos, las expectativas de usuarios y desarrolladores, y la cultura del software libre <<http://producingoss.com/>>.

■ **Chris DiBona, Sam Ockman, Mark Stone et al.** *Open Sources: Voices from the open source revolution*. O'Reilly Media, Inc. (1999). ISBN-10: 1565925823. Una clásica y excelente compilación de artículos, ensayos y escritos de importantes actores en la escena del software libre. Entre ellos se encuentran Linus Torvalds, Eric S. Raymond, Richard Stallman, Bruce Perens y muchos otros <<http://oreilly.com/catalog/opensources/book/toc.html>>.

■ **Donald K Rosenberg.** *Open Source: The Unauthorized White Papers*. Hungry Minds, 2000. ISBN-10: 0764546600. Se trata de una visión general del software libre <<http://www.stromian.com/Book/FrontMatter.html>>.

■ **Lawrence Rosen.** *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall PTR, 2004. ISBN-10: 0131487876. Este libro trata de casi todo lo que hay que saber sobre las licencias de

software libre <<http://www.rosenlaw.com/oslbook.htm>>.

■ **Lawrence Lessig.** *Free Culture*. Penguin Press HC, 2004. ISBN-10: 1594200068. Este libro explica cómo los grandes medios de comunicación usan la tecnología y las leyes para bloquear la cultura y controlar la creatividad <<http://free-culture.org/>>.

■ **Karl Fogel, Moshe Bar.** *Open Source Development with CVS*. Paraglyph Inc. (2003). ISBN-10: 1932111816. Explica muchos aspectos del software libre a través de la descripción de técnicas y herramientas de desarrollo. <<http://cvsbook.red-bean.com/>>.

### Otras publicaciones

■ **Qualipso Project.** *Qualipso deliverable on OSS business models* (2008). Explica muchos aspectos de los modelos de negocio del software libre junto con el estado del arte actual <<http://qualipso.org/sites/default/files/media/A2/A2.D1.2.3%20The%20business%20models%20for%20using%20OS.pdf>>.

■ **A. Abella, M. A. Segovia.** *Libro blanco del software libre en España* (2007). <<http://libroblanco.com/document/>

<http://www.rosenlaw.com/oslbook.htm>>.

■ **2020 FLOSS Roadmap.** Esta "hoja de ruta" fue presentada en el OpenWorldForum 2008 en París. En un documento muy útil creado de forma colaborativa por un gran conjunto de expertos de diferentes empresas y entidades <<http://www.2020flossroadmap.org/>>.

■ **Rishab Aiyer Ghosh et al.** *Study on the: Economic impact of open source software on innovation and the competitiveness of the Information and Communication Technologies (ICT) sector in the EU (FLOSSImpact Report 2006)*. Este informe fue elaborado en el marco del proyecto FLOSSImpact <<http://www.flossimpact.eu/>> y una gran cantidad de investigadores participaron en él <<http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf>>.

■ **Qualipso Project.** *Analysis of most important aspects of Open Source Competence Centres*. <<http://qualipso.org/node/48>>.

# Máster en Software Libre

## Universidad Rey Juan Carlos

"Amplía tus conocimientos sobre el mundo software libre"

### Temario

Programa principal 37,5 ECTS  
Seminarios específicos 2,5 ECTS  
Prácticum 12 ECTS  
Tesis fin de máster 8 ECTS



### Metodología

Uso de técnicas de blended learning  
Ponentes de la comunidad GNOME, KDE, Mozilla, Apache, Cherokee, Liferay, etc  
Posibilidad de prácticas en empresas colaboradoras

### Precio

La matrícula del máster tiene un precio de 7.000€, pero gracias a la colaboración de varios organismos y empresas se ofrecen becas y descuentos de hasta el 50%.



Más información en [master.libresoft.es](http://master.libresoft.es) o [master-sw-libre@libresoft.es](mailto:master-sw-libre@libresoft.es)



Jesús M. Gonzalez-Barahona<sup>1</sup>,  
Teófilo Romera Otero<sup>1</sup>, Björn  
Lundell<sup>2</sup>

<sup>1</sup>Universidad Rey Juan Carlos, Madrid;  
<sup>2</sup>Centro de Investigación en Informática de  
la Universidad de Skövde, Suecia

<{jgb,teo}@gsyc.es>,  
<bjorn.lundell@his.se>

### 1. El concepto de software libre, (free, open source) software

Software libre es el término que utilizamos en este artículo para referirnos tanto al "free software" según lo define la Free Software Foundation<sup>1</sup>, como al "open source software" (software de fuentes abiertas) según lo define la Open Source Initiative<sup>2</sup>. Aunque ambas definiciones son diferentes, cubren casi la misma colección de software. En cualquier caso, los términos (al menos en inglés) no son intercambiables. Para algunos, la palabra "free" (que se puede traducir como libre o como gratis al castellano) se refiere a las libertades que el software libre brinda y por esto mismo, nunca debería ser omitida. Para otros, el sentido ambiguo de la palabra "free" que puede entenderse también como "gratis" es un problema para el entendimiento del concepto, especialmente en entornos empresariales. Para evitar esta discusión, usaremos el término software libre a lo largo de este texto.

En cualquier caso, podemos aportar una breve descripción del software libre mencionando las "cuatro libertades". El software libre es aquel que permite a los que lo reciben: ejecutarlo y usarlo; estudiarlo y adaptarlo a necesidades específicas; redistribuirlo a otros; y mejorarlo y añadirle funcionalidad. Todas estas libertades pueden ejercerse una vez que el software es obtenido, sin necesidad de permisos adicionales de los dueños del copyright más allá de la licencia que se aplica para ese software.

Desde este punto de vista, el software libre es principalmente un concepto legal. Define algunos permisos básicos que los dueños del copyright garantizan a aquellos a quienes dan el software. De hecho, los dueños del copyright no ceden todos sus derechos. Algunos quedan reservados en forma de cláusulas y condiciones especificadas en las licencias de software libre.

Desde un punto de vista económico y de negocio, es importante remarcar que no se especifica nada acerca de cómo se obtiene el software. Puede ser obtenido gratis de un repositorio público en Internet o comprado por un sustancioso precio en una tienda, puede que en una bonita caja.

Desde un punto de vista técnico, el software libre no es una tecnología en sí mismo. De hecho, la única característica tecnológica común para que un programa sea software libre, es que el código fuente ha de estar

## El software libre en el mundo corporativo

**Resumen:** el software libre es un nuevo mundo en sí mismo tanto para las empresas como para los profesionales. Es por eso que comprender cómo encontrar oportunidades en él es cada vez más importante a medida que los productos de software libre se usan más y más en la industria del software y en otros sectores que dependen del software para sus actividades. En este texto, se exploran algunos de los nuevos aspectos que aparecen al aproximarse al software libre y cómo las empresas reaccionan a ellos.

**Palabras clave:** comunidades, empresas, estrategia, FLOSS, OSS, software libre.

### Autores

**Jesús M. Gonzalez-Barahona** es profesor e investigador en la Universidad Rey Juan Carlos de Móstoles (Madrid). Comenzó a involucrarse en software libre en 1991. Desde entonces ha colaborado en numerosos grupos de trabajo, ha desarrollado algunas líneas de investigación y ha comenzado programas de formación diversos en la materia. También colabora con varios proyectos y asociaciones de software libre, escribe en diversos medios sobre temas relacionados con el software libre y lleva a cabo consultoría para compañías y administraciones públicas sobre software libre, todo ello desde el marco del grupo de investigación GSyC/LibreSoft, <<http://libresoft.es>>.

**Teófilo Romera Otero** es Ingeniero en Informática por la Universidad Rey Juan Carlos de Madrid, donde está actualmente llevando a cabo tareas de investigación para la consecución de su doctorado. Sus intereses académicos son el software de fuentes abiertas y cómo las empresas se relacionan con él. También está interesado en el desarrollo de software global aplicado al desarrollo de software de fuentes abiertas y la habilitación tecnológica a través de software de fuentes abiertas. Trabaja para el grupo de investigación GSyC/LibreSoft como coordinador de proyectos de investigación nacionales y europeos, algunos de ellos: Calibre, Edukalibre, FLOSSWorld, Morfeo, Vulcano, QualiPSO, FLOSSInclude o Tree. Otras tareas que lleva a cabo en el grupo son consultorías técnicas sobre software de fuentes abiertas y la coordinación docente del *Master on Libre Software* organizado por GSyC/LibreSoft. Es miembro del *Qualipso Network Board* que gestiona la red de centros de competencia Qualipso y del grupo de trabajo sobre Software Libre de la plataforma NESSI. Ha realizado estancias de investigación en la universidad de Leeds en el Reino Unido y en el *Irish Software Engineering Research Centre* (Lero) de la universidad de Limerick (Irlanda).

**Björn Lundell** es investigador en el Centro de Investigación en Informática de la Universidad de Skövde. Sus intereses académicos incluyen el software de fuentes abiertas, su evaluación y el soporte de métodos. Fue gestor técnico del proyecto de investigación COSI (*Codevelopment Using Inner & Open Source in Software Intensive Products*). Es miembro fundador del grupo de trabajo IFIP 2.13 sobre software libre y preside la asociación industrial *Open Source Sweden*. Lundell es doctor en Ciencias de la Computación por la Universidad de Exeter.

disponible (el término "software de fuentes abiertas" hace hincapié en esto). Pero aparte de este hecho, existen algunas particularidades comunes en la manera en que se desarrolla el software libre: modelos de desarrollo abiertos, con más información disponible públicamente; existencia de comunidades de soporte; mezcla de voluntarios y desarrolladores pagados; meritocracia y reglas de gobierno basadas en la comunidad, etc.

Desde un punto de vista ético, la ética *hacker* es una fuerte convicción para muchos de los participantes en los proyectos de software libre, pero no se trata de un aspecto "obligatorio". De hecho, muchos otros no la reconocerán o incluso no saben de qué se trata.

Puede que esta breve introducción a la idea de

software libre sea suficiente para mostrar cómo de complejo es este fenómeno, con implicaciones legales, de negocio, técnicas y éticas. Debido a esta naturaleza diversa y las interrelaciones de los diferentes intereses de la gente que participa, no suele ser fácil de entender para los neófitos.

### 2. Problemas comunes que encuentran las empresas

En algún momento una empresa, que puede estar enfocada a las TIC o bien utilizarlas para otros propósitos, considera el producir, integrar, usar o mejorar servicios basados en software libre. Existen varias razones que pueden llevarla a este punto. Razones ampliamente discutidas en otros lugares y que no serán consideradas aquí. En lugar de centrarse en esas razones, este artículo toma esa

decisión de "pasarse al software libre", que cada vez es más común en el mundo empresarial, como el punto de partida.

De hecho, una vez que se ha tomado esa decisión es cuando viene la parte más difícil. Las empresas han de lidiar, a partir de ese punto, con algunos aspectos que si son manejados con acierto pueden producir beneficios considerables, pero si son pasados por alto, pueden también ser causa de numerosos riesgos y problemas. Muchos de estos aspectos son someramente discutidos en el resto de esta sección.

## 2.1. Lidiar con comunidades

El desarrollo de software libre se suele llevar a cabo en comunidades que no sólo incluyen a los desarrolladores principales, sino también a colaboradores esporádicos o desarrolladores de productos relacionados (por ejemplo de ramas alternativas del mismo software, para el uso corporativo) y usuarios de diferentes tipos (desde usuarios finales que no saben nada del desarrollo a integradores con experiencia que conocen el producto casi tan bien como los desarrolladores principales). Los individuos en estas comunidades pueden ser simplemente voluntarios que dependen de su propia motivación, profesionales contratados por empresas para contribuir de acuerdo a los intereses de estas o incluso alguna combinación de ambos.

Cualquier compañía que se acerque (o que pretenda crear) a una de estas comunidades se enfrentará a un entorno bastante diferente de las relaciones contractuales que mantiene con otras compañías o empleados y a las que está acostumbrada. Por ejemplo, normalmente no hay contratos que regulen las relaciones entre participantes en la comunidad y que puedan comprometer a la gente una vez firmados. Al contrario, la motivación y la creación de relaciones fructíferas para todas las partes es aquí el *modus operandi*. Adaptarse a estas nuevas reglas no es siempre fácil.

La imagen corporativa también adquiere una nueva dimensión en estos entornos. Cuando una empresa (generalmente a través de sus empleados) se convierte en miembro de una comunidad de software libre, los desarrolladores comienzan a forjarse una imagen de la empresa, que normalmente se sobrepone a la imagen "tradicional" de dicha empresa. Otros comienzan a considerarla en términos de cuánto esa empresa contribuye a la comunidad, cuánto se beneficia de ella, cómo colabora con otros participantes dentro de ella o si es percibida como si estuviera intentando controlarla. Esta imagen puede producir todo tipo de efectos secundarios.

## 2.2. Asuntos legales

Las licencias de software libre, aunque se basan en las mismas leyes de propiedad intelectual con las que las empresas de TIC están familiarizadas, son utilizadas de modos muy diferentes. Lamentablemente, sigue siendo

poco común encontrar expertos legales con experiencia y conocimientos plenos en los modelos de licenciamiento libres. Pero analizar los esquemas de licenciamiento con acierto y comprender no sólo sus implicaciones legales, sino también las económicas o las técnicas, es de crucial importancia para las compañías. Los esquemas de desarrollo, la disseminación de nuevas tecnologías o los modelos de negocio se ven normalmente restringidos o potenciados dependiendo de las licencias que entren en juego.

De hecho, las licencias de software libre son de algún modo el tejido que mantiene unidos a todos los actores interesados en un producto de software libre. Dependiendo de la licencia utilizada, de cómo es explicada y cómo se hace cumplir, otros actores podrán estar más o menos interesados en colaborar. Las licencias también brindan una serie de garantías a empresas interesadas en un producto y tienen impacto en muchos aspectos diferentes incluyendo la reutilización de componentes o la dependencia estratégica. Por supuesto que también muestran una fuerte relación con otros aspectos legales que preocupan a las compañías, como las leyes reguladoras de la propiedad industrial (patentes).

A pesar de la importancia de las licencias en el mundo del software libre, su estudio detallado no es siempre simple. De hecho, es un campo que el mundo corporativo suele reconocer como potencialmente problemático.

## 2.3. Nuevos procesos de desarrollo

El software libre puede ser desarrollado usando métodos tradicionales, que son comunes en las empresas de desarrollo de software, pero este no suele ser el caso. Las comunidades de software libre suelen utilizar sus propios procesos, que tienen en cuenta o incluso aprovechan las características particulares del equipo de desarrollo. Una de las principales diferencias que normalmente encuentran las compañías es que los desarrolladores no están atados por contratos o trabajan para una única compañía.

Al contrario, las comunidades de desarrollo de software libre por regla general, están compuestas de una mezcla de voluntarios y profesionales contratados por diferentes empresas. Debido a esta mezcla, las relaciones entre desarrolladores, y entre las compañías y los desarrolladores, son bastante diferentes a las de los proyectos tradicionales, con una jerarquía clara de gestión y restricciones contractuales claras si varias empresas se involucran a un tiempo.

En la mayoría de los proyectos de software libre, la palabra clave es: motivación. Los desarrolladores no pueden ser "gestionados" en el sentido clásico de la palabra. Por contra, trabajan juntos, con procesos de toma de decisiones y planificación normalmente basados en el concepto de meritocracia y confianza mutua. Por supuesto esto no significa que no

haya toma de decisiones, o que las opiniones y propuestas de todos los participantes se tengan en consideración en la misma medida. Los proyectos de software libre suelen seguir ciertas reglas, que en algunos casos existen por escrito (al menos parcialmente), pero pueden ser muy diferentes de aquellas que se utilizan en el mundo empresarial.

Este nuevo entorno suele ser difícil de entender para algunos desarrolladores de empresas, que puede ser que tengan muchos años de experiencia, pero que pueden ser desconocedores de las particularidades de los proyectos de software libre, como por ejemplo las posibilidades de que una contribución sea aceptada o no o cómo hacerla de la mejor manera para que lo sea, que es una de las cosas que puede ser difícil de entender para ellos.

Cuando una empresa adopta un producto de software libre o decide lanzar el suyo propio, este tipo de problemas han de ser tenidos en cuenta. De cualquier modo, hay diferentes estrategias para lidiar con ellos, desde contratar nuevo personal con experiencia en entornos de software libre hasta proporcionar formación al equipo existente, para que aprendan y experimenten el tipo de situaciones que se van a encontrar.

## 2.4. Impulsar un producto

Cuando una empresa lanza un nuevo producto de software libre, normalmente está interesada en mantener algún tipo de control sobre él y su evolución futura. En el caso del software propietario, este control se ve garantizado por los estrictos términos de las licencias. Pero si se trata de software libre, otros pueden trabajar en él también, mejorarlo (puede que de una manera en que a la empresa no le guste) y redistribuir nuevas versiones, proporcionar servicios basados en él, y puede que, con el tiempo, hacerse con el control del producto (siendo por ejemplo que distribuya la versión más conocida y usada).

Para evitar esto, las empresas pueden utilizar diferentes estrategias, pero no dejan de ser estrategias en un nuevo entorno para ellas. De hecho, están basadas en mantener su estatus de productor principal del producto, que normalmente depende de cuánto esfuerzo y recursos están dispuestas a dedicar al producto una vez ha sido distribuido. Es por esto que han de planear de antemano si no quieren arriesgarse a perder su posición privilegiada. La situación es más problemática cuando la empresa no es el único productor, sino que se ha unido a una comunidad de desarrollo.

Normalmente, son varias las compañías que compiten en ella por ser consideradas "cabeza del proyecto" Buscar situaciones sinérgicas que involucren a otros actores y entender como contribuir a la creación de una comunidad sana que asegure la viabilidad del proyecto al mismo tiempo que se mantenga un cierto grado de control sobre ciertas decisiones, es un juego difícil.

## 3. Nuevas estrategias para nuevas situaciones

El software libre es un mundo nuevo para las compañías. Es territorio desconocido no sólo para los que producen software como su principal línea de actividad, sino también para aquellos que necesitan software para sus productos o servicios. Esta última categoría, engloba a una enorme porción de todas las compañías, puesto que son estratégicamente más y más dependientes del software.

Por tanto, se necesitan nuevas estrategias y muchas compañías las están explorando activamente para beneficiarse del software libre, para usar software libre como medio para reforzar otros fines, o incluso como parte integral de la política de negocio de la compañía. En esta sección se muestran algunas de estas estrategias, las cuales en muchos casos vienen a describir nuevas maneras de colaborar con otros actores.

### 3.1. Comunidades de empresas

Las comunidades de empresas basadas en el software libre son una nueva forma de colaboración empresarial no basada en relaciones contractuales. El núcleo alrededor del cual se forma la comunidad suele ser un conjunto de proyectos de software libre que pueden estar interrelacionados o no. Las empresas participan en la comunidad porque tienen algún tipo de interés en el software (producirlo, usarlo, proveer servicios alrededor de él, etc.), o incluso en la comunidad misma, por las oportunidades de negocio que pueden surgir de ella. De alguna manera, estas comunidades están modeladas según las comunidades tradicionales de software libre pero con la diferencia principal de que están compuestas por compañías en lugar de por individuos. No suelen estar sujetas a alianzas institucionales formales, aunque en algunos casos existe el concepto explícito de membresía. Los mismos principios de meritocracia y "los que más contribuyen son los que deciden" que se encuentran en otras comunidades de software libre, también se ponen en funcionamiento aquí, hasta cierto punto.

Estas comunidades de empresas son en parte similares a los grupos de interés informales pero con un lazo especial que armoniza los diferentes intereses: el código que se produce en la comunidad. En este marco, las empresas interesadas en obtener un producto concreto simplemente empiezan a desarrollarlo, puede que tras una colaboración pactada con otros actores. Las acciones de negocio pueden ser coordinadas por varios miembros y en algunos casos se encuentra cierto grado de aprovisionamiento común de fondos y recursos para lanzar nuevas ideas. Normalmente se trata de una mezcla de empresas grandes y pymes, junto con organismos públicos de investigación y otros entes. Esta peculiar mezcla facilita la transferencia tecnológica desde el mundo académico a la industria y la creación de ecosistemas ricos alrededor de proyectos de software libre específicos.

El hecho de que el software producido sea software libre garantiza que nadie tiene control absoluto sobre él y que se busquen soluciones con las que todos los diferentes actores se sientan a gusto. Esto de alguna manera allana el camino y ayuda a establecer algunas normas, lo que facilita el hecho de que compañías competidoras puedan, en algunas circunstancias, colaborar en proyectos específicos de su mutuo interés sin el engorro y la complejidad de detallados acuerdos legales. Algunos ejemplos de estas comunidades son OW2<sup>3</sup> (antiguamente Object Web), construida alrededor de tecnologías web, o Morfeo<sup>4</sup> que trata con diversas tecnologías.

### 3.2. Alianzas empresariales

Las alianzas son un caso específico de grupo de empresas con un objetivo específico y común a todos los participantes, que actúa como lazo de unión que las mantiene unidas. Este objetivo común es normalmente la producción o promoción de una tecnología específica o un estándar basado en el software libre. Las empresas participantes normalmente firman un acuerdo común e invierten recursos en ese fin. En general, no son muy diferentes de otros tipos de alianzas empresariales y sus principales diferencias son debidas al hecho de estar construidas alrededor del software libre. De hecho, esto suele ayudar a que la alianza se establezca como un punto neutral de control, abierto a nuevos actores. Ya que nadie controla la tecnología (puesto que es software libre), en principio el campo está allanado para que todos los participantes compitan según sus capacidades.

Algunos ejemplos de alianzas basadas en software libre son Limo Foundation<sup>5</sup>, Open Handset Alliance<sup>6</sup>, y Symbian Foundation<sup>7</sup> (todas en el área de los sistemas operativos para dispositivos móviles), o Genivi Alliance<sup>8</sup> (dedicada a la producción de una plataforma de software para coches).

### 3.3. Promoción de las comunidades tradicionales

Otra estrategia utilizada habitualmente por las empresas es la creación y promoción de una comunidad de desarrolladores y usuarios, diseñada según las comunidades de software libre tradicionales. En este caso, una compañía que lidera un proyecto de software libre intenta que otras compañías e individuos se interesen, usando una estructura basada en la meritocracia, la contribución y la confianza, de manera similar a lo que ocurre en las comunidades formadas por voluntarios.

La empresa que promueve la comunidad se posiciona en una figura de promotor benevolente, de algún modo renunciando a algunos de los beneficios de ser el principal productor. El hecho de que el producto sea software libre también proporciona ciertas garantías a otros actores de que el promotor no mantiene control total, lo que hace que sea más interesante para ellos el participar. Por regla general, se encuentran relaciones sinérgicas donde

los promotores consiguen esfuerzo e innovación invertido en el producto en cantidades substanciales, mientras que los actores se benefician bien de los recursos del promotor, bien de las oportunidades de negocio que se dan gracias a la escala de la colaboración.

Algunas empresas productoras de distribuciones de Linux, como RedHat<sup>9</sup> con su comunidad Fedora<sup>10</sup> o Canonical<sup>11</sup> con su comunidad Ubuntu<sup>12</sup> fueron de las primeras en explorar esta estrategia. Más recientemente, Nokia<sup>13</sup> ha sido un caso sonado también, con su comunidad Maemo<sup>14</sup>. Como estos casos ha habido muchos otros recientemente.

### 3.4. Involucrarse en comunidades tradicionales

En algunos casos, la mejor estrategia percibida por una empresa es unirse a una comunidad tradicional de desarrollo de software libre existente. En ocasiones, estas comunidades comenzaron como un grupo de individuos, probablemente voluntarios, con el objetivo común de compartir esfuerzos para el desarrollo y el mantenimiento de algún producto de software específico. Pero con el tiempo, es habitual que desarrolladores contratados por empresas interesadas en el producto acaben por unirse a la comunidad o que las empresas contraten directamente a desarrolladores de la comunidad, para de algún modo unirse al desarrollo colaborativo (y en algunos casos tratar de influenciarlo). Este ha sido el caso, por ejemplo, de una de las más conocidas comunidades de software libre: Apache<sup>15</sup>. Desde sus comienzos, varios desarrolladores clave han estado trabajando para compañías mientras participaban en Apache en tiempo de trabajo. La comunidad Apache reconoció este hecho pero trató de mantenerse como una comunidad de individuos en la cual los miembros no se consideran representantes de la empresa para la que trabajan. Aunque siempre puede haber algún tipo de tensión debido a intereses empresariales, de cuando en cuando.

En algún momento, algunas comunidades formalizan la participación de empresas. La comunidad acepta recibir donaciones (en forma de empleados, fondos u otros valores como infraestructuras, servidores, etc.) y se estipulan reglas para equilibrar los intereses de las empresas junto con los del proyecto como un todo.

Normalmente, las empresas tratan de impulsar el proyecto ya no requiriendo más influencia a cambio de sus donaciones, sino produciendo código para las funcionalidades específicas que están buscando. En algunos casos, esto lleva a la creación de nuevos productos dentro del proyecto. Esto puede distorsionar la relación entre voluntarios y empleados en el proyecto. Pero también produce muchos beneficios, uno de los cuales puede ser el incremento de los recursos humanos que trabajan activamente en el proyecto o la solución a problemas específicos en los que los

voluntarios podrían no estar tan interesados. Existen varios ejemplos de casos como este. El Proyecto GNOME<sup>16</sup> es probablemente el caso más conocido. A finales de los 90 y comienzos de esta década, muchas empresas (grandes corporaciones como Sun<sup>17</sup> o Novell<sup>18</sup>, o pequeñas *startups* como Eazel<sup>19</sup> o Ximian<sup>20</sup>) colaboraron juntas. Un caso más moderno puede ser la relación de la Mozilla Foundation<sup>21</sup> con varias compañías, especialmente Google<sup>22</sup>, de la que recibe una cantidad considerable de ingresos.

#### 4. El papel de las pymes

Algunas pequeñas y medianas empresas tienden a ser muy activas en términos de innovación y creación de nuevas tecnologías. Normalmente son consideradas más ágiles y dinámicas que las grandes corporaciones. Es por esto que no es sorprendente que muchas de ellas hayan sido pioneras en el campo del software libre.

Algunos de los primeros casos se dieron gracias a desarrolladores de software libre que fundaban sus propias pymes para rentabilizar el tiempo y esfuerzo invertido en la creación de producto de software libre exitoso. En estos casos, el desarrollo fue comenzado por una comunidad de voluntarios, pero en algún momento, una parte o todos ellos, decidieron crear una empresa que probablemente se encargaba de todo o casi todo el desarrollo y ofrecía servicios con ánimo de obtener beneficios. Un ejemplo muy representativo de esto es el caso de MySQL<sup>23</sup>, donde los principales desarrolladores fundaron MySQL AB, la empresa que lideró a partir de entonces el desarrollo casi al completo. Al mismo tiempo, ofrecieron servicios con los que obtenían fondos para financiar el desarrollo.

Este modelo se encuentra en muchos otros casos antes y después del de MySQL. De hecho, es tan popular que en muchos casos, la empresa se establece incluso antes de que el desarrollo comience formalmente, con el objetivo formal de dirigirlo cuando empiece. De algún modo, Red Hat o Ximian siguieron este modelo, siendo sólo dos de los muchos ejemplos existentes. Con el tiempo, estas compañías pueden llegar a ser compradas por grandes corporaciones (como ha sido el caso de MySQL AB, adquirida por SUN o Ximian, adquirida por Novell), que de esta manera, incorporan innovación en su estrategia corporativa.

En algunas pymes innovadoras los desarrolladores de software libre encuentran trabajo con más facilidad. De hecho, para estas compañías, contratar a estos desarrolladores es una manera de mostrar su implicación en los productos y servicios que ofrecen (cuando están basados en software libre), y una manera de diferenciarse de la competencia. La relación entre pymes y voluntarios en algunos proyectos es tan cercana que los desarrolladores pasan de vez en cuando de una pyme a otra, sin dejar de trabajar en el mismo proyecto y viceversa: cambian de proyecto sin llegar a tener que cambiar de

empresa. Desde muchos puntos de vista, muchos proyectos serían simplemente inviables sin la participación de las pymes y los recursos que invierten en el desarrollo, mientras que al mismo tiempo estas pymes no serían viables sin el proyecto exitoso de software libre con el que trabajan.

La relación de las pymes con las grandes corporaciones en las comunidades de desarrollo no es siempre fácil. Las pymes tienen problemas para crear ecosistemas ricos por sí mismas, lo que significa que normalmente agradecen la adhesión de grandes empresas. Pero una vez que entran en juego, puede darse una situación de desequilibrio para los actores más pequeños. Aunque por norma suelen ser más activos y dinámicos, el mero volumen de grandes actores y la disponibilidad de sus recursos puede dirigir el proyecto en direcciones inesperadas. En cualquier caso, cuando estas capacidades diferentes se utilizan con el ánimo de buscar sinergias se puede conseguir un enorme impulso para el proyecto y beneficio para todos los actores.

#### 5. El software libre y los profesionales informáticos

Los profesionales de las TIC se sitúan en el corazón del desarrollo de software libre. A pesar de lo que todavía digan fuentes desinformadas, la mayoría de proyectos de software libre son liderados por profesionales experimentados en el desarrollo de software, normalmente con muchos años de experiencia. En algunos casos, han estado involucrados en el software libre desde siempre y en otros vienen de la industria del software tradicional. Puede que empezaran trabajando en proyectos de software libre como *hobby* en su tiempo libre o puede que lo hicieran debido a una decisión de su compañía. Pero, en cualquier caso, ellos como individuos son el pegamento que mantiene unidos los elementos de los proyectos.

Como hemos mencionado repetidamente en este texto, en muchos casos las empresas están involucradas en proyectos, pero no son "ciudadanos de primera" en ellos. La mayoría de las comunidades de software libre están basadas en las relaciones entre personas, no en acuerdos corporativos. Es por esto que normalmente, el papel de los profesionales informáticos que participan en estas comunidades sea mucho más importante que el de las empresas para las que trabajan. De algún modo, la mayoría de los proyectos de software libre están en realidad liderados por profesionales de las TIC que son reconocidos en la medida en que merecen por ello y se esfuerzan en el proyecto.

De hecho, este reconocimiento del mérito y el esfuerzo nos lleva a un nuevo papel para los desarrolladores profesionales. En lugar de estar restringidos por acuerdos privados y reglas empresariales que les impiden mostrar a sus colegas sus capacidades y logros, el juego en el mundo del software libre es exac-

tamente el contrario. Todo el trabajo es público, y todas las acciones son trazables. Los profesionales que hacen un buen trabajo son reconocidos por sus colegas y se llevan su reputación consigo según se mueven de compañía en compañía.

Desde el punto de vista del desarrollo profesional a largo plazo, tan importante para los profesionales de las TIC, el software libre es un mundo de posibilidades. Al poder leer el código, entender nuevas arquitecturas de programas libres, nuevas tecnologías y nuevos paradigmas resultan tareas más sencillas. En conclusión, el software libre brinda nuevas oportunidades para profesionales. Por supuesto que existen algunos riesgos evidentes, como los problemas derivados de exponer públicamente prácticas de desarrollo deficientes, por ejemplo. Pero en general, los profesionales tienen mucho más de lo que beneficiarse cuando trabajan con software libre.

#### 6. Conclusión

El software libre es un nuevo campo en el panorama de las TIC, no porque sea una nueva tecnología sino porque brinda nuevos procesos y posibilidades. Tanto las empresas como los profesionales pueden obtener beneficios si encuentran la manera de aprovechar sus capacidades y pueden alinear sus objetivos con los de la comunidad del software libre. De todos modos, como todo nuevo mundo, es crucial entender las nuevas reglas y explorar las oportunidades emergentes para beneficiarse de él. Por supuesto también existen riesgos, especialmente para aquellos que no se den cuenta de que, en muchos aspectos, el entorno ha cambiado.

#### Notas

- <sup>1</sup> Free Software Foundation <<http://www.fsf.org/>>.
- <sup>2</sup> Open Source Initiative <<http://www.opensource.org/>>.
- <sup>3</sup> OW2 Consortium <<http://www.ow2.org/>>.
- <sup>4</sup> Morfeo Project <<http://morfeo-project.org/>>.
- <sup>5</sup> LiMo Foundation <<http://www.limofoundation.org/>>.
- <sup>6</sup> open handset alliance <<http://www.openhandsetalliance.com/>>.
- <sup>7</sup> Symbian Foundation <<http://www.symbian.org/about/index.php>>.
- <sup>8</sup> GENIVI Alliance <<http://www.genivi.org/>>.
- <sup>9</sup> redhat <<http://www.redhat.com/>>.
- <sup>10</sup> fedora <<http://fedoraproject.org/>>.
- <sup>11</sup> CANONICAL <<http://www.canonical.com/>>.
- <sup>12</sup> ubuntu <<http://www.ubuntu.com/>>.
- <sup>13</sup> NOKIA <<http://www.nokia.com/>>.
- <sup>14</sup> maemo.org <<http://maemo.org/>>.
- <sup>15</sup> The Apache Software Foundation <<http://www.apache.org/>>.
- <sup>16</sup> GNOME: The Free Software Desktop Project <<http://www.gnome.org/>>.
- <sup>17</sup> Sun microsystems <<http://www.sun.com/>>.
- <sup>18</sup> Novell <<http://www.novell.com/>>.
- <sup>19</sup> <<http://en.wikipedia.org/wiki/Eazel>>.
- <sup>20</sup> <<http://en.wikipedia.org/wiki/Ximian>>.
- <sup>21</sup> mozilla.org <<http://www.mozilla.org/>>.
- <sup>22</sup> Google <<http://www.google.com/intl/en/corporate/>>.
- <sup>23</sup> MySQL <<http://www.mysql.com/>>.

Carlo Daffara  
Director de I+D de Conecta Research Ltd.,  
Italia

<cdaffara@conecta.it>

## Buenas prácticas para la adopción del software libre

**Traducción:** Juan Jose Amor Iglesias (GSyC/LibreSoft, Universidad Rey Juan Carlos)

### 1. Introducción

En una empresa, la decisión de adoptar soluciones basadas en software libre (*Free/Libre Open Source Software*, FLOSS), o la de integrar el software libre en sus servicios, se basa normalmente en los siguientes casos de uso:

- Migración/Sustitución básica: consiste en usar el FLOSS en la infraestructura informática, sustituyendo normalmente instalaciones basadas en software privativo.
- Nuevo desarrollo: introducción del FLOSS para desarrollar un nuevo proyecto de la compañía (adopción).
- Venta de servicios basados en FLOSS.
- Venta de productos que contienen FLOSS internamente en una cantidad significativa.

En este sentido, una empresa puede encontrar útil el software libre desde un punto de vista táctico (cuesta menos dinero implementarlo, con menos dependencia de proveedores, o puede ayudar a introducir productos en el mercado en menos tiempo). Los procesos de adopción interna se suelen modelizar usando un modelo incremental, desarrollado inicialmente por P. Carbone [1] y otros. Este modelo es un proceso de pasos sucesivos, siendo el primero el más complejo de todos, normalmente (ver **figura 1**).

En todo caso, el proceso de migración y adopción es complejo, con esfuerzos multidisciplinarios que afectan a varias áreas y requieren una comprensión completa de los flujos de trabajo individuales, cómo se ejecutan y cómo la gente interactúa con los sistemas de información en su trabajo diario. En este sentido, una migración a software libre es un reto importante y, como muchas otras tareas complejas, puede ir mal. Hay muchos obstáculos que nos encontramos al enfrentarnos a una migración, pero algunos pueden evitarse fácilmente siguiendo prácticas simples. Muchas de las dificultades no son de naturaleza técnica, sino organizativas, y casi todo el esfuerzo que requerirán vendrá de arriba (del equipo de gestión de la empresa); otro aspecto importante es el impacto social de la migración (tal como la aceptación de los usuarios de los nuevos sistemas), lo cual requerirá atención aparte.

### 2. Pautas de gestión

El camino a seguir para hacer una migración correcta al uso de FLOSS siempre empieza por tener un panorama preciso de las funcio-

**Resumen:** muchas empresas y administraciones públicas se plantean la adopción del software libre (*Free/Libre and Open Source Software*, FLOSS) en sus proyectos, o simplemente la migración a este, puesto que les puede implicar menores costes de implementación de nuevos sistemas informáticos o de mantenimiento de los actuales. Pero, en general, los procesos de adopción o migración son complejos: se requieren esfuerzos multidisciplinarios que cubran diversas áreas, una completa comprensión de cómo son los flujos de trabajo internos en la empresa y de cómo interactúan los usuarios con los sistemas informáticos. Este artículo propone una serie de pautas con el fin de maximizar el éxito de una adopción o migración a software libre en organizaciones, pautas que se dividirán en tres grupos: pautas de gestión (orientadas a la gestión de mayor nivel en la compañía), pautas técnicas (orientadas al punto de vista técnico) y pautas sociales (orientadas a características típicas del FLOSS que las compañías querrán aprovechar, como la relación con las comunidades de desarrollo o la mejora del soporte técnico gracias a la experimentación y formación internas).

**Palabras clave:** adopción de software libre, comunidades de software libre, empresas, migración, software libre.

#### Autor

**Carlo Daffara** es el miembro italiano del Grupo de Trabajo Europeo sobre software libre. Preside además diversos grupos de trabajo como el grupo de *middleware* de código abierto del comité técnico en computación escalable del IEEE y el grupo de trabajo de las PYME del Competitiveness Task Force de la UE. Su actividad investigadora actual se centra en la sostenibilidad de los modelos de negocio basados en el software libre. Actualmente es jefe de I+D en Conecta Research Ltd, empresa de consultoría en software libre.

nes de Tecnologías de la Información (TI) de la empresa, una clara visión de las necesidades y beneficios de las transiciones y un soporte permanente. Las diferencias con los modelos de desarrollo y soporte en el software libre pueden requerir un cambio significativo en la manera en la que el software y los servicios son gestionados y, en general, un desplazamiento de la responsabilidad desde los proveedores externos hacia el personal interno.

#### Asegúrese de gestionar el compromiso a la transición

El soporte de la gestión y el compromiso se muestran repetidamente como las variables más influyentes en el éxito de los esfuerzos más complejos de gestión de TI, y las migraciones a software libre no son una excepción. Este compromiso debe mantenerse durante un período de tiempo suficiente para cubrir la migración completa. Esto significa que, en las organizaciones cuyos directores de Informática son cambiados con frecuencia, o donde la gestión cambia con cierta periodicidad (por ejemplo, en las Administraciones públicas donde ocurre a menudo), debe haber un proceso para transferir la gestión de la migración. El compromiso debe también extenderse a la financiación (puesto que tanto los cambios como la formación requerirán re-

ursos, tanto económicos como humanos). La mejor forma de garantizar una coordinación constante es asignarle un equipo con experiencia mixta (gestión y técnica) para dar una realimentación continua y una gestión del día a día.

*A tener en cuenta:* si las únicas personas que trabajan en la migración son personal del departamento de informática, probablemente no tendrán información suficiente sobre la gestión de más alto nivel, ni planes de financiación para continuar la migración tras los primeros pasos.

#### Prepare una panorámica clara de lo que se espera de la migración o adopción, incluyendo puntos medibles

La transición puede empezarse por varios motivos, incluyendo el mejor control de los costes de informática, independencia de los proveedores, flexibilidad o soporte de los estándares abiertos. Para asegurarse de que la migración está produciendo beneficios de forma efectiva, o está yendo de acuerdo al plan de migración, es fundamental conocer de antemano los indicadores que servirán para evaluar el progreso. Los requisitos deben ser realistas, y, en particular, las expectativas de

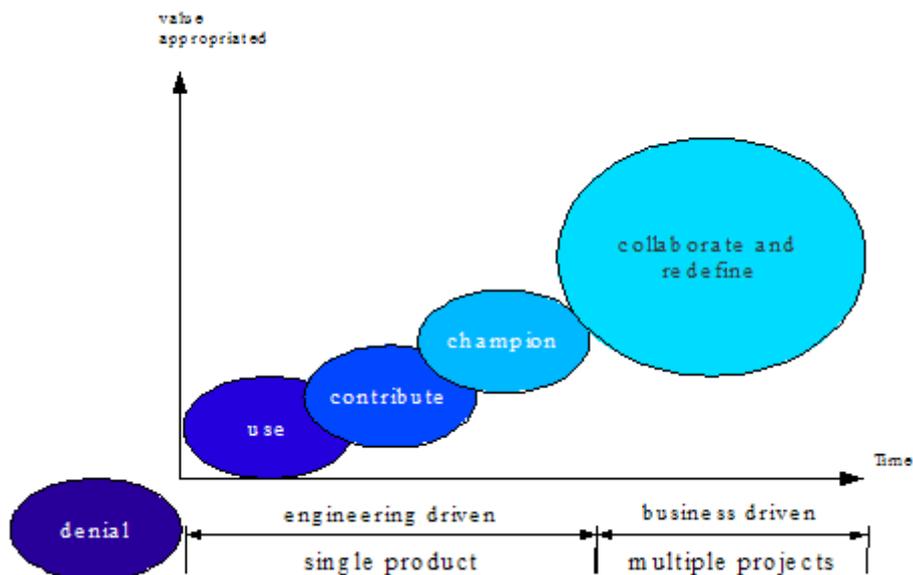


Figura 1. Modelo de pasos sucesivos en la adopción de software libre.

reducción del coste total de la propiedad (TCO) deben poderse contrastar con datos públicamente disponibles.

*A tener en cuenta:* si la única ventaja que se percibe es que "el software se baja gratis de la red", puede que tengamos una serie de premisas incorrectas que probablemente desembocarán en una opinión negativa sobre la migración.

**Asegúrese de que el calendario es realista**

La introducción de una nueva plataforma informática requerirá siempre mucho tiempo; por regla general, el tiempo para realizar una transición completa hacia el FLOSS puede considerarse comparable al de la introducción de un nuevo software ERP (Enterprise Resource Planning) en la compañía; para transiciones más pequeñas, el esfuerzo en tiempo debe escalar proporcionalmente.

*A tener en cuenta:* cuando el tiempo de migración se mide en días, y no se planifica esfuerzo para después de la migración, el proceso puede quedarse detenido forzosamente si los recursos planificados se agotan.

**Revise los procedimientos actuales de obtención y desarrollo de software o tecnología informática**

Puesto que los esfuerzos de implementación se desplazan desde el software comercial hacia el libre, los procesos de obtención y desarrollo deben ser actualizados convenientemente. En particular, la mentalidad de "compra" puede que deba cambiarse hacia una mentalidad de "servicio", puesto que se com-

parará menos software "precintado" y este cambio puede requerir diferencias en el reparto del presupuesto interno para informática. El software desarrollado internamente requerirá ser *portado* o adaptado a un nuevo software que será multi-plataforma o accesible usando interfaces estándar (por ejemplo, aplicaciones web), y esto debe tenerse en cuenta en el plan informático global.

*A tener en cuenta:* Cuando no se planifican cambios en el proceso de obtención y desarrollo, es porque no se ha comprendido el alcance de los cambios requeridos para la adopción de software libre en la empresa.

**Busque consejo o información de experiencias parecidas**

Dado que el número de empresas y administraciones que han hecho una migración ya es considerable, es fácil encontrar información sobre lo que hacer. En este sentido, el proyecto COSPA ha desarrollado una base de conocimiento que es accesible a través de su sitio web principal [2]; las Administraciones públicas pueden también contactar con el centro de competencias en software libre que tengan más a mano, para que les proporcione información y apoyo durante el proceso de migración.

**Evite una transición "de golpe" e intente que las migraciones sean incrementales**

Casi todas las migraciones a gran escala han sido realizadas de una vez<sup>1</sup> (implicando un cambio abrupto entre un entorno informático y el nuevo) y suelen venir afectadas por costes técnicos y de soporte extremadamente altos.

Aunque la necesidad de apoyo técnico en más de un entorno a la vez incrementa tanto los costes de soporte como de gestión, las migraciones incrementales o "suaves" suelen traer una mejor experiencia a los usuarios, así como un trastorno mínimo de los procesos de negocio.

Un ejemplo de migración suave sería aquella que comienza por la migración de las aplicaciones del lado del servidor, que normalmente siguen estándares o simplemente basadas en conceptos de red (cliente/servidor) que las hace fáciles de reemplazar, dejando las aplicaciones de escritorio o de interfaz de usuario para el final. Un esquema así se mostró en [3] (Ver figuras 2 y 3).

**Asigne al menos una persona que interactúe con la comunidad de desarrollo e intente encontrar buenas fuentes de información en la red**

Una ventaja importante del software libre es la disponibilidad en la Red de recursos gratuitos, en forma de bases de conocimiento, listas de correo, wikis (sitios cooperativos) que pueden proporcionar una cantidad de apoyo técnico que, en muchos casos, es comparable a lo que ofrece un servicio comercial.

El gran problema es la localización de esas fuentes de conocimiento; en este sentido, asignar recursos para encontrar, clasificar e interactuar con esas fuentes de información es una forma de reducir el coste del soporte; una manera de ofrecer de forma unificada estas fuentes de información es mediante una sencilla página en la Intranet con enlaces a recursos públicos en línea.

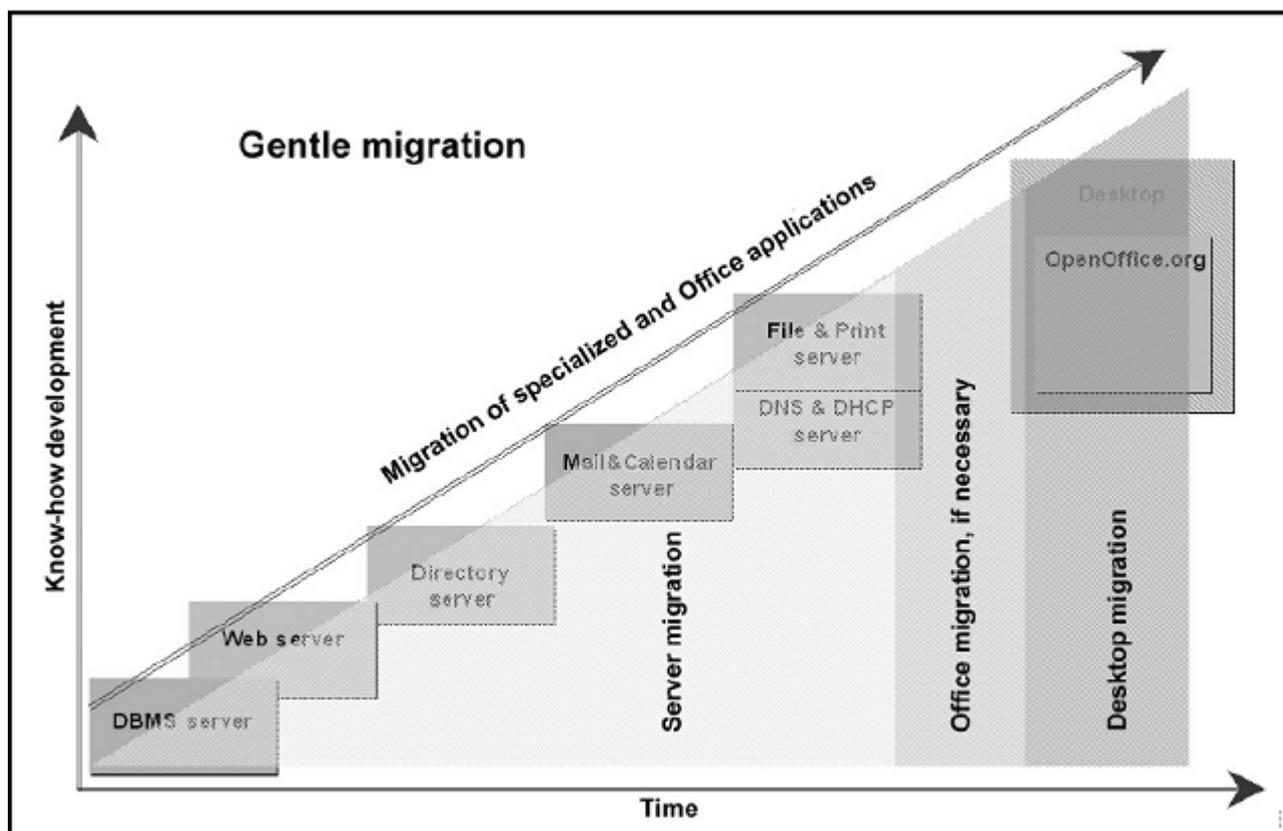


Figura 2. Esquema de migración "suave"comenzando por las aplicaciones del servidor.

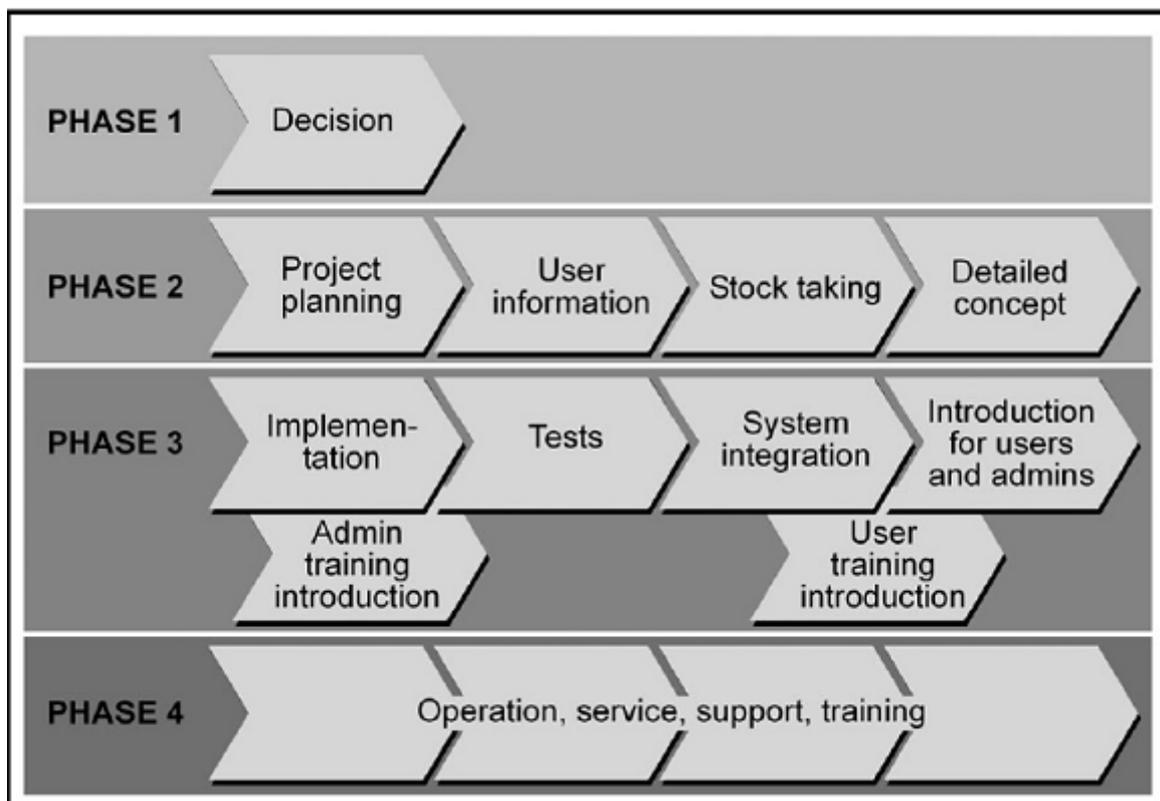


Figura 3. Esquema de fases de una migración "suave".

*A tener en cuenta:* vigilar si la gente no sabe cómo encontrar información sobre las herramientas que utilizan, o cuándo todos tienen que buscar en los sitios web para encontrar ayuda.

### 3. Pautas técnicas

Una diferencia que caracteriza las adopciones de FLOSS es el modelo de desarrollo adoptado por casi todos los proyectos de software libre, que afecta especialmente a la frecuencia de entregas de actualizaciones o soporte. Esto requiere un cambio en la manera de adoptar el FLOSS y gestionar las actualizaciones, para reducir en lo posible los problemas de interoperabilidad.

#### Entienda la manera de desarrollar el software libre

Casi todos los proyectos siguen un modelo de desarrollo cooperativo, con un grupo central de programadores haciendo casi todo el código (normalmente desde una empresa) y un gran número de cooperadores adicionales. Este modelo proporciona una gran calidad en el código y rápidos ciclos de desarrollo, pero también requiere un esfuerzo importante en el seguimiento de los cambios y las actualizaciones. La adopción de un paquete de software libre debe considerarse cuando:

- El proyecto en sí mismo está "vivo", es decir, tiene alrededor una comunidad de desarrollo activa.
- Hay una clara distinción entre software "estable" e "inestable". En muchos proyectos, hay dos líneas de desarrollo: una, centrada en integrar los últimos cambios e implementaciones (la "inestable"), y otra, centrada en mejorar la estabilidad y corregir los fallos (la "estable"); periódicamente, los programadores "congelarán" el desarrollo para convertir una versión "inestable" en otra "estable", creando aparte una nueva rama de desarrollo con las últimas novedades implementadas. Esta distinción permite a los programadores satisfacer tanto a los usuarios que quieren probar las últimas funcionalidades, como a aquellos que usan el software en sistemas de producción, pero requiere un esfuerzo extra para recolectar la información sobre las nuevas versiones.

Si las nuevas funcionalidades o correcciones son necesarias, puede ser más simple pedir una versión con soporte comercial; en muchos casos, el proveedor comercial será también un cooperador económico en el proyecto de software libre que vende.

*A tener en cuenta:* precaución cuando el gestor de TI o los desarrolladores creen que el software libre viene a ser como un software comercial que alguien ha puesto gratis en la red y simplemente "funciona".

#### Prepare una lista completa de software y hardware que serán afecta-

#### dos por la migración y qué funcionalidad está buscando la compañía

Cuando la situación inicial es poco conocida, la migración puede ser un fracaso. Muchas empresas y Administraciones no tienen implementadas auditorías de las plataformas de software y hardware, y no están capacitadas para cuantificar las herramientas y el software que deben ser reemplazados o integrados en una migración a software libre. El proceso de elaboración de la lista debe tener en cuenta también el número de usuarios simultáneos, el uso medio en la organización y si las aplicaciones utilizan protocolos de comunicación o formatos de intercambio abiertos o cerrados.

Esta encuesta será la base de la decisión sobre a qué usuarios afectará la migración en primer lugar, y para tener en cuenta el coste de la adaptación del software o la migración a un formato de datos diferente. Las herramientas de inventario automático de software están ahí y pueden reducir el coste de realizar el inventario y permitir un control más estricto sobre el software instalado (reduciendo el coste de mantenimiento).

Algunos aspectos a verificar:

- El formato de datos usado, a nivel de intercambio de documentos pero también a nivel de base de datos y protocolos de comunicación de red.
- La lista de aplicaciones en uso, incluyendo aquellas desarrolladas internamente, macros y documentos.
- La funcionalidad disponible.
- Las deficiencias o los problemas de la infraestructura actual.

Es fundamental que el software migrado cubra los mismos requisitos funcionales de la infraestructura informática actual, y es preferible que además mejore la misma en términos de funcionalidad o calidad (disponibilidad, fiabilidad, rendimiento...).

#### Utilice la flexibilidad del software libre para hacer adaptaciones locales

Lo que hace diferente al software libre es la flexibilidad y la libertad que da a los usuarios y programadores para crear nuevas versiones o adaptaciones de cualquier paquete. Esta flexibilidad puede aumentar mucho el valor que se percibe del software libre, por ejemplo es posible crear paquetes personalizados que contienen configuraciones locales, tipos de letra específicos u otro material de interés como macros prediseñadas o plantillas que se usen habitualmente en la empresa. Además, la personalización del aspecto visual puede aumentar la aceptación de los usuarios, al presentarles un escritorio más atractivo que mantenga a mano los enlaces o elementos de menú más habituales.

Esta personalización puede integrarse de forma simple en casi cualquier distribución de Linux, o bien creando un repositorio de software local. Obsérvese que, en casi todos los casos, no hay que escribir programas, puesto que la personalización consiste en seleccionar el paquete adecuado, cambiar el aspecto visual o proporcionar plantillas de documentos.

#### Hay mucho más software disponible que el que se instala por defecto

Temas de licencia o el propio diseño son los que limitan sustancialmente la cantidad de software que normalmente se incluye en las instalaciones predefinidas de las distribuciones Linux habituales. Por ejemplo, solo unas pocas incluyen capacidad para reproducir los formatos de vídeo o audio más populares, debido a restricciones de licencia o patente; también es habitual que algunos paquetes que nos parezcan interesantes no estén incluidos porque solo los usan una minoría de los usuarios.

Por esta razón, es importante investigar e incluir en las instalaciones predefinidas aquellos paquetes adicionales que puedan ayudar en el periodo de transición, como paquetes que incluyan tipos de letra adicionales, herramientas multimedia y otros paquetes que puedan ser útiles en un entorno mixto.

Al seleccionar los paquetes, dé preferencia siempre a la estabilidad sobre la funcionalidad. Entre los muchos paquetes potencialmente útiles para cada función, siempre los hay con más funcionalidades o con más estabilidad. En general, entre los paquetes candidatos deben elegirse los más estables, o aquellos que sean más populares (pues esto generalmente hace que haya más información de soporte disponible), y también aquellos con menor variación entre versiones.

*A tener en cuenta:* Cuidado si el administrador de TI siempre quiere poner lo último en los escritorios de los usuarios.

#### Diseñe la infraestructura de soporte para reducir el número de incompatibilidades

Cada transición de una infraestructura informática a otra siempre conlleva pequeñas diferencias e incompatibilidades; esto puede verse por ejemplo al convertir documentos de un formato a otro.

La infraestructura general debe reducir el número de estos pequeños problemas, por ejemplo rediseñando las plantillas de documento en el formato abierto ODT (*OpenDocument Text*) en lugar de usar versiones previas hechas con herramientas privadas. Esto reduce en gran parte las diferencias de estilo y formato que se producen al convertir un documento de un formato a otro.

## Introduzca un sistema de gestión de incidencias<sup>2</sup>

Una dificultad en cada nuevo despliegue tecnológico es cómo asegurarse la satisfacción de los usuarios y el grado de aceptación de la nueva solución, especialmente en compañías de tamaño medio donde es más difícil obtener realimentación de los usuarios. Un sistema de gestión de incidencias puede ofrecer una forma simple de recolectar los fallos del despliegue y ayudar a identificar si los usuarios necesitan capacitación adicional, por medio del análisis de las estadísticas de envío de incidencias por usuario. Además puede encontrar puntos débiles del despliegue, por ejemplo cuando se localizan muchas incidencias relacionadas con un área concreta.

## Recopile los detalles de la migración en un diario

Una migración a gran escala requiere una acción coordinada e información clara y actualizada. La mejor forma de recopilar esta información es manteniendo un "diario de migración", un lugar donde tener la documentación recolectada y preparada para la migración (incluyendo los objetivos, el plan detallado y la documentación técnica), y un diagrama de tiempos actualizado convenientemente de acuerdo con el avance del proyecto. Esto además simplifica la gestión del proyecto cuando hay un cambio en el equipo que dirige la migración.

## 4. Pautas sociales

### Proporcione información general sobre software libre

Un obstáculo habitual en la adopción de software libre es la aceptación de los usuarios, quienes suelen tener pocas nociones sobre software libre y estándares abiertos. Casi siempre, el software libre se percibe como peor, dado que es "gratis", al poderse obtener fácilmente de la red como muchos paquetes *shareware* o hechos por aficionados. Es importante acabar con esta percepción y proporcionar información sobre cómo se desarrolla el software libre y qué ideas generales y de negocio subyacen en él.

### No fuerce a los usuarios a cambiar, déles explicaciones

El cambio en la infraestructura tecnológica provocará un cambio importante en la forma de trabajar de los usuarios o en el uso de los recursos internos; este cambio puede causar resistencia en esos usuarios. Por ello debe simplificarse el cambio, explicando claramente el por qué y cómo tendrá lugar y qué beneficios se obtendrán a largo plazo, tanto internamente (menor coste, mayor flexibilidad y seguridad) como externamente (apertura, cumplimiento de estándares internacionales, menor carga a los usuarios externos).

*A tener en cuenta:* Hay que tener cuidado

cuando los usuarios creen que la migración simplemente se hace para ahorrar dinero en el software.

### Aproveche la migración como una manera de mejorar las habilidades del personal

Puesto que la nueva infraestructura requerirá capacitación, se puede aprovechar para mejorar las habilidades generales sobre TI. En muchas empresas y Administraciones públicas, se imparten pequeños cursos formales a los usuarios. Esto no solo incrementa la confianza, sino que armoniza habilidades en el grupo y en general mejora el desempeño.

Esto puede provocar cierta resistencia por parte de los "gurús de la empresa", que pueden interpretar esta formación como un mecanismo para destronarles como líderes técnicos. La mejor forma de evitar esta resistencia es localizar a esos usuarios y sugerirles que repasen el material de formación de mayor nivel (que puede estar disponible en un sitio web, por ejemplo).

Además, puede ser interesante localizar a los "campeones", es decir, entusiastas del software libre que pueden ayudar dando soporte a otros usuarios y ofrecerles formación adicional o un reconocimiento a nivel de gestión.

En general, es útil crear una Intranet interna con enlaces a los diferentes paquetes de formación.

### Facilite la experimentación y el aprendizaje

La libertad de licenciamiento, fundamental en el movimiento del software libre, permite la libre redistribución del software y los materiales de formación. En este sentido, entregar a los usuarios distribuciones en CD "live" (que no requieren instalación en el disco duro para probarlas) o material impreso que puedan llevarse a casa, puede aumentar la aceptación general del nuevo software.

### Agradecimientos

Este artículo se basa en material preparado para los proyectos COSPA, OpenTTT y FLOSSMETRICS financiados por la UE. Puede conseguirse más información en la guía FLOSSMETRICS para PyMES (FLOSSMETRICS guide for SMEs), <<http://guide.conecta.it>>.

## Referencias

- [1] P. Carbone. *Value Derived from Open Source is a Function of Maturity Levels*, OCRI conference "Alchemy of open source businesses", 2007. <[http://www.ocri.ca/email\\_broadcasts/041907partnership.html](http://www.ocri.ca/email_broadcasts/041907partnership.html)>.
- [2] EU COSPA Project. *D6.1 Report evaluating the costs/benefits of a transition towards ODS/OS*. <<http://www.cospa-project.org/>>.
- [3] KBSt, Alemania. *Migration guide*, 2006.

## Notas

<sup>1</sup> Este enfoque es conocido también como "enfoque Big Bang" o "adopción Big Bang" <[http://en.wikipedia.org/wiki/Product\\_Software\\_Adoption:\\_Big\\_Bang\\_Adoption](http://en.wikipedia.org/wiki/Product_Software_Adoption:_Big_Bang_Adoption)>.

<sup>2</sup> Un caso típico de estos sistemas son los llamados "Trouble Ticket Systems" (o también "Issue Tracking Systems") en los cuales se introducen las incidencias como "tickets", que son asignados a personas o equipos de trabajo que se encargan de resolverlos <[http://en.wikipedia.org/wiki/Issue\\_tracking\\_system](http://en.wikipedia.org/wiki/Issue_tracking_system)>.

Stéphane Ribas, Michel Cezon

Capítulo Local de OW2 Europe, Saint Ismier Cedex (Francia)

<{stephane.ribas,michel.cezon}@inria.fr>

# Construir y mantener una comunidad de práctica: método aplicado a proyectos de software libre

**Traducción:** Daniel Izquierdo Cortázar (GSyC/LibreSoft, Universidad Rey Juan Carlos)

## 1. Introducción

En el dominio de I+D, tecnologías como Web 2.0 han fomentado el desarrollo de nuevos modos de trabajo colaborativo, a menudo basados en relaciones entre pares (*peer-to-peer*) que finalmente evolucionan hacia comunidades más organizadas.

La creación de dichas comunidades ha mostrado su eficiencia y dinamismo en diversos dominios, más específicamente en el desarrollo de software libre. Y aún más, el incremento de las conexiones a nivel internacional ha generado una nueva necesidad de colaboración remota entre diferentes comunidades de expertos alrededor del mundo.

Desarrollado dentro del instituto INRIA, basado en la experiencia del Capítulo Local de OW2 Europa [1] y en la creación de la comunidad Aspire-RFID [2], este artículo describe una metodología para crear y mantener una comunidad de práctica.

## 2. ¿Qué es una comunidad?

Una comunidad puede ser considerada como un grupo de personas que comparten un interés común, las mismas inquietudes o la misma pasión por algo. Una comunidad puede estar también basada en roles o especialidades (por ejemplo administrador de redes, grupos de usuarios de Linux o el grupo de usuarios de un sintetizador modular). Estas personas mejoran y profundizan en sus conocimientos cuando los comparten, ya sea resolviendo problemas para otros, interactuando regularmente con los miembros de la comunidad, preguntando y resolviendo preguntas, o reutilizando buenas ideas [3].

Estos entusiastas participan activamente y se crea una nueva identidad virtual basada en un fuerte vínculo social que es de esperar produzca resultados colectivos [4].

## 3. Tipologías de las comunidades

Brevemente, podemos distinguir cuatro tipos de comunidades [3]: Comunidad de aprendizaje (por ej. el Proyecto Plume [5]), comunidad de interés (por ej. Audiofanzine [18]), comunidad de apasionados (por ej. administradores de sistemas participando en la Conferencia JRES 2009 [19]), y comunidad de práctica (por ej. Linux User Group [20]).

La comunidad de aprendizaje consiste en un

**Resumen:** *tecnologías como la Web 2.0 han fomentado el desarrollo de nuevas formas de colaboración en el dominio I+D, evolucionando a menudo hacia la creación de comunidades organizadas. El incremento de las conexiones a nivel internacional ha sacado a la luz la necesidad de colaboración remota alrededor del mundo, lo cual enfatiza la necesidad de organizar la creación y mantenimiento de dichas comunidades. Este artículo presenta un método, desarrollado por el Instituto INRIA, para construir y mantener una comunidad de práctica. Nuestro enfoque, basado en cinco pasos, ha sido evaluado en un proyecto europeo: el proyecto ASPIRE-RFID. Los resultados son aquí revisados, con la convicción de que la transparencia, la confianza y el compromiso de las personas son la clave para allanar el camino hacia el éxito. Crear una comunidad desde cero y desarrollarla satisfactoriamente no siempre surge espontáneamente sino que ha de basarse en un enfoque estructurado. Presentando este método esperamos fomentar el que cualquier persona que desee crear una comunidad se lance a la aventura de un modo más productivo.*

**Palabras clave:** *compartir conocimiento, comunidades, entornos colaborativos, metodología, relaciones sociales, software libre, Web 2.0.*

### Autores

**Stéphane Ribas** (M.Sc, University of Surrey 1996) ha trabajado 12 años en la industria del software y servicios, mayormente dedicado a tecnologías innovadoras tales como comercio electrónico, sistemas de salud electrónicos, seguridad en las telecomunicaciones, Intranet, sistemas expertos, *middleware*, aplicaciones de servidor, Java y software libre. Ha trabajado muchos años en varios países europeos y ha participado en diversos e importantes proyectos ofreciendo soporte y consultoría técnica para grandes clientes. Ha desarrollado importantes cualidades en la construcción y promoción de comunidades en Internet. Gracias a esas experiencias, ha ido adquiriendo un bagaje técnico muy sólido en software. Comenzó a trabajar en INRIA en 2008 para coliderar el Capítulo Local de OW2 Europa y contribuir en diversos proyectos de software libre y grupos de trabajo (Xwiki Concerto, AspireRFID, QualiPSO, NESSI OSS Working Group).

**Michel Cezon** ha trabajado alrededor de 25 años en industrias y servicios informáticos, mayormente en tecnologías innovadoras (inteligencia artificial, NLP, negocios electrónicos, EDMS, Intranet). Lideró numerosos proyectos europeos (FP4 EPTO-1993, FP6 SUPREME-1997, FP6 RENAISSANCE-1997, FP6 PRORAD-1998) además de grandes consorcios de carácter internacional (FP4 MOVIT-1995, United Nations/WIPO-2000), o proyectos multimillonarios para grandes clientes. Graduado como Ingeniero en Informática y especializado en inteligencia artificial, robótica y visión artificial, posee un dominio de la gestión de proyectos y de las habilidades de calidad aprendido en grandes compañías como Hewlett Packard o Cap Gemini Ernst and Young. Siendo consciente de las cuestiones de ámbito multicultural, pasó años en Singapur, Estados Unidos y Suiza, viajando frecuentemente alrededor de Europa. Se unió a INRIA en enero de 2007 para gestionar y coordinar proyectos europeos. Actualmente está trabajando en diversos proyectos europeos financiados por FP6 y FP7 (NESSI-Soft, NESSI-Grid, Qualipso, AspireRFID) y colidera el Capítulo Local de OW2 Europa.

grupo de personas que comparten valores y creencias comunes y que aprenden activamente unos de otros. Dichas comunidades han llegado a ser la plantilla de inicio para otros tipos de comunidades. Se basan en un tipo avanzado de diseño educativo o pedagógico considerado a menudo como un enfoque interdisciplinar para la educación superior [6]. Los participantes en comunidades de aprendizaje deben sentir algún tipo de lealtad con el grupo que dirige sus esfuerzos al

trabajo continuo y a ayudar a otros, influyendo en lo que ocurre en la comunidad (de manera activa o reactiva). Este tipo de comunidades son lo suficientemente flexibles como para dar la oportunidad a los participantes de expresar opiniones personales, pedir ayuda o información específica y compartir historias de eventos [4][6].

Una comunidad de interés podría ser considerada como un grupo de personas que com-

parten temas que no requieren una comunidad formal sino discusiones más hilvanadas para la colaboración y la compartición de conocimiento. Representan grupos de personas no demasiado unidas entre sí y sin un compromiso claro en términos de resultados. Están bien al corriente de los desarrollos en los temas de su interés y se dedican a formular y responder preguntas [4].

Una comunidad de apasionados está formada por un grupo de gente con un conjunto de actividades, formas de gobernabilidad y estructura más formales. Los miembros tienen un rol en concreto (por ej. "asesor de seguridad de redes"), y ayudan activamente a otros miembros a encajar y evolucionar en su rol, persiguiendo todos ellos obtener un dominio completo de su disciplina [4].

Las comunidades de práctica tienen una estructura menos formal y están basadas en especialidades de trabajo comunes entre sus integrantes. Los miembros tienen un rol particular o una especialidad concreta (por ej. seguridad) enfocado a la adquisición de experiencia y desarrollo de facultades en ese rol o especialidad. Un factor de motivación importante es el aprendizaje sobre la especialidad y la resolución de problemas [4].

La metodología discutida en este artículo se diseñó y aplicó a este último tipo de comunidad.

## 4. Método para construir y mantener comunidades de práctica

La metodología está formada por diversos pasos, comenzando desde un "objetivo común" hasta la "monitorización" (ver figura 1). Las diferentes fases pueden ser organizadas en cinco categorías principales [17]: Análisis, construcción, promoción, mantenimiento y monitorización.

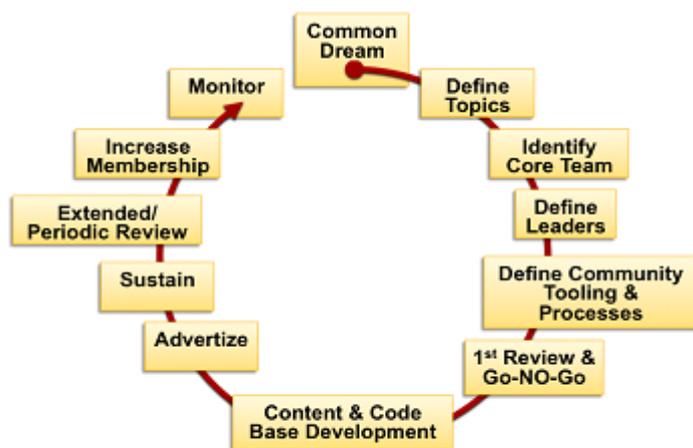


Figura 1. Método para crear y mantener una comunidad.

### 4.1. Análisis

El objetivo de la fase de análisis [3][4][7] consiste en comenzar el proceso y recoger toda la información relevante que permita comprender lo que queremos hacer, a dónde queremos ir, por qué y cómo. Como ya hemos mencionado, una comunidad puede emerger desde cero, a través de un idea común de un pequeño grupo de personas entusiastas, pero que para llegar a tener éxito, necesita un poco más de organización y gestión interna.

En esta fase, debes comenzar "soñando" sobre: ¿cómo trabajaría la comunidad idealmente?, ¿cómo debe estar organizada?, ¿quién liderará y tomará decisiones? En definitiva, deja volar tu imaginación libremente.

La identificación de la temática y subtemáticas permitirá aclarar tus ideas y compartirlas con tus colegas. Llegados a este punto, identificar el equipo principal también es importante y conocer quién está comprometido y deseando participar en este viaje.

Una vez que el equipo ha sido definido, debes tratar los temas básicos: concretamente ¿cómo se va a organizar el trabajo en equipo? Esto implica más pensamiento imaginativo y quizás sesiones de tormentas de ideas y de intercambio de información. Todos estos procesos abarcarán el gobierno interno, la comunicación, el ciclo de desarrollo, el entorno colaborativo, diseminación y promoción. De nuevo papel y lápiz deberán ser reemplazados por algunas herramientas más sofisticadas (preferiblemente software libre) para permitir un mejor y más eficiente trabajo en grupo. Identifica (y usa) las herramientas de la comunidad para organizar tus colaboraciones, tu contenido y facilitar tus actividades de diseminación (promoción y concienciación). Estas herramientas estarán centradas alrededor una forja de software que

permitirá una interacción y comunicación diarias, siendo el repositorio central de la comunidad y del proyecto. La metodología permite usar un conjunto de herramientas que ayuden durante esta fase.

A través de este proceso, debemos llegar a una idea mejor acerca de qué es lo que intentamos construir y llevar a cabo. Podría ser deseable parar en cierto punto y echar un vistazo alrededor para ver si ya existen comunidades o proyectos que compartan las mismas inquietudes y temas. Unirse a dichas iniciativas o comenzar desde cero requiere una decisión bien meditada y conlleva dos hojas de ruta diferentes.

Al final de la fase de análisis debe tomarse una decisión tipo "go-no-go", es decir, qué hay que hacer y qué no, en base a toda la información recolectada y a la hoja de ruta planificada. Los criterios para tomar tal decisión pueden ser objeto de un nuevo artículo en el futuro.

Una decisión de tipo "no-go" necesitará ser explicada en detalle y cuidadosamente para identificar cuales son los mayores impedimentos, dado el actual contexto, y evitar así realizar de nuevo la misma pregunta una y otra vez, cuando la iniciativa o las ganas de trabajar aparezcan nuevamente sobre la mesa.

Una decisión 'go' es incluso más difícil porque el tiempo y el esfuerzo gastado hasta el momento son más pequeños de lo que realmente será necesario para iniciar y construir la comunidad. Un compromiso fuerte y una clara comprensión del camino a seguir y de los esfuerzos necesarios son obligatorios y deben ser conocidos por todos los miembros principales del equipo.

Si seleccionaste la opción de comenzar una comunidad desde cero, debes pasar ahora a la segunda fase principal: la construcción.

### 4.2. Construcción

En esta fase es donde pones toda la carne en el asador. Comienzas codificando, creando documentación (por ejemplo guías de usuario, APIs, guías de desarrollo, casos de uso, etc.), configuras tu portal (lema de tu proyecto, licencia de uso, estado de desarrollo, cómo instalar el software, sección de descargas, capturas de pantalla, etc.), publicas paquetes de software, publicas tareas del proyecto y su planificación, gestionas voluntarios y contribuciones externas [8]. Tus actividades diarias serán guiadas por el gobierno de la comunidad, política de propiedad intelectual, ciclo de vida del proyecto y reglas de comunicación (simples y flexibles). Detallaremos más información sobre esta fase en un artículo futuro.

Una vez completadas estas tareas, es hora de darte a conocer.

### 4.3. Promoción

Tienes un equipo trabajando, un código fuente base, documentación y un portal donde la gente puede encontrar todo lo que necesitan saber, ya sea para contribuir o para usar tu código fuente. ¡Básicamente algo que mostrar! Tu comunidad se ha establecido, así que ahora debes publicarla: envía artículos a listas de noticias, usa listas de correo y envía mensajes a múltiples personas, usa tu propia red de contactos a tu alrededor y anima a los miembros de tu comunidad a hacer lo mismo. Pueden publicar noticias en periódicos locales e internacionales y revistas *online*. Podrías también usar conocidas redes sociales (por ej. LinkedIn) para dar a conocer tu comunidad. Una vez que tu comunidad está a pleno funcionamiento y has llegado a cierto nivel de reconocimiento, podrías acceder a la última fase: Mantenimiento.

### 4.4. Mantenimiento

Necesitas seguir en la cresta de la ola y mantener ese impulso conseguido y, como consecuencia, los líderes de la comunidad deberían dedicar tiempo y liderazgo a animar y estimular la comunidad [14][16]. Recomendamos que lleves a cabo eventos periódicos (por ej. jornadas informativas y de desarrollo, asambleas generales), audioconferencias y publicación regular de resultados en forma de notas, ser activo en foros, y algunas veces incluso copiar temas interesantes que provengan de fuentes de datos externas en tus noticias y foros internos. Recomendamos que seas transparente en cualquier decisión y que además escuches a la gente (permite que sea posible discutir cualquier tema o problema) [9][11].

También necesitas alcanzar una masa crítica de miembros [15] que lleven a cabo tus operaciones diarias (administración de sistemas, gestores de errores, correctores de errores, webmaster, gestor del proyecto, traductores...). Por tanto, tú también necesitas incrementar el número de personas en tu comunidad: propón iniciativas (por ej. concursos de codificación, *hackmeeting*, encuentros entre desarrolladores y usuarios, reuniones de proyecto, etc.), usa las redes sociales, explica cómo la gente puede participar y llegar a ser miembro u ofrece incentivos para unirse. No dudes en cruzar los límites de tu comunidad y recoger buenas ideas de otras comunidades (por ej. comunidades de video-jugadores *online*) [12].

Otros dos factores que son muy importantes para los usuarios externos que vayan a contribuir activamente en una comunidad son: el reconocimiento [13] y la compartición de conocimiento. Podrías proponer que los últimos usuarios registrados o los desarrolladores que más errores arreglan aparezcan en la página web de inicio. Después, debes dedicar recursos a organizar el conoci-

miento dentro de la comunidad: propón que haya maneras de compartir ese conocimiento (por ej. seminarios *online* [10], talleres, casos de uso, etc.).

Tal como dijimos en la fase de análisis, debes echar un vistazo a tu alrededor y buscar cualquier comunidad o proyecto que pueda estar interesado en usar tu código fuente o que pueda estar interesado en cooperar contigo. Identifica esas comunidades, acércate a sus líderes y proponles colaboración [3].

### 4.5. Monitorización

Para comprobar el progreso, las tendencias y validar los resultados de tus acciones, te recomendamos que definas tu propio sistema de monitorización mejor que usar plantillas predefinidas: identifica el objetivo (¿por qué y qué querías medir?), define las métricas, recoge los datos (analiza), identifica problemas y tendencias, define consecuentemente las acciones necesarias para hacer frente a ambos, ejecuta esas acciones y valida una vez más la salud del proyecto.

Te recomendamos que monitorices el crecimiento de tu comunidad, pero también la salud de tu proyecto, usando para ello métricas tales como número de errores generados, número de descargas, número de visitas, número de mensajes en los foros e intercambios de correos, número de eventos organizados, etc. Finalmente, ten en cuenta que debes monitorizar también la comunidad globalmente [9]: ¿Estás en una fase inicial de tu comunidad? ¿O en una fase de crecimiento de comunidad? ¿O en una fase de madurez? ¿O estás en el punto donde la comunidad está declinando (o reviviendo)? Una encuesta podría ayudarte a descubrir en qué punto se encuentra tu comunidad de desarrollo. Ofreceremos más pautas de actuación en un artículo futuro.

## 5. La comunidad del proyecto AspireRFID

Se trata de un proyecto parcialmente financiado por la Comisión Europea en su programa de I+D FP7. El proyecto ASPIRE se propone el desarrollo y la promoción de *middleware* libre, ligero, que siga los estándares, escalable, respetuoso con la privacidad e integrado, enfocado al desarrollo, el despliegue y la gestión de aplicaciones basadas en RFID y en sensores.

El consorcio ASPIRE quiso crear una comunidad vibrante alrededor de su marco de trabajo RFID basado en software libre y así asegurar el tiempo de vida del código base más allá del periodo de financiación del proyecto europeo. Por eso surgió la idea de crear una comunidad de práctica (y pasión) alrededor del código base.

Usamos este proyecto como una manera de

comprobar que nuestro método era válido a través de la auditoría del proyecto y diversas recomendaciones. Al mismo tiempo, validamos la metodología y afinamos la primera versión de configuración y herramientas.

Estando implicados en el desarrollo de ambos proyectos, ASPIRE y la comunidad OW2, encontramos rápidamente un punto de encuentro común en el dominio RFID: por una parte, el proyecto ASPIRE buscaba el desarrollo de un marco de trabajo basado en software libre para RFID y por otra parte, la iniciativa RFID fue promocionada por el consorcio OW2. Lo primero de todo consistió en despegar, mientras que el segundo paso consistió en buscar nuevas contribuciones. Fue una manera perfecta de trabajar con un gran beneficio mutuo.

Hasta ahora debemos resaltar tres objetivos cumplidos:

**Incrementar la velocidad de la fase de inicio:** mediante la metodología, hicimos una revisión de la comunidad existente y descubrimos que el consorcio OW2 podría ayudar en la creación de la comunidad AspireRFID: ellos tienen unas herramientas excelentes de comunidad que encajan con las necesidades, en cuanto a creación de comunidad, de AspireRFID. OW2 propuso una forja de tal manera que los desarrolladores pudieran ponerse a implementar rápidamente. Además, OW2 propuso también herramientas de difusión y promoción.

**Mejorar la diseminación y la promoción:** aplicamos el método y conseguimos resultados rápidamente, gracias a la promoción con acciones concretas como artículos, reuniones de desarrolladores, *hackmeetings*, conferencias de desarrolladores, etc. Encontramos que la lista de correo de OW2 fue un gran canal de revalorización y promoción.

Finalmente, el éxito de la diseminación y promoción incrementó los resultados de explotación: en un estudio comparativo sobre RFID realizado por el consorcio industrial PICOM [21] el marco de trabajo de ASPIRE fue puesto a prueba con productos comerciales. A pesar del hecho de que el proyecto no estaba finalizado y el marco estaba aún en desarrollo, ASPIRE cosechó mejores resultados que sus competidores. Podríamos suponer que el método ayudó a reforzar la valoración del producto, a publicitar su hoja de ruta, a mejorar su reputación y a asegurar la vida del proyecto a largo plazo.

## 6. Conclusión

Como en todo nuevo proceso, el método necesita ajustes y mejoras. Estamos trabajando para integrar nuestros descubrimientos en una versión revisada, pero a continuación exponemos los resultados más importantes:

■ El método debería ser aplicado de manera

correcta desde el principio para maximizar los beneficios; el proyecto comenzó unos meses antes de que esta metodología fuera aplicada.

■ El compromiso de las personas es clave. La comunidad está basada en personas y no hay métodos o herramientas que puedan reemplazar la buena voluntad y disponibilidad.

■ Dedicar tiempo a la transmisión de conocimiento es importante, así como encontrar a personal con cierto nivel de creatividad y liderazgo.

■ La visión de la comunidad debe ser compartida entre los miembros y claramente publicitada.

■ Los pasos de la metodología no son secuenciales; el solapamiento de procesos impulsará el calendario de trabajo.

■ El proyecto *AspireRFID* es un proyecto del FP7 donde el consorcio tiene que llevar a cabo ciertos entregables basados en una hoja de ruta que puede diferir de los deseos de los miembros de la comunidad. Los conflictos de intereses deben ser identificados y tratados públicamente tan pronto como sea posible.

Definitivamente, el método podría ser mejorado: la siguiente versión está en camino. Lanzar una comunidad y desarrollarla satisfactoriamente no es algo que surja espontáneamente (y por suerte), sino que se basa en un enfoque estructurado que se podría beneficiar de una metodología como la presentada en este artículo.

## 7. El consorcio OW2 y el capítulo local europeo

OW2 es una comunidad de software libre de carácter global, cuyo objetivo es el desarrollo de *middleware* distribuido bajo licencias libres, presentado en forma de componentes flexibles y adaptables. Estos componentes van desde marcos específicos de software y protocolos hasta plataformas integradas. Los desarrollos de OW2 siguen una aproximación basada en componentes. El consorcio es una organización independiente sin ánimo de lucro abierta a empresas, administraciones

públicas, universidades e individuos. La misión de OW2 consiste en desarrollar software libre *middleware* e impulsar una comunidad vibrante y un ecosistema de negocios.

Un Capítulo Local es un grupo de contribuidores que desean unir sus esfuerzos y promocionar los objetivos de un consorcio dentro de una comunidad caracterizada por su geografía o su lengua. Para conseguir representar a OW2 a nivel europeo, el capítulo local europeo fue aprobado y lanzado por el Consejo de OW2 el 15 de mayo de 2008. El capítulo local europeo generó una primera versión de sus estatutos y ha definido su estrategia través de flujos de trabajo y tareas: impulsar la comunidad de investigación y académica de OW2 en Europa, ayudar en el desarrollo del ecosistema de negocios de OW2 en Europa y proporcionar apoyo específico a las necesidades de la comunidad local.

Los objetivos se abordan a través de tres grandes líneas de acción: Comunicación con la parte académica, concienciación y promoción, y puesta en valor del proyecto. Se puede encontrar más información en <http://www.ow2.org/> y <http://europe.ow2.org/>.

## Referencias

- [1] OW2 Consortium. Europe Local Chapter, <http://europe.ow2.org/>.
- [2] *AspireRFID Project*. <http://fp7-aspire.eu>.
- [3] Stan Garfield. *Implementing a Successful KM Programme*. Ark Group Australia, 2007. <http://stangarfield.googlepages.com>.
- [4] E. Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1998. ISBN: 0-521-66363-6.
- [5] PLUME Project. *Promoting economical, Useful and Maintained software for the Further Education And THE Research communities*. <http://www.projet-plume.org/en>.
- [6] P. Goodyear, M. De Laat, V. Lally. Using Pattern Languages to Mediate Theory-Praxis Conversations in Designs for Networked Learning.

ALT-J, *Research in Learning Technology*. 1741-1629, Volume 14, Issue 3, 2006, pp. 211 – 223.

[7] E. Wenger, R. McDermott, W. M. Snyder. *Cultivating Communities of Practice*. Harvard Business School Press, 2002. ISBN:1578513308.

[8] Karl Fogel. *Producing Open Source Software*, <http://producingoss.com/>.

[9] C. Latemann, S. Stieglitz. *Framework for Governance in Open Source Communities*. Postdam University, 2005. ISBN 0-7695-2268-8.

[10] OW2 Europe Local Chapter. Webinar Platform, <http://www.ow2.org/view/Activities/EuropeLocalChapterWebinars>.

[11] M. Gouvêa, C. Motta, F. Santoro. Recommendation as a Mechanism to Induce Participation in Communities of Practice, *UFRJ, UNIRIO, Brazil*, 2006. ISBN 978-3-540-32969-5.

[12] C. Ruggles, G. Wadley, M. Gibbs. Online Community Building Techniques Used by Video Game Developers. *Lecture Notes in Computer Science*, 2005. ISBN 978-3-540-29034-6. <http://www.springerlink.com/content/47dc4v3xej51p5m4/>.

[13] C. Cruz, M. Gouvêa, C. Motta, F. Santoro. A Proposal for Recognizing Reputation within Communities of Practice. *UFRJ, UNIRIO, Brazil*, 2008. ISBN 978-3-540-92718-1.

[14] P. Muller. Reputation, trust and the dynamics of leadership in communities of practice. *Journal of Management and Governance*, 2006 ISSN 1385-3457 (Print) 1572-963X (Online).

[15] Fabernovel Consulting. *Modèles économiques des logiciels open source et logiciels libres* (2007) [http://www.fabernovel.com/businessmodels\\_opensource.pdf](http://www.fabernovel.com/businessmodels_opensource.pdf).

[16] L. M. Sama, V. Shoaf. Ethical Leadership for the Professions: Fostering a Moral Community. *Journal of Business Ethics*, 2007 ISSN 0167-4544 (Print) 1573-0697 (Online).

[17] McDermott Consulting. Communities of Practice, <http://www.mcdermottconsulting.com/communitypractice.shtml>.

[18] *Audiodanzine*. French music webzine, <http://fr.audiodanzine.com/>.

[19] JRES 2009. *Journées réseaux*, <https://2009.jres.org/>.

[20] *Linux Online*. WorldWide Linux User Groups <http://www.linux.org/groups/>.

[21] PICOM. *Pôle des industries du commerce*, Lille-France, <http://www.picom.fr/>.

## ¿Estudiante de Ingeniería Técnica o Ingeniería Superior de Informática?

Puedes aprovecharte de las condiciones especiales para hacerte

socio estudiante de ATI

y gozar de los servicios que te ofrece nuestra asociación,

según el acuerdo firmado con la

Asociación RITSI

Infórmate en [www.ati.es](http://www.ati.es)

o ponte en contacto con la Secretaría de ATI Madrid

[secretmdr@ati.es](mailto:secretmdr@ati.es), teléfono 91 402 93 91



Martin Michlmayr  
Open Source Program Office, Hewlett-Packard

<tbm@hp.com>

# Dinamización de comunidades en proyectos de software libre

**Traducción:** Teófilo Romera Otero (GSyC/LibreSoft, Universidad Rey Juan Carlos)

## 1. Introducción

Las comunidades en Internet están reclamando cada vez más atención por diversas razones. Una de ellas es la enorme abundancia de conocimiento que los usuarios que participan en estas comunidades han creado de forma colaborativa. Internet ha jugado un importante papel en la creación de este conocimiento al posibilitar que gente con fines comunes se encuentre y compartan sus pasiones e intereses. Según Lessig [1], las oportunidades de colaboración que ofrece Internet han llevado a una transformación de una cultura predominantemente de sólo lectura a una de lectura y escritura. En otras palabras, menos individuos permanecen como consumidores pasivos y cada vez más gente comparte sus ideas de manera activa y crean nuevo conocimiento o productos. En lugar de meramente ver televisión, los aficionados usan Internet para discutir acerca de un espectáculo con sus amigos; además de leer libros, los fans escriben sus propias versiones de ficción o remezclas musicales y las comparten con otros.

Incluso en Internet, la mayoría de usuarios son consumidores pasivos pero hay una tendencia hacia una dedicación más activa. Esto ocurre en parte porque los sitios web se diseñan cada vez más con el objeto de atraer aportación y participación activa de usuarios y clientes. O'Reilly [2] utiliza el término *arquitectura de la participación* para describir "la naturaleza de los sistemas diseñados para la participación de los usuarios".

Hay muchos ejemplos de sitios en Internet que han utilizado esta arquitectura de la participación para atraer una base significativa de colaboradores activos. Wikipedia es la afamada enciclopedia que "cualquiera puede editar", Flickr es un sitio para compartir fotos que almacena incontables fotos de alta calidad (muchas de ellas bajo licencias Creative Commons), Amazon dispone de una abundante cantidad de reseñas de libros contribuidas por los propios lectores y quién no ha oído hablar de YouTube, que permite "emitirse a uno mismo". Lo que hace especiales a estos sitios, no es la tecnología subyacente que puede ser replicada fácilmente por cualquiera, sino la comunidad de usuarios que han atraído, que contribuye al sitio y hacen del sitio lo que es.

Reconociendo la importancia de las comuni-

**Resumen:** la creciente complejidad de las comunidades establecidas alrededor de proyectos de software libre junto con un aumento del interés de la explotación comercial de los mismos, ha llevado a que cada vez más de estas comunidades estén incorporando un puesto de "dinamizador de la comunidad". El dinamizador de la comunidad actúa como enlace entre empresa y comunidad y se asegura de que haya una buena relación entre los dos. También promueve la comunidad, ayuda a su crecimiento y cuida de que permanezca sana. Este artículo discute la dinamización de comunidades en proyectos de software libre y ofrece una visión sobre el papel del dinamizador de comunidades.

**Palabras clave:** colaboración, comunidad, dinamización, software libre, voluntarios.

## Autor

**Martin Michlmayr** ha estado participando en varios proyectos de software libre desde hace más de 10 años. Ha sido coordinador voluntario del proyecto GNUstep, Director de Publicidad de Linux International y líder del proyecto Debian. Durante los dos años al frente de Debian, Martin representaba al proyecto y llevaba a cabo importantes tareas de coordinación y organización. Martin ingresó en Hewlett-Packard en 2007 donde trabaja como experto en comunidades de software libre y dinamizador de la comunidad de FOSSBazaar. Martin también forma parte del Consejo de Open Source Initiative (OSI). Es diplomado en filosofía, sicología e ingeniería del software y doctor por la universidad de Cambridge.

dades en Internet, Seth Godin, un conocido experto en marketing, identificó el papel del organizador de comunidades en Internet (o dinamizador de comunidades<sup>1</sup>) como uno de los más importantes trabajos del futuro [3]. Los dinamizadores de las comunidades ayudan tanto a la creación de comunidades como al sustento y crecimiento de comunidades existentes. Este artículo discutirá la dinamización de comunidades y el papel de dinamizador de comunidades en el contexto de los proyectos de software libre, que se apoyan en comunidades de desarrolladores y otros participantes para crear software. En la sección siguiente se discute el fenómeno de la dinamización de comunidades y los diferentes puntos de vista del puesto de dinamizador de comunidades.

## 2. Una definición de dinamización de comunidades

Los proyectos de software libre tienen una larga historia de desarrollo y participación de comunidades. El paradigma del software libre fue iniciado por individuos que compartían el objetivo común de crear software que diera a los usuarios más libertades que el software propietario [4]. Aunque el software libre puede ser desarrollado por una compañía sin implicación por parte de participantes externos, la mayoría del mismo es, o bien desarrollado por una comunidad de voluntarios (que pueden o no estar recibiendo un salario por parte de una empresa con interés en el software), o por una empresa que lidera

el desarrollo de manera colaborativa junto con una comunidad de usuarios interesados en mejorar ese software. La comunidad es una característica definitoria de la mayoría de proyectos de software libre, junto con la licencia y las metodologías de desarrollo. La literatura académica también asocia los métodos colaborativos en los que se basa el software libre con varios aspectos positivos, como los altos niveles potenciales de revisión [5] y la innovación a través de los usuarios<sup>2</sup> [6].

Aunque la comunidad siempre ha tenido un papel importante en los proyectos de software libre, la dinamización de comunidades a cargo de personas con dedicación permanente y el puesto de dinamizador de la comunidad en sí, es un fenómeno relativamente reciente. Esto es en parte debido a que el éxito del software libre ha dado lugar a un enorme crecimiento de su base de desarrolladores y usuarios y esta evolución de la comunidad puede asociarse a ciertas dificultades en su crecimiento que necesitan ser gestionadas [7]. Además, los modelos económicos "híbridos" que combinan economías comerciales y de compartición están siendo empleadas de forma creciente en el contexto del desarrollo del software libre, y las interrelaciones entre ambas han de ser gestionadas activamente. Lessig [1] define un híbrido como "o bien una entidad comercial que busca impulsar el valor de una economía participativa, o bien una economía

participativa que construye una entidad comercial para ayudar a sostener sus objetivos". Mozilla, un proyecto de software libre que desarrolla el popular navegador de Internet Firefox, es un ejemplo de esto último puesto que la organización que lo sostiene ha obtenido fondos sustanciales a través de diferentes asociaciones con diferentes entidades. Existen innumerables ejemplos de empresas cuyos modelos de negocio se basan en Linux y otros proyectos de software libre y que tienen intereses clave en asegurarse de que las comunidades relacionadas se mantengan sanas.

Hay numerosas razones por las que las empresas quisieran establecer grandes comunidades alrededor de los proyectos de software libre que construyen. Además de obtener participantes que ayudan en el desarrollo y hacen evolucionar el software, el tener una comunidad le da a la gente un lugar en el que crear un vínculo con la empresa [8]. Esto, a su vez, puede llevar a una cierta lealtad a la marca, innovación de productos, y usuarios que pueden llegar a ser defensores de la compañía y sus productos y que pueden crear marketing viral. Red Hat (con Fedora) y Novell (con OpenSUSE) son dos buenos ejemplos de empresas de la industria alrededor de Linux que han establecido proyectos dirigidos por comunidades de los que las empresas se benefician de diversas maneras.

Los beneficios de las comunidades y la necesidad de promoverlas y gestionarlas han propiciado la figura del dinamizador de comunidad. Mientras que tradicionalmente el líder de proyecto se encargaba de que existiera una comunidad fuerte alrededor de su proyecto, actualmente las responsabilidades a menudo se reparten entre diferentes personas. El líder de proyecto es típicamente visto como una figura de tipo técnico que tiene la responsabilidad de hacer progresar el software y de ocuparse de los desarrolladores. El dinamizador de la comunidad, por otro lado, se asegura de que hay una comunidad sana en el proyecto, interactúa con los usuarios, los desarrolladores y otros actores y coordina aspectos organizativos del proyecto. Cuando una empresa es la fuerza motriz que impulsa un proyecto, el dinamizador de la comunidad actúa como enlace entre la compañía y la comunidad y se asegura de que hay una buena relación entre las dos.

Muchos proyectos de software libre exitosos que son dirigidos por empresas o que tienen altos niveles de implicación empresarial, tienen en estos momentos un dinamizador de la comunidad dedicado. Mientras muchos de ellos utilizan el título de "dinamizador de la comunidad"<sup>3</sup>, en realidad hay diferencias sustanciales entre las diferentes visiones existentes de este papel. Adam Williamson [9], antiguo empleado de Mandriva y actualmente

trabajador de Red Hat, hizo la siguiente observación: *Es muy interesante observar las diferencias existentes en cómo el rol es percibido; Yo era una especie de portavoz de la comunidad de usuarios en Mandriva, Jono [Jono Bacon, dinamizador de la comunidad de Ubuntu] considera "comunidad" esencialmente a la comunidad de desarrolladores/colaboradores y Zonker [Joe Brockmeier, dinamizador de la comunidad de OpenSUSE] dice que su papel acaba siendo más que nada evangelización.*

Esta observación muestra que las responsabilidades del dinamizador de la comunidad son muy amplias. Tienen que tratar con los usuarios, los desarrolladores, los colaboradores y otros actores. A menudo actúan como defensores y promotores y llevan a cabo muchas otras tareas (las más importantes serán discutidas en el apartado siguiente). Una de las funciones clave de un dinamizador de la comunidad es asegurarse de que todo el mundo puede contribuir de manera óptima. Greg DeKoenigsberg [10], arquitecto de la comunidad de Red Hat, recalcó en una entrevista que *"es una figura que se centra en hacer la vida de los colaboradores individuales tan productiva y tan sencilla como sea posible"*. DeKoenigsberg comenta sobre el papel de dinamizador de la comunidad: *Quieres hacer crecer la base de colaboradores, como sea. Quieres que tus desarrolladores principales se orienten a la comunidad, por supuesto. Pero también quieres que pasen la mayor parte de su tiempo haciendo lo que mejor saben hacer, que es (o eso espero) programar. Esto significa que quieres que sea alguien más quien se preocupe de mantener tu comunidad de colaboradores feliz y creciente.*

Asegurarse de que los colaboradores están a gusto y sacan partido de su participación en el proyecto es importante porque de otra manera podrían dejar de contribuir y dirigir sus energías a otros proyectos. En cualquier caso, esto lleva a preguntarse qué factores y valores son importantes para los voluntarios en proyectos colaborativos. Mark Shuttleworth, fundador del proyecto Ubuntu, sugiere que los voluntarios cuentan con encontrar las siguientes condiciones [1]:

■ **Respeto:** la comunidad y sus colaboradores han de ser respetados y considerados como corresponde.

■ **Responsabilidad:** los colaboradores de la comunidad deben tener suficiente autoridad en el proyecto. Esto es particularmente importante en proyectos liderados por compañías puesto que tienen que equilibrar ese liderazgo con adjudicar a los colaboradores externos autoridad e influenciar en la dirección del proyecto.

■ **Significado:** los colaboradores quieren tener la sensación de que el proyecto para el que contribuyen tiene sentido, "quieren sentir que forman parte de algo que es grande,

importante y bonito". Si un proyecto no da a los voluntarios la sensación de que sus aportaciones y el proyecto en sí mismo tienen significado, invertirán su tiempo en un proyecto que sí lo haga.

Estos valores son importantes porque están fuertemente relacionados con la motivación de los voluntarios y la única manera de influenciar el trabajo voluntario es a través de motivación. Jim Grisanzio [11], antiguo dinamizador de la comunidad de OpenSolaris, hizo algunos comentarios en su blog acerca de la distinción entre poder y liderazgo que son importantes en este contexto: *No se puede sustituir liderazgo con poder. Los líderes de voluntarios obtienen cooperación apelando a los valores compartidos de la comunidad y eso es un ejercicio mucho más exigente que dictar ordenes con amenazas o forzar a la gente a echarse atrás.*

En otras palabras, los dinamizadores de la comunidad no tienen autoridad sobre las actividades de colaboradores voluntarios (no pagados) y tienen que encontrar mejores modos que el poder para animarlos a trabajar en ciertas actividades. Han de liderar la comunidad escuchando sus necesidades y ayudando al mismo tiempo.

En resumen, la figura de dinamizador de comunidades es un desafío que incluye muchas tareas y responsabilidades diferentes. Los dinamizadores de la comunidad interactúan con varios actores y llevan a cabo muy diversas actividades. No tienen poder sobre los voluntarios y han de encontrar incentivos interesantes, como ofrecer un significado al proyecto, si quieren motivar a los voluntarios.

### 3. Tareas y aptitudes para la dinamización de comunidades

Como puede deducirse de la descripción de la figura de dinamizador de la comunidad expuesta anteriormente, un dinamizador de la comunidad ha de llevar a cabo varias y diversas actividades, incluyendo:

■ **Habilitación:** una de las más importantes responsabilidades del dinamizador de comunidades es la de habilitar a la comunidad de diferentes maneras. Por ejemplo, el dinamizador de la comunidad ha de ayudar a que esta crezca mediante la captación de nuevos miembros para el proyecto y ayudándoles a iniciarse. Además, el dinamizador de la comunidad mantendrá en mente el "panorama general" del proyecto de manera que pueda poner en contacto a la gente. Si alguien ha trabajado o está trabajando en la solución a un problema que alguien más está sufriendo, pero no se conocen el uno al otro, el dinamizador de la comunidad podría ponerlos en contacto para que colaboren. Saber qué está pasando en todo el proyecto también es importante porque permite al dinamizador de

la comunidad sugerir actividades desatendidas a nuevos miembros del proyecto. La comunicación en general es muy importante tanto en pequeños como en grandes proyectos y es tarea principal del dinamizador de la comunidad el hacer circular noticias y mantener a la gente informada de lo que ocurre en el proyecto. Finalmente, resulta crucial asegurar interacciones saludables dentro de la comunidad. Muchos proyectos, especialmente cuando llegan a cierto tamaño, atraerán *trolls* que aportan energía negativa y hacen perder tiempo a los colaboradores. Algunos proyectos han adoptado un Código de Conducta que gobierna las comunicaciones y las interacciones dentro del proyecto. Es responsabilidad de todo el proyecto que este Código de Conducta sea respetado pero el dinamizador de la comunidad debería intervenir si nadie más lo hace.

- **Delegación:** como se ha tratado anteriormente, la habilitación resulta una responsabilidad clave. No es responsabilidad del dinamizador de la comunidad el llevar a cabo todas las tareas del proyecto. De hecho, el dinamizador de la comunidad debería tener la precaución de no ser demasiado activo en algunas áreas para no desanimar la participación de voluntarios potenciales. Por ejemplo, si un proyecto tiene un foro de soporte, el dinamizador de la comunidad no debería responder inmediatamente a todas las preguntas. En su lugar, es deseable que anime a otros voluntarios a ayudar y que traslade las preguntas a desarrolladores expertos que podrían contestarlas fácilmente. El dinamizador de la comunidad debería asegurarse de que el foro sirve de ayuda y puede por supuesto contestar preguntas de vez en cuando, especialmente si nadie más lo hace.

- **Marketing:** los dinamizadores de la comunidad a menudo llegan a jugar un papel de promotores. Esto podría implicar viajar a conferencias para hablar sobre un proyecto y animar a nuevos voluntarios a que se unan. También puede incluir hablar con la prensa sobre ciertos aspectos del proyecto. Finalmente, los dinamizadores de la comunidad son a menudo encargados de mantener buenas relaciones con otros proyectos y empresas.

- **Escucha y observación:** una de las ventajas clave de tener una comunidad grande es recabar sugerencias y opiniones de un gran número de personas con puntos de vista diferentes. Es responsabilidad del dinamizador de comunidades escuchar las necesidades y las ideas de los colaboradores y asegurarse de que esas reflexiones llegan a la gente adecuada dentro del proyecto o dentro de la empresa que lo promueve. Además de escuchar a la comunidad, el dinamizador de comunidades ha de comprobar que las sugerencias son tenidas en cuenta y que se hace algo al respecto de ellas.

- **Creación de una visión:** como ya se ha expuesto, los voluntarios necesitan una motivación clara para contribuir a un proyecto en

lugar de hacerlo a otro. El dinamizador de la comunidad ha de ayudar a crear una cultura común para la comunidad así como una visión del proyecto. Es importante recoger sugerencias de los colaboradores y tenerlas en cuenta a la hora de hacer avanzar la comunidad en una dirección determinada.

- **Información:** el dinamizador de la comunidad ha de estar al tanto e informar a la empresa sobre el estado de salud de la comunidad. Al mismo tiempo ha de utilizar esta información con miras a mejorar la comunidad.

Basándonos en estas actividades, las aptitudes que se pueden derivar como vitales para un dinamizador de comunidades son varias:

- **Comunicación:** el dinamizador de la comunidad es un comunicador. Ha de interactuar con usuarios, desarrolladores, colaboradores y otros actores interesados o implicados en el proyecto. Ha de poner al día a la gente de lo que está ocurriendo en el proyecto y trabajar junto con representantes de otros proyectos, compañías o con la prensa. Es esencial tener habilidades para la escritura y muy recomendable saber hablar en público. Mas aun, puesto que muchos proyectos grandes son normalmente internacionales, es muy útil que los dinamizadores de comunidades conozcan y sepan como tratar con otras culturas de modo que colaboradores de otras culturas se sientan bienvenidos al proyecto.

- **Marketing:** dar a conocer el proyecto es muy importante para atraer nuevos colaboradores y potencialmente también para atraer otros recursos, como financiación. Aunque no es habitual que el dinamizador de la comunidad haga marketing en el sentido tradicional, no es raro que se le haga responsable del "marketing de la comunidad".

- **Coordinación:** organizar actividades varias es otra de las responsabilidades clave para los dinamizadores de comunidades y por eso han de ser buenos coordinadores. Por ejemplo, puede que tengan que organizar eventos tales como reuniones de usuarios, realizar el seguimiento de tareas que han de ser llevadas a cabo e identificar quién está al cargo de cuáles.

- **Habilidades técnicas:** aunque el dinamizador de comunidades no es una figura técnica, siempre ayuda saber algunos detalles técnicos. Esto es especialmente útil para entender problemas técnicos o retos dentro del proyecto y así poder ayudar a nuevos voluntarios a encontrar el área en que pueden participar mejor. Las habilidades técnicas pueden también ayudar cuando el dinamizador de la comunidad quiere montar un foro nuevo o cualquier otra infraestructura para el uso de la comunidad y el resto de la gente está demasiado ocupado con otras cosas.

- **Paciencia y buena disposición:** el dinamizador de la comunidad ha de escuchar

a muchos colaboradores e interactuar con gente que es nueva en el proyecto y por tanto no está familiarizada con sus prácticas y su cultura. Además, hacer crecer la comunidad puede llevar mucho tiempo [12]. Es por esto que la paciencia y la buena disposición son virtudes importantes para un dinamizador de la comunidad.

- **Estar visible y estar presente:** como ya se ha comentado, el dinamizador de la comunidad no debería contestar todas las preguntas del foro o resolver los problemas de todo el mundo. Pero ha de estar disponible y visible de manera que los colaboradores sepan que siempre tendrán alguien a quién recurrir si tienen un problema. Foster [13] argumenta que ser un dinamizador de comunidades no es "un trabajo de 9 a 5", en parte porque ha de responder inmediatamente cuando hay una crisis y en parte porque las comunidades internacionales a menudo engloban muchas zonas horarias.

#### **4. Actividades para crear comunidad**

Existen muchas maneras de construir y hacer crecer una comunidad. Citamos a continuación varias actividades comunes empleadas por proyectos de software libre para dar una visión general del tipo de actividades que resultan adecuadas.

- **Reuniones cara a cara:** mientras que el trabajo diario en proyectos de software libre se lleva a cabo típicamente a través de Internet, las reuniones cara a cara han mostrado ser muy útiles por varias razones. Una de ellas es que son muy efectivas, especialmente cuando dan lugar a una lluvia de ideas o a resolver problemas conocidos. Otra razón es que las reuniones presenciales permiten que gente, que normalmente interactúa por Internet, se conozca en persona. Esto puede ayudar mucho a la colaboración futura, motiva a la gente y ayuda a crear un sentimiento de comunidad más fuerte.

- **Stands en conferencias:** pueden servir para conseguir nuevos voluntarios y permiten la interacción entre desarrolladores y usuarios. Ir a conferencias y ferias puede ser una buena motivación para los desarrolladores porque pueden conocer a la gente que usa su software, y también para muchos usuarios resulta grato conocer a los desarrolladores del software que utilizan. Encontrarse con los usuarios es también muy buena manera de obtener sugerencias para mejorar el software.

- **Formación en Internet:** varios proyectos han empezado recientemente a organizar eventos especiales para recién llegados que estén interesados en conocer más acerca del proyecto y de cómo contribuir. Por ejemplo, la *OpenSUSE Community Week* consiste en varias sesiones de formación acerca de diversos aspectos de OpenSUSE. La formación se imparte a través de IRC (Internet Relay Chat), un sistema de chat a menudo utilizado por

proyectos de software libre como medio vital de comunicación.

■ Reuniones de caza de errores<sup>4</sup>: otra manera de facilitar que los recién llegados participen y de conseguir que algunos voluntarios colaboren de manera regular es organizar eventos (presenciales o a través de Internet) durante los cuales se analizan y reparan errores en el software. Estos eventos son a menudo coordinados por algún experto del proyecto en el control de calidad<sup>5</sup> y junto a muchos otros desarrolladores experimentados que ayudan a los nuevos colaboradores a empezar arreglando su primer fallo en el código. Esto puede convertirse en una muy buena motivación para los novatos que bien podrían decidir involucrarse más en el proyecto.

■ Boletín de noticias: publicar anuncios y noticias sobre el proyecto de manera regular es una estupenda manera de mantener a la comunidad informada sobre el desarrollo del proyecto. El boletín de noticias puede además ser utilizado para publicar peticiones de ayuda o para promover actividades que los usuarios pudieran encontrar interesante.

Como se ha mencionado anteriormente, estos son sólo algunos ejemplos de actividades que los proyectos de software libre utilizan con éxito para construir sus comunidades. De algún modo, las actividades que saldrán bien dependerán de las particularidades del proyecto. Pero parece sensato observar otros proyectos para obtener ideas de actividades que pueden llevarse a cabo.

### 5. Conclusiones

Los proyectos de software libre se han apoyado siempre en comunidades de colaboradores que mejoran el software. Debido al incremento del tamaño de las comunidades y del interés comercial en las mismas, el papel de dinamizador de la comunidad está ganando relevancia. Un dinamizador de la comunidad puede actuar como el enlace entre la empresa y la comunidad y puede asegurarse de que haya una buena relación entre las dos. La figura del dinamizador de la comunidad comprende actividades diversas, desde la habilitación hasta la promoción. Este amplio conjunto de responsabilidades se refleja en las aptitudes que un dinamizador de la comunidad debería poder poner sobre la mesa, incluyendo muy buenas habilidades para la comunicación y la organización así como paciencia y buena disposición.

### Referencias

- [1] **Lawrence Lessig.** *Remix*. Bloomsbury Academic, London, 2008. ISBN: 978-1408113479.
- [2] **Tim O'Reilly.** *The Architecture of Participation* (2004). <[http://www.oreillynet.com/pub/a/oreilly/tim/articles/architecture\\_of\\_participation.html](http://www.oreillynet.com/pub/a/oreilly/tim/articles/architecture_of_participation.html)>.
- [3] **Seth Godin.** Jobs of the future, #1: Online Community Organizer (2007). <[http://sethgodin.typepad.com/seths\\_blog/2007/07/jobs-of-the-fut.html](http://sethgodin.typepad.com/seths_blog/2007/07/jobs-of-the-fut.html)>.
- [4] **Richard Stallman.** The GNU Operating System and the Free Software Movement. In: DiBona, C., Ockman, S., Stone, M. (Eds.), *Open Sources: Voices from the Open Source Revolution*. pp. 53-70 (1999).
- [5] **Eric S. Raymond.** *The Cathedral and the Bazaar*. O'Reilly & Associates, 1999. ISBN-10: 1565927249.
- [6] **Karim R. Lakhani, Eric von Hippel.** How open source software works: 'free' user-to-user assistance. *Research Policy*, 32(6), 923-943 (2003).
- [7] **Martin Michlmayr, Benjamin Mako Hill.** Quality and the Reliance on Individuals in Free Software Projects. *Proceedings of the 3rd Workshop on Open Source Software Engineering*, 105-109 (2003).
- [8] **Dawn Foster.** *Why Companies Should Have Online Communities* (2008). <<http://fastwonderblog.com/2008/04/16/why-companies-should-have-online-communities/>>.
- [9] **Adam Williamson.** *LinuxFest recap* (2009). <<http://www.happyassassin.net/2009/04/27/linuxfest-recap/>>.
- [10] **Greg DeKoenigsberg.** Interview with Greg DeKoenigsberg, Red Hat Community Architect. *LinuxQuestions.org* <<http://www.linuxquestions.org/questions/interviews-28/interview-with-greg-dekoenigsberg-red-hat-community-architect-725426/>>.
- [11] **Jim Grisanzio.** *The Distinction Between Power and Leadership* (2009). <[http://blogs.sun.com/jimgris/entry/the\\_distinction\\_between\\_power\\_and](http://blogs.sun.com/jimgris/entry/the_distinction_between_power_and)>.
- [12] **Matthias Stürmer.** *Open Source Community Building*. Master thesis, University of Bern, 2005.
- [13] **Dawn Foster.** *What Does it Take to Manage a Community?* (2007). <<http://fastwonderblog.com/2007/09/03/what-does-it-take-to-manage-a-community/>>.

### Notas del traductor

<sup>1</sup> En el original se utiliza el término *community manager*, que hemos optado por traducir como *dinamizador de la comunidad* en la versión en castellano.

<sup>2</sup> Del original *user innovation*, se refiere al término acuñado por Eric von Hippel para describir la innovación llevada a cabo por los usuarios y consumidores finales en lugar de por los proveedores.

<sup>3</sup> Nuevamente, *community manager* en el original.

<sup>4</sup> En el original *Bug squashing parties*. Hace alusión al doble sentido de la palabra en inglés *bug* que generalmente se traduce como bicho pero además sirve como sinónimo de error en argot informático. Por otro lado, el uso que se hace de *squash* en este contexto, significa aplastar. De modo que la frase original se podría haber traducido literalmente como "comitiva de aplastamiento de bichos". Pero lo que se describe es una reunión en la que los desarrolladores dedican a buscar y solucionar errores en el código.

<sup>5</sup> De las siglas QA (*Quality Assurance*) en el original.

Cristina Breña, Andrés  
Leonardo Martínez Ortiz  
Telefónica Investigación y Desarrollo

<{crisbb,almo}@tid.es>

# La Comunidad Morfeo: estrategias Open Source para la Open Innovation



Cristina Breña y Andrés Leonardo Martínez, 2009. Este documento se distribuye bajo la licencia Creative Commons "Reconocimiento-Compartir Igual" disponible en: <<http://creativecommons.org/licenses/by-sa/3.0/es>>.

## 1. Introducción

Una de las mayores contribuciones a la revolución tecnológica del siglo XX ha sido la popularidad que han alcanzado los métodos de colaboración colectiva. Internet y algunos de sus servicios más famosos, como Flickr, Wikipedia o Facebook, son resultados de la cooperación. Asimismo, estos servicios se han convertido en paradigmas colaborativos [1][2]. Mucho antes, comunidades de código abierto como Debian, Linux kernel, Apache, Mozilla u OpenOffice ya habían definido el modelo de colaboración para el desarrollo del software y produjeron la infraestructura requerida [3][4] para soportar una comunidad de desarrolladores de ámbito mundial.

No obstante, existen todavía muchos problemas complejos [5] cuya solución pasa por la definición de mecanismos colaborativos entre los agentes interesados en resolverlos. Por ejemplo, la gestión de la investigación científica, el desarrollo tecnológico y, aún más complicado, los procesos de innovación continúan sin ser resueltos. Así, podemos encontrar todavía algunos escenarios donde parece difícil definir modelos de colaboración, como por ejemplo las colaboraciones entre empresas, donde la competitividad parece inherente al mercado.

Es en este contexto, en la búsqueda de soluciones para promover la productividad, la ventaja competitiva y las estrategias efectivas para competir en el mercado del software, donde surgió la Comunidad de software libre Morfeo en la que están implicados todos los interesados en los procesos de I+D+i.

El proyecto Morfeo [6] es un ecosistema tecnológico y económico compuesto por empresas, administraciones públicas, universidades y grupos de investigación, centros tecnológicos y grupos industriales, y pequeñas y medianas empresas (PYMES). Todas estas organizaciones están implicadas en la investigación, el desarrollo y la innovación de la producción tecnológica, y comparten los siguientes objetivos:

- La promoción de estándares abiertos y plataformas de software.
- La definición de modelos de negocio basados en las tecnologías licenciadas como abiertas.
- La promoción de los mercados competitivos de provisión tecnológica como medio para lograr soluciones innovadoras, produc-

**Resumen:** Morfeo es una comunidad de Software Libre, cuyos miembros son grandes empresas, administraciones públicas, universidades y centros de investigación, centros tecnológicos y pequeñas y medianas empresas. Todos ellos comparten el principio de que el Software Libre es una estrategia efectiva para mejorar los procesos de investigación, desarrollo e innovación de la tecnología (software). El software de plataformas permite definir modelos de negocio viables y relaciones win-win que impulsan la colaboración. MyMobileWeb, ezWeb y las contribuciones de CENATIC y OpenFwPA, framework de e-Administración del Principado de Asturias, son casos de éxito de producción tecnológica en la Comunidad Morfeo. No obstante, la mejora de la comunicación interna/externa, el desarrollo de nuevas herramientas que soporten la colaboración entre empresas y la búsqueda de soluciones para otros problemas complejos de colaboración empresarial tales como la producción de estándares o el desarrollo empresarial, constituyen retos actuales de la Comunidad Morfeo.

**Palabras clave:** fomento empresarial, forjas, innovación abierta, software libre.

## Autores

**Cristina Breña** es estudiante de último curso de periodismo en la Universidad Rey Juan Carlos. Durante su periodo de estudiante, ha trabajado como locutora de radio y ha obtenido becas para el estudio en el extranjero (Helsinki) en 2007. Actualmente, está haciendo prácticas de periodismo en Telefónica I+D, ayudando en tareas de comunicación. Está a cargo de elaborar estudios de viabilidad a través de redes sociales, escribir notas de prensa y artículos, y presentar mejoras y refuerzos del plan de Comunicación de la Comunidad Morfeo.

**Andrés Leonardo Martínez Ortiz** es Licenciado en Informática por la Universidad Politécnica de Madrid. Desde 2007 es especialista en OSS (*Open Source Software*) en Telefónica I+D. Durante el periodo entre 1998 y 2002 fue ingeniero del departamento de I+D de Teldat, un fabricante de infraestructuras y desde 2002 hasta 2007 fue miembro activo del grupo de investigación GSyC/LibreSoft de la universidad Rey Juan Carlos, donde era profesor. Empezó en software libre cuando era ingeniero de I+D en Teldat, donde portó el *kernel* de Linux a la arquitectura M860. En la actualidad, es miembro del Proyecto Morfeo, donde se encarga de la dinamización de la comunidad y participa en iniciativas como OSS marketplace o la definición de programas para la certificación y la formación de tecnologías OSS. Participa en Vulcano y Qualipso que tienen objetivos similares: promover la adopción del software libre en la industria. Como resultado de esta participación, está definiendo y montando el Centro de Competencia de Morfeo, que se incluye en la red de centros de competencia de Qualipso, y gestiona el desarrollo de nuevas herramientas de colaboración. Forma parte de grupos de trabajo en software libre de INES OSS y NESSI OSS, la asociación IEEE y ACM.

ción tecnológica más barata y evitar situaciones de oligopolio/monopolio.

Cada miembro tiene su propia motivación (beneficio derivado de la cooperación) para participar en las actividades de la comunidad, como por ejemplo:

- Las universidades y grupos de investigación pueden complementar (más o menos) sus investigaciones con actividades de transferencia tecnológica realizadas por grandes empresas y PYMES.
- Las administraciones públicas, como grandes consumidoras de tecnología y a veces también proveedoras de ella, necesitan una gran base de proveedores/clientes y mejorar

sus posiciones para definir estándares de administración electrónica.

- Las empresas consiguen masas críticas en sus inversiones tecnológicas reduciendo el riesgo de adopción tecnológica.
- Pequeñas y medianas empresas pueden participar en la última parte de la cadena de valor disminuyendo el riesgo de crear nuevos servicios y productos. En otras palabras, las PYMES pueden acceder a las actividades de I+D+i y vencer sus restricciones financieras.

## 2. El proyecto Morfeo

A finales del año 2004, el Proyecto Morfeo fue creado conjuntamente por Telefónica Investigación y Desarrollo (TID), una compañía

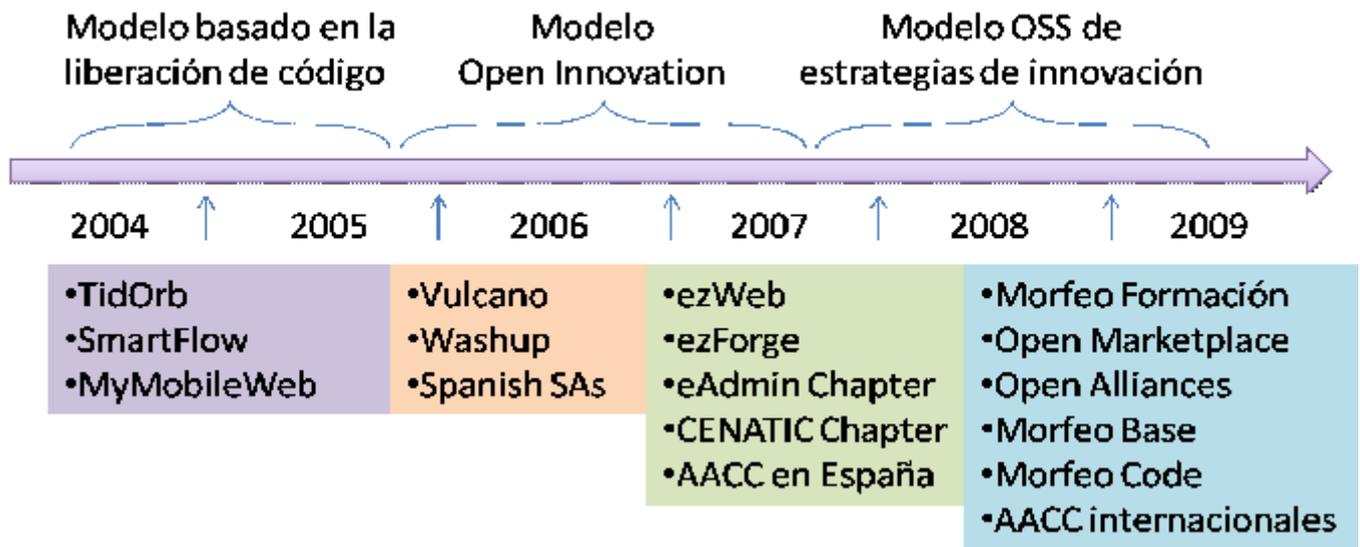


Figura 1. Evolución del modelo de la Comunidad Morfeo.

que pertenece al grupo Telefónica, la Universidad Politécnica de Madrid (UPM) y la Universidad Rey Juan Carlos (URJC). La iniciativa persiguió demostrar la viabilidad no sólo en el uso del software de código abierto en las industrias sino también en la participación de las compañías en el desarrollo del software de código abierto. La evolución de la comunidad muestra cómo fueron tomados diferentes enfoques para alcanzar los objetivos de Morfeo (ver figura 1).

**Desde mitad del año 2004 hasta el final del 2005: liberación de código**

Durante ese tiempo, Telefónica I+D fue ayudado por la URJC y la UPM en el proceso de liberación de proyectos de software internos (privativos) para convertirlos en proyectos de código abierto. Fueron liberados; TidOrb, una implementación en Java de un Object Request Broker de Corba, SmartFlow, un motor de estados basado en redes de Petri, y MyMobileWeb, un marco para desarrollar contenidos web para dispositivos móviles. Todos ellos eran productos maduros con

una hoja de ruta cercana al final e intensamente validados dentro del grupo Telefónica. Los principales retos fueron construir alrededor de esos productos una comunidad (tradicional) dirigida principalmente a desarrolladores voluntarios. Para hacer esto, por ejemplo, se creó una forja de comunidad [7], que es una herramienta colaborativa para el desarrollo del software.

Hoy en día, se puede decir que el éxito de esta estrategia no ha sido uniforme. Mientras que para TidOrb y SmartFlow fue imposible crear una comunidad, MyMobileWeb ha tenido un éxito destacable ya que en torno a esta tecnología existe una comunidad pequeña, pero activa formada por desarrolladores en todo el mundo. La conclusión principal que obtenemos de esta etapa es que es muy difícil mezclar en la misma comunidad socios industriales, sin experiencia en el desarrollo de software de código abierto, con miembros de la comunidad OSS (*Open Source Software*) tradicional, con importantes prejuicios hacia las compañías y sus necesidades e intereses.

**Desde 2006 hasta el final del 2007: construir una comunidad de innovación abierta**

En la mitad del 2005, el *Board* de la Comunidad Morfeo se dio cuenta de las dificultades de mezclar la visión tradicional del OSS con la industria en Morfeo. El principal problema fue un desconfianza mutua: los usuarios tradicionales de software de código abierto no tenían confianza acerca de la intención de la industria y cómo esta podría contribuir a la comunidad de OSS; por otra parte, la industria tuvo problemas para adoptar la tecnología de código abierto tanto en el soporte de sus procesos de negocio como en la entrega de servicios y productos, principalmente debido al poco conocimiento sobre la tecnología OSS y su proceso de desarrollo. Por lo tanto, se optó por cambiar de estrategia: construir una comunidad de innovación abierta [8] para organizaciones que participan en procesos de I+D+i. Así, Morfeo empezó a construir un ecosistema de investigación, desarrollo e innovación tecnológica con suficiente calidad como para ser aceptada bajo criterios industriales. La respuesta fue inmediata: los implicados en I+D+i encontraron un esquema para resolver el riesgo en la actividades relacionadas con la administración tecnológica. Como resultado, Morfeo tuvo que definir nuevas reglas para fomentar la participación con nuevos canales de comunicación y nuevas infraestructuras. También, por ejemplo, se hizo una remodelación de la comunidad adaptada a las nuevas clases de miembros que empezaron a formar parte de ella. Esta vez los resultados fueron bastante distintos. Desde 2006 hasta ahora, Morfeo ha incubado muchas propuestas de I+D+i. Muchas de ellas han tenido financiación pública y el total del presupuesto ha estado en torno a los 25 millones de euros. El progreso de los presupuestos se puede ver en la figura 2.

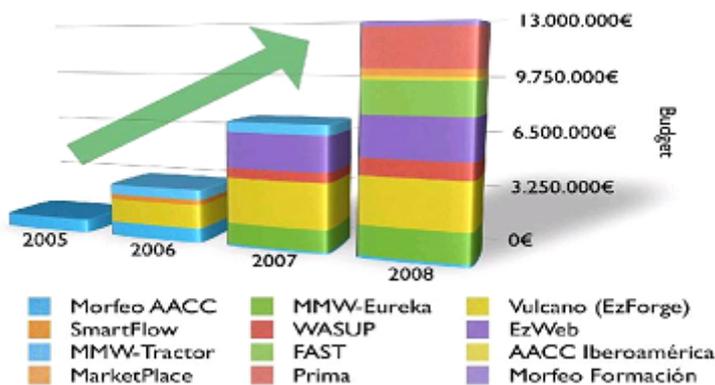


Figura 2. Evolución del presupuesto total de Morfeo (© 2008 Javier Soriano).

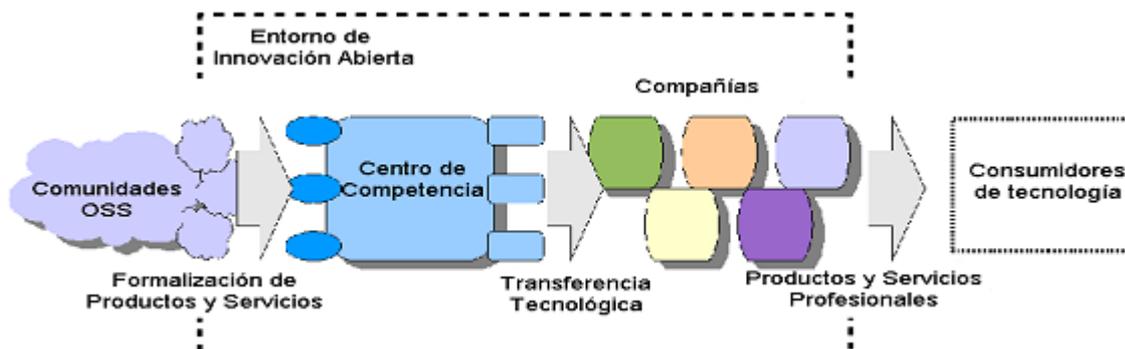


Figura 3. Centro de Competencia de Morfeo.

**Desde 2008 hasta ahora: Estrategia OSS para la innovación**

Desde luego, la historia previa ha mostrado cómo la estrategia de software de código abierto puede ser usada para crear un esquema de colaboración no competitiva entre las organizaciones. Por lo tanto, fue lógico abordar nuevos desafíos para intentar aplicar estas estrategias a otros problemas complejos en las relaciones entre empresas. A finales de 2008, los miembros de Morfeo empezaron a trabajar en nuevos problemas: los procesos de estandarización, el desarrollo de servicios profesionales basados en la tecnología OSS y el desarrollo de facilidades en las relaciones empresariales para las PYMES. Estas tres iniciativas han recibido los siguientes nombres: Morfeo Open Alliance, Centro de Competencia Morfeo y Morfeo Base.

Morfeo Open Alliance es una iniciativa para crear un consorcio de empresas que compartan una visión común de las tecnologías y de sus arquitecturas asociadas. Además, este consorcio acuerda unas especificaciones abiertas y potencia las implementaciones de código abierto de los componentes en la arquitectura acordada. Este modelo de colaboración acelera la definición de estándares lo que se traduce en un periodo de tiempo más corto en la comercialización y en un bajo riesgo de

inversión tecnológica. La iniciativa ha sido probada en un dominio específico: Service Front-end [9].

El Centro de Competencia Morfeo [10] es una iniciativa para fomentar el desarrollo de servicios profesionales basados en tecnologías OSS. El concepto del centro de competencia en OSS fue introducido por el proyecto QualiPSo [11] como un modo de "dirigir ayudas a las industrias y gobiernos en busca de abastecimiento innovador y competitivo, estimulando el uso de software de código abierto flexible y de bajo precio para desarrollar sistemas de información viables e innovadores". El centro de competencia Morfeo es una propuesta para ser incluida en la Red QualiPSo de centros de competencia. El centro de competencia pretende impulsar la adopción del OSS para fomentar el desarrollo de servicios profesionales de tecnologías de OSS que pueden ayudar a reducir el riesgo de la adopción de OSS en la industria. La figura 3 muestra su estructura y modelo de negocio.

Finalmente, Morfeo Base [12] es la iniciativa para aumentar el desarrollo económico de las PYMES. Morfeo en esta propuesta está trabajando en la definición de las estrategias de modulación de las cadenas de valor bajo

aproximaciones colaborativas. Como caso especial, la comunidad está tratando de desarrollar una estrategia específica basada en la *comoditización* de soluciones software (la Ley de Christensen [13]) y trasladar valores añadidos desde los productos (vendiendo licencias) hasta los servicios (pago por uso). Esto permitiría crear un ecosistema de las PYMES alrededor de las comunidades OSS.

**3. La estructura de una comunidad de innovación abierta**

La estrategia actual del Proyecto Morfeo está basada en una estructura específica de comunidad. Esto es así porque la Comunidad Morfeo no es como el resto de las comunidades de código abierto. Morfeo no está compuesta principalmente por voluntarios y desarrolladores individuales de software sino por agentes de los procesos de I+D+i. También, la clase de instituciones que forman parte de la comunidad hacen de Morfeo la única comunidad con dicha estructura.

La comunidad está organizada en proyectos y capítulos (ver figura 4). Un capítulo no es más que una agrupación de proyectos que tienen alguna clase de relación entre ellos. Cada proyecto tiene sus propias reglas debido a que Morfeo es muy flexible en ese criterio. En el Proyecto Morfeo no hay que pagar ninguna cuota por ser miembro, tan sólo seguir estas tres condiciones:

- 1) Tecnología libre: los resultados del proyecto deben ser libres bajo el criterio OSI [14].
- 2) Gestión de la Propiedad Intelectual definida: Así, los contribuidores potenciales pueden conocer las condiciones de colaboración.
- 3) Fuerte mentor: La propuesta de proyecto debe ser liderada al menos por una organización como una empresa, universidad o grupo de investigación, una administración pública, un centro tecnológico o una PYME.

Incluso para aquellos proyectos que no cumplen con la tercera condición, Morfeo les ofrece una comunidad más tradicional conocida como Morfeo Bazaar. En cualquier caso, para estar en Morfeo, el "futuro" miembro debe enviar una solicitud de alta de proyecto en la que tiene que rellenar los objetivos y una

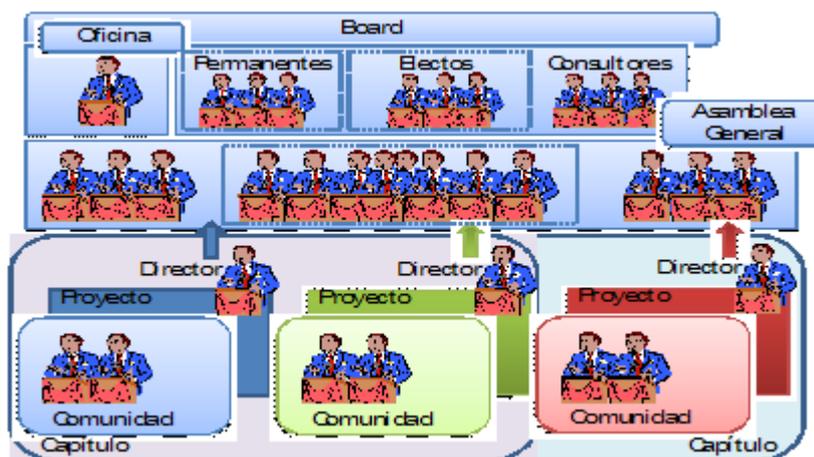


Figura 4. Modelo organizativo de Morfeo.

descripción clara del proyecto, el consorcio, la licencia y la política de gestión de propiedad intelectual. Después de las validaciones, el proyecto se convertirá en un proyecto Morfeo y los participantes en el consorcio del proyecto entrarán a formar parte de la Comunidad Morfeo.

Cada proyecto tiene un representante en la Asamblea General donde el *Board* informa cada año acerca del estado de la comunidad, comparten *roadmap* (hoja de ruta), nuevas propuestas de servicios horizontales, nuevos proyectos y miembros. El *Morfeo Board* es el comité ejecutivo de la comunidad formado por miembros permanentes, como por ejemplo, los miembros fundadores o permanentes, los representantes de proyectos elegidos para ser miembros del *Board* y consultores, que son profesionales prominentes de OSS con gran habilidad asesora. El *Board* está al cargo de identificar nuevas estrategias de OSS y de introducir nuevos servicios para fomentar la colaboración entre los miembros de Morfeo. Finalmente, la Oficina Morfeo es la encargada de las actividades diarias como son el mantenimiento de la infraestructura, la administración de los proyectos que se aprueban, las tareas de comunicación, etc. La oficina Morfeo apoya al *Board* dándole información para la toma de decisiones.

#### 4. Produciendo tecnología de innovación abierta

Actualmente Morfeo tiene diferentes proyectos tecnológicos, reflejando la evolución de la comunidad. Así, podemos encontrar los siguientes tipos de proyectos:

■ **Proyectos heredados:** MyMobileWeb [15] "es una plataforma de software de bajo-coste, modular, basado en estándar abiertos que simplifica el desarrollo de aplicaciones web de alta calidad para móviles y portales. Proporciona una mejora en los contenidos y en las adaptaciones de las aplicaciones". Inicialmente fue desarrollado por Telefónica I+D y liberado como un proyecto OSS cuando comenzó la comunidad. Después de eso, MyMobileWeb ha sido capaz de crear una comunidad alrededor suyo. Está liderada por Telefónica I+D y es una de los pocos casos de éxito de esta clase de proyectos.

■ **Proyectos de innovación abierta:** EzWeb [16] es una plataforma *mash-up* que cuenta con mecanismos enriquecidos de comunicación entre *gadgets* y que permite construir el *front-end* de una nueva generación de arquitectura SOA. Es asimismo la propuesta de Morfeo de *front-end* de la nueva Internet del futuro. La plataforma ha sido desarrollada desde cero en el ámbito de la comunidad. Dentro del proyecto participan muchas organizaciones, principalmente PYMES, pero también trabajan con universidades y administraciones públicas. Este proyecto está liderado por Telefónica I+D aunque actualmente hay otras empresas que se están implicando en la

tecnología EzWeb. Claramente se puede decir que éste es otro de los logros en los proyectos de la comunidad de innovación abierta.

■ **Proyectos comunitarios:** el capítulo CENATIC [17] y OpenFWPA [18] son ejemplos de proyectos comunitarios. En estos casos las administraciones públicas o las agencias públicas establecieron un marco de trabajo con la Comunidad Morfeo. Ellos ofrecen claras estrategias de adopciones de OSS o tecnología OSS para la implementación de la administración electrónica, mientras que Morfeo ofrece una activa comunidad de organizaciones implicadas en procesos de I+D+i. Esta clase de proyectos no están liderados por ninguno de los miembros fundadores, sino por organizaciones externas que selecciona la Comunidad Morfeo para incubar esta clase de proyectos. Los proyectos comunitarios son un ejemplo del efecto expansivo que tiene la comunidad que se refleja en un entorno de soporte tecnológico.

#### 5. Conclusiones

El Proyecto Morfeo ha conseguido algunos éxitos importantes en estos años. Morfeo ha sido capaz de definir estrategias efectivas de OSS para la gestión de la innovación. Se ha construido un entorno de innovación abierta donde la gestión de la propiedad intelectual abierta y basada en licenciamiento libre es un aspecto fundamental. Además, Morfeo es un referente en el programa público de I+D+i ya que ha llegado a ser un verdadero entorno de transferencia tecnológico, fomento empresarial y crecimiento económico. A pesar de que Morfeo comenzó siendo una iniciativa de ámbito español, actualmente, tiene un reconocimiento también europeo. La comunidad ha empezado a colaborar en Latinoamérica con la incorporación de una nueva red de oficinas en Iberoamérica (Brasil, Argentina, Uruguay, Paraguay y Chile).

#### Reconocimientos

Este artículo no hubiera sido posible hacerlo sin la experiencia, opiniones y comentarios de algunos de los principales contribuidores del Proyecto Morfeo. Ha sido un placer entrevistar a las siguientes personas: Pedro Acebes, Juan Antonio Cáceres, Jesús M. González, Álvaro Polo, Marcos Reyes y Roberto Santos.

#### Referencias

- [1] D. Tapscott, A.D. Williams. *Wikinomics: How mass collaboration changes everything*. Portfolio Hardcover, 2006. ISBN-10: 1591841380.
- [2] C. Shirky. *Here comes everybody: The power of organizing without organizations*. The Penguin Press HC, 2008. ISBN-10: 1594201536.
- [3] E.S. Raymond. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, 2001. ISBN-10: 0596001312.
- [4] K. Fogel. *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly Media, 2005. ISBN-10: 0596007590.
- [5] Software Engineering Institute, Carnegie Mellon University. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. <[http://www.sei.cmu.edu/uls/files/ULS\\_Book2006.pdf?bcsi\\_scan\\_87A666907766D0F0=0&bcsi\\_scan\\_filename=ULS\\_Book2006.pdf](http://www.sei.cmu.edu/uls/files/ULS_Book2006.pdf?bcsi_scan_87A666907766D0F0=0&bcsi_scan_filename=ULS_Book2006.pdf)>.
- [6] Morfeo Project. <<http://www.morfeo-project.org>>.
- [7] Morfeo Forge. <<http://forge.morfeo-project.org>>.
- [8] H. W. Chesbrough. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business School Press, 2005.
- [9] Morfeo Open Coalition for Service Front Ends. <<http://sfe.morfeo-project.org>>.
- [10] Morfeo Competence Center. <<http://cc.morfeo-project.org>>.
- [11] QualiPSO Project. <<http://www.qualipso.org>>.
- [12] Morfeo Base. <<http://base.morfeo-project.org>>.
- [13] C.M. Christensen, M.E. Raynor. *The Innovator's Solution: Creating and Sustaining Successful Growth*. Harvard Business School Press, 2003. ISBN-10: 1578518520.
- [14] Open Source Initiative. Section about Open Source Definition and Open Source Licenses <<http://www.opensource.org>>.
- [15] MyMobileWeb Project. <<http://mymobileweb.morfeo-project.org>>.
- [16] ezWeb Project. <<http://ezweb.morfeo-project.org>>.
- [17] Morfeo Capítulo CENATIC. <<http://cenatic.morfeo-project.org/>>.
- [18] OpenFWPA Project. <<http://openfwpa.morfeo-project.org>>.

Frank van der Linden  
Philips Healthcare, Países Bajos

<frank.van.der.linden@philips.com>

# Aplicación de los principios del software libre en líneas de producto

**Traducción:** Tomás Aguado Gómez (GSyC/LibreSoft, Universidad Rey Juan Carlos)

## 1. Introducción

En las últimas décadas, los proveedores de sistemas empotrados han introducido la Ingeniería de Líneas de Producto con el objeto de mejorar la gestión de la diversidad de sus productos así como para reducir el esfuerzo de desarrollo. El movimiento hacia la Ingeniería de Líneas de Producto normalmente tiene por objetivo la reutilización a gran escala, llevando a la reducción de costes y del período de comercialización así como a la mejora de la calidad y la reducción de los costes de mantenimiento.

La Ingeniería de Líneas de Producto Software es un medio de desarrollar aplicaciones software usando plataformas y personalizaciones masivas. Las personalizaciones en masa permiten la producción eficiente y rápida de un producto individualizado para el cliente. Los artefactos utilizados para los diferentes productos han de ser lo suficientemente adaptables para encajar en los diferentes sistemas que se producen en la línea de producto. Esto supone la necesidad de gestionar la variabilidad de todos los artefactos de la línea de producción. Específicamente una línea de producto generalmente define la plataforma de una arquitectura, sus componentes y las herramientas de apoyo para proporcionar una gestión del cambio (variabilidad) eficiente y estandarizada.

Otras tendencias en la Ingeniería del Software no están siempre relacionadas con líneas de producto software. Sin embargo, la aplicación de estas metodologías en una organización que las desarrolle podría ser beneficioso. Asimismo, otros enfoques de la Ingeniería del Software podrían beneficiarse de la introducción de principios basados en el desarrollo de líneas de Producto. Esto último aplica al desarrollo de software libre como un modelo válido y eficiente de desarrollo de software.

Existen diferentes opciones a la hora de desarrollar software libre. Sin embargo, el denominador común de todas es que un grupo distribuido de personas trabaja conjuntamente para generar software. La principal diferencia con el enfoque basado en líneas de Producto es que las comunidades de desarrollo de software libre tienen motivaciones propias y específicas tratándose a menudo de desarrollos con el único objetivo de ofrecer solución a un problema determinado y de compartir esta solución con otros.

**Resumen:** *la Ingeniería de Líneas de Producto mejora la gestión de la variabilidad y la reutilización de sistemas empotrados. Ayuda a explotar la personalización en masa con el objeto de presentar al cliente un producto individualizado. Sin embargo la Ingeniería de Líneas de Producto también ha llevado a procesos de desarrollo muy densos y a la necesidad de una planificación global para un rango completo de productos (una línea de producto). El desarrollo de software libre no ha tenido un impacto profundo en las compañías de desarrollo de sistemas empotrados, en parte debido a que la cultura inherente al desarrollo de software libre no se mantiene demasiado fiel al establecimiento de procesos de desarrollo pesados y de planificación global. Sin embargo ambos esquemas de desarrollo necesitan (a menudo) de desarrollo distribuido, y es en este punto donde ambos pueden aprovechar los logros del otro. Este artículo investiga las diversas opciones en las que el software libre y las metodologías de desarrollo asociadas pueden utilizarse para reducir los problemas que aparecen en el desarrollo distribuido así como para aumentar la calidad del software resultante. Una pequeña parte del artículo estudia las opciones disponibles para que el software libre se beneficie de las metodologías de gestión de la variabilidad de las líneas de producto software y de este modo aumentar las posibilidades de personalizar los productos resultantes.*

**Palabras clave:** *comercialización, fuente interna, gestión del cambio, líneas de producto software, software libre.*

## Autor

**Frank van der Linden** trabaja en las oficinas de Philips Healthcare. Recibió su doctorado en Matemáticas en 1984 en la Universidad de Amsterdam y desde entonces ha trabajado para Philips. Sus intereses más recientes incluyen las metodologías de desarrollo software y arquitecturas, principalmente metodologías orientadas a componentes y objetos. En 1991 se implicó en el desarrollo de una Línea de Producto dentro de Philips, aspecto en el que ha trabajado hasta el día de hoy. Fue jefe de proyecto en tres proyectos ITEA sucesivos que versaban sobre Ingeniería de Líneas de Producto, y posteriormente en desarrollo distribuido incluyendo metodologías de desarrollo de software libre. Como parte de estos proyectos fue miembro del comité organizativo de talleres y conferencias sobre líneas de producto (PFE & SPLC). Entre 2005 y 2008 fue líder del proyecto ITEA COSI sobre desarrollo distribuido y prácticas de software libre en la industria de sistemas empotrados. En este proyecto ha organizado gran cantidad de talleres sobre Software Libre y Líneas de Producto. Es coautor de diversos libros y numerosos artículos sobre estos temas.

No existe un único beneficio a la hora de aplicar conceptos del software libre a la Ingeniería de Líneas de Producto. Del mismo modo, debido al diverso uso que se hace del software libre, la Ingeniería de Líneas de Producto puede ser atractiva para las comunidades de desarrollo de software libre. Sin embargo, muchas de las prácticas seguidas desde el enfoque de Ingeniería de Líneas de Producto no encajan con las seguidas en las comunidades de software libre. Por ejemplo los procesos *gestionados* no son siempre viables o aceptados. Siempre será beneficioso combinar las ventajas de ambos enfoques aunque esto no será posible si no se eliminan estas incompatibilidades. Actualmente, existe una interacción limitada entre las comunidades de software libre y las de desarrollo de líneas de producto, aunque este interés ya ha sido plasmado en anteriores estudios [3]. Este artículo investiga las relaciones entre el software libre y el desarrollo de líneas de

producto. Se basa en los resultados obtenidos en el proyecto COSI de ITEA [5]. A partir de este estudio se pueden establecer las siguientes conclusiones:

- Implantar una línea de producto software implica realizar una gran inversión, planificaciones a muy largo plazo, gestión explícita del cambio y desarrollo *distribuido*.
- La industria informática usa el software libre como medio para obtener software de calidad de manera rentable. Esto ayuda a reducir costes compartiendo esfuerzos, y disminuye el tiempo de desarrollo del proyecto a través de metodologías ágiles de desarrollo; todos estos procedimientos constituyen en sí mismos una manera *distribuida* de concebir y desarrollar sistemas.
- El problema de la paralelización en el ámbito del desarrollo distribuido es la principal inspiración para intentar combinar las prácticas de Software Libre y de la Ingeniería de Líneas de Producto.

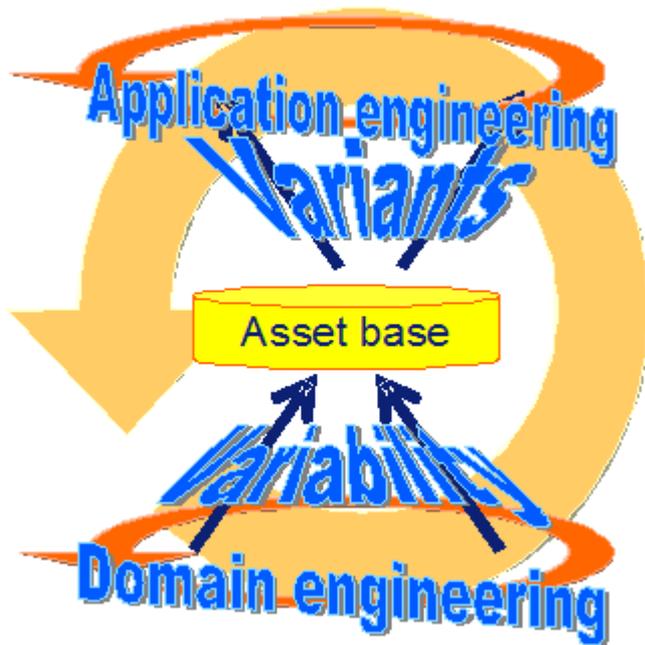


Figura 1. Procesos de Ingeniería de Líneas de Producto.

En este artículo se analizan los diferentes aspectos de la Ingeniería de Líneas de Producto, y se relacionan con prácticas de Software Libre. El artículo finaliza con un resumen de la aplicación de prácticas de Ingeniería de Productos en el desarrollo de Software Libre.

**2. Conceptos básicos de Ingeniería de Líneas de Producto**

**2.1. Dos procesos de desarrollo**

La Ingeniería de Líneas de Producto hace énfasis en la separación entre los conceptos *construir una plataforma robusta* y *crear en un corto espacio de tiempo aplicaciones personalizadas para el cliente*. Estos dos conceptos conducen a dos procesos interrelacionados de desarrollo. (ver figura 1):

- Establecimiento de la plataforma (Ingeniería de Dominio), incluyendo la definición de características comunes y de variabilidad de todos los elementos que componen la línea de producto.
- Derivación de aplicaciones (Ingeniería de Aplicación), incluyendo la vinculación de la variabilidad en las aplicaciones.

La separación en estos dos procesos también indica la separación de los problemas más importantes relacionados con la gestión del cambio. La Ingeniería del Dominio es responsable de asegurar que la variabilidad es la apropiada para el desarrollo de aplicaciones. La Ingeniería de Aplicación se focaliza en el desarrollo de sistemas individuales sobre la plataforma. Reusa la plataforma y vincula a las aplicaciones. Estos dos procesos están pensados para funcionar de manera desacoplada pero sincronizados por las

implementaciones de la plataforma. Debido a que estos dos procesos están débilmente acoplados pueden ser gestionados por modelos de ciclo de vida totalmente diferentes.

**2.2. Variabilidad**

La Ingeniería de Líneas de Producto Software [16] [11] tiene por objetivo apoyar un amplio espectro de productos, orientados a diferentes clientes específicos. En vez de comprender cada sistema individual por sí mismo la Ingeniería de Líneas de Producto Software observa la línea de producto como un todo y sobre éste la variación de todos los sistemas individuales. Igualmente importante es conocer las características comunes de todos los componentes que se gestionan en una línea de producto. A través de esta Ingeniería se deben gestionar tanto las características comunes como la variabilidad. La variabilidad se define mediante la Ingeniería de Dominio y se explo-

ta durante la Ingeniería de Aplicación vinculando las variantes apropiadas. La variabilidad se introduce a lo largo de muchas fases durante el proceso de Ingeniería de Dominio.

En cada nivel se refina la variabilidad del nivel anterior y se introducen características adicionales que no son resultado de este refinamiento. Este proceso se describe en la figura 2. La variabilidad se define inicialmente durante el análisis de las necesidades de todas las entidades implicadas para ser posteriormente refinada a lo largo de las siguientes fases de proceso. A esto se le denomina *variabilidad interna*. Las necesidades específicas de cada fase llevan a la introducción de nuevas variabilidades internas.

Establecer la infraestructura de una línea de producto no es un objetivo en sí mismo. El objetivo primordial es su explotación a lo largo del proceso de Ingeniería de Aplicación. A esto también se le denomina *instanciación* de la variabilidad. Los siguientes términos, que describen entidades autocontenidas en todo tipo de artefactos de desarrollo, son importantes en la gestión de la variabilidad:

- *Punto de Variación*: Describe dónde se encuentran las diferencias entre los sistemas finales.
- *Variante*: Las diferentes posibilidades que existen para satisfacer un *punto de variación*.

En la mayoría de los casos los puntos de variación no cambian independientemente. La selección de una variante específica para un punto de variación dado, influye en las opciones posibles para otros puntos de variación. Para permitir la gestión de la variabilidad, es necesario un modelo de variabilidad para asegurar que se realiza una elección consistente de variantes a lo largo de todos los puntos de variación. En propuestas tempranas de modelado de variabilidad, este modelado se integraba con la notación subyacente (utilizando por ejemplo la herencia). Sin embargo, se admite comunmente que es necesario establecer una distinción clara entre

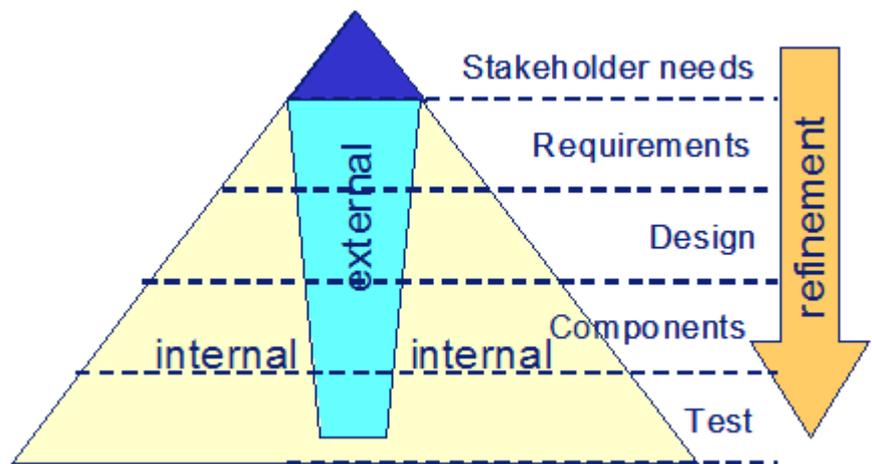


Figura 2. Pirámide de Variabilidad.

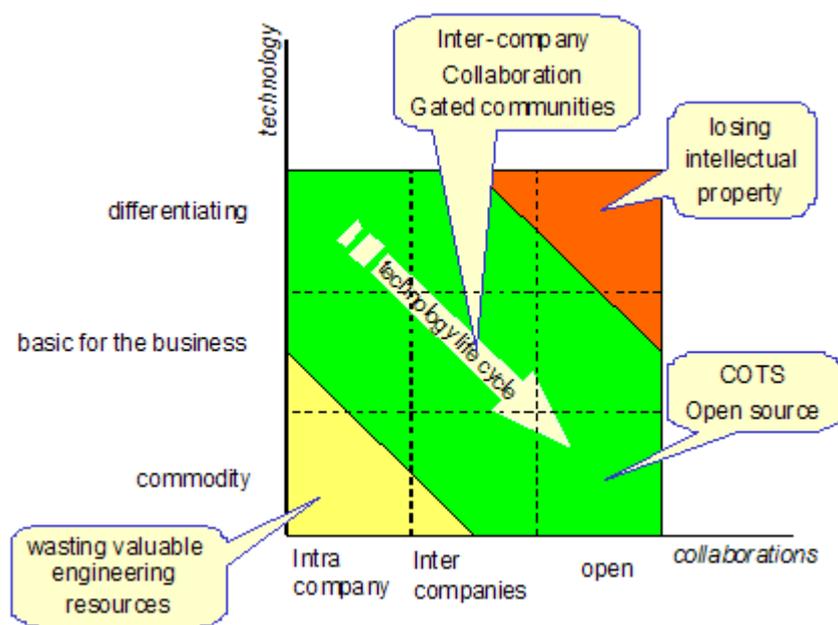


Figura 3. Desarrollo eficiente y efectivo de software.

el modelo de variabilidad y el del propio sistema. Esto hace mucho más sencillo aplicar ajustes complejos, da un apoyo mejorado a la toma de decisiones y es mucho más escalable [1].

Existen diversas propuestas de modelos de variabilidad. Los grupos más importantes son los Modelos de Variabilidad basados en Árboles de Características [10] y los Modelos de Variabilidad Ortogonal (OVM) [16].

También debe distinguirse entre variabilidad en el tiempo y en el espacio [2]. Ambos tipos de variabilidad son posibles con puntos de variación y variantes diseñados y aplicados de manera apropiada. La variabilidad en el tiempo está asociada a la evolución de la línea de producto y representa la existencia de diferentes versiones de un artefacto que son válidas en diferentes lapsos de tiempo. Esto último está relacionado con el versionado de los elementos del sistema. La variabilidad en el espacio está referida a la existencia de un artefacto en diferentes formas dentro del mismo espacio de tiempo. Esto lleva a la existencia de diferentes configuraciones que conforman sistemas válidos al mismo tiempo. Estas prácticas están relacionadas directamente con la Ingeniería de Líneas de Producto.

### 2.3. Software libre aplicado a líneas de producto

El interés de las empresas por el software libre tiene su origen en el reconocimiento del hecho de que la práctica totalidad del software se está convirtiendo en una materia prima de primera necesidad, lo que hace que sea muy interesante para los desarrolladores de sistemas empotrados introducirlo tanto en sus productos como en las líneas de producto asociadas.

### 3. La mercantilización del software

Como consecuencia de la mercantilización del software [13] gran parte del mismo ha dejado de estar particularizado para un producto concreto. De hecho actualmente no es común que un producto software esté desarrollado por una única compañía. Por el contrario, el software se produce en estrecha colaboración dentro de la propia empresa y más allá de sus fronteras. Más aun, es muy común que software de terceros se integre como parte del producto propio.

Para la mayoría de los productos y líneas de producto, únicamente una pequeña parte<sup>1</sup> (5%-10%) del software es diferenciador. Esta pequeña parte es la que proporciona valor añadido sobre el producto ofrecido por otras empresas competidoras. El resto del software es más o menos común en el dominio, o incluso entre diversos dominios. El desarrollo de software eficiente y efectivo únicamente se concentra en la producción de módulos diferenciadores. El software como mercancía (la parte común) puede y debería ser adquirido a terceros, implicando desarrollo distribuido y software de terceros como COTS (*Commercial Off The Shelf*) o software libre. Como la Ingeniería de Dominio está más influenciada por el software puede necesitar diversos caminos para ajustarse a una aplicación específica.

La figura 3 muestra un esquema confrontando la tecnologías contra las decisiones empresariales de construir o adquirir software. Las dos esquinas marcadas en la figura deberían ser evitadas a la hora de desarrollar tecnología. La esquina superior derecha debería ser evitada a cualquier precio ya que implicaría regalar el valor añadido propio a

las empresas competidoras. La esquina inferior izquierda también debería ser evitada para ahorrar recursos de desarrollo debido a que la parte común de la tecnología puede ser obtenida de manera mucho más barata comprándola en vez de desarrollándola.

Un desarrollo software saludable está caracterizado por hallarse en el área central de la figura. El software diferenciador se desarrolla dentro de la propia organización y el software que da funcionalidades comunes se adquiere en el mercado o se dispone de él sin coste (software libre).

Cualquier software, incluyendo el desarrollo de líneas de producto, se desplaza de arriba hacia abajo a lo largo de la figura 3. El software comienza siendo diferenciador para alguna de las partes implicadas. Más adelante, en un momento dado ese software no proporciona el suficiente valor añadido a los productos y pasa a la consideración de "básico para el negocio". En etapas posteriores el software llega a convertirse en *commodity* (mercadería sin valor diferencial).

El desarrollo coherente de software también se caracteriza por moverse de izquierda a derecha al mismo tiempo, desde desarrollo interno a colaboraciones abiertas entre empresas. Para evitar alcanzar las situaciones indeseadas que se han descrito es necesario realizar estos movimientos de apertura al ritmo adecuado. Un ejemplo de este movimiento es el caso del software DVTk de Phillips Healthcare<sup>2</sup> [13] que evolucionó en varias etapas desde software propietario y diferenciador a software libre y *commodity* entre 1994 a 2002.

Cada compañía necesita analizar el software que desarrolla con respecto a la figura 3 con el objeto de conocer cuándo cambiar la manera en la que colabora con otras compañías, esto es de especial importancia para las compañías que desarrollan líneas de producto. Debido a la larga vida de una línea de producto, hay partes de la misma que constantemente se convierten en *commodity*, mientras que la propia línea de producto ha de mantenerse viva y próspera. Esto requiere un esfuerzo adicional durante la fase de planificación y definición de recursos. Una complicación adicional es el hecho de que las compañías tienen un control limitado sobre el software libre que utilizan en sus desarrollos.

### 4. Software libre para líneas de producto

Debido al tamaño del desarrollo, las organizaciones implicadas en desarrollos de líneas de producto suelen optar por el desarrollo distribuido que tan eficientemente ha funcionado en las comunidades de desarrollo de software libre. Además, a la hora de reutilizar software es inteligente no ignorar la gran

cantidad de software libre disponible. Dentro del proyecto COSI [5] experimentamos con casos de estudio en los que se aprovechaba el software libre de 5 maneras diferentes:

- 1) Adopción de las prácticas de desarrollo dentro del desarrollo de la línea de producto (desarrollo interno).
- 2) Uso de herramientas de software libre en el desarrollo de la línea de producto. Ésta es la forma más sencilla de usar software libre en líneas de producto.
- 3) Uso de componentes de software libre en los productos de la línea de producto. Esto implica una mayor participación y planificación.
- 4) Liberación de productos de la línea de productos. Productos o líneas completas de producto que fueron inicialmente desarrollados de manera interna se liberan en comunidades de software libre.
- 5) Establecimiento de una relación simbiótica. Desarrollando la línea de producto usando los recursos de una comunidad de software libre.

## 4.1. Fuente interna

La *fente interna* es una manera de aprovechar las ventajas del desarrollo distribuido de la manera en que lo hace el software libre, pero con el deseo de evitar problemas en la planificación, la propiedad y el control del proyecto. Diversas compañías han adoptado modelos de desarrollo basados en *fente interna* [9].

En el modelo de desarrollo de fuente interna un conjunto de equipos colaboran en un ecosistema cooperativo. De manera similar al desarrollo de software libre el desarrollo de fuente interna aplica un modelo de cooperación abierto y concurrente. Este modelo implica la propiedad y el control distribuido, publicaciones tempranas y frecuentes de las aplicaciones desarrolladas, y canales eficientes y abiertos para realimentación por parte de todos los grupos implicados.

Se hace uso de los mecanismos de organización que ya están implantados en la empresa, por ejemplo los designados para la gestión de conflictos, definición de tareas, etc. El modelo de fuente interna proporciona flexibilidad a la hora de gestionar las colaboraciones (establecerlas, cancelarlas y modificarlas) y también es muy útil a la hora de coordinar equipos y definir sus prioridades a través de fronteras organizacionales y geográficas. Las compañías pueden usar este modelo como un paso intermedio para la integración de software libre en sus productos.

El modelo de desarrollo de fuente interna es el establecido dentro de Phillips Healthcare [19]. La compañía entrega una línea de producto que implica un amplio conjunto de productos de imagen médica a sus clientes (generalmente hospitales) en varias modali-

dades. El grupo de Ingeniería de Dominio proporciona una plataforma compuesta de un conjunto de componentes reusables y reconfigurables basados en una arquitectura común. Este grupo abordó el problema de convertirse en un cuello de botella de los diferentes grupos de Ingeniería de Aplicación. Las entregas del grupo de Ingeniería de Dominio se planifican en un proceso con gran cantidad de personas implicadas, por lo que el calendario de entregas no puede satisfacer siempre la planificación de todos los grupos de Ingeniería de Aplicación. Las dinámicas del mercado pueden forzar a cambios en la planificación de ciertos grupos de productos y sin embargo el proceso de planificación del dominio carece de la flexibilidad para adaptarse a estos cambios.

El desarrollo de *fente interna* está basado en elementos del desarrollo de software libre, apoyados por un entorno web de desarrollo abierto y colaborativo. Desacopla la Ingeniería de Aplicación de la Ingeniería de Dominio ya que cada grupo de Ingeniería de Aplicación puede decidir por sí mismo sobre la plataforma:

- *Usarla tal cual.* Esperar a la siguiente entrega de la plataforma.
- *Contribuir con parches.* Obtener una versión anterior de un componente y aplicar parches de manera opcional.
- *Trabajar juntos como un equipo virtual.* Asumir la responsabilidad del desarrollo para los componentes de dominio cruciales para el grupo.

El principio más importante del desarrollo de fuente interna es el de contar con un acceso sencillo a toda la información de la línea de producto. Del mismo modo que en el caso del software libre la política deberá ser publicar rápido y a menudo para favorecer que los flujos de información entre Ingeniería de Dominio y Aplicación sean lo más fluidos posible. Los equipos de Ingeniería de Dominio poseen y desarrollan los componentes de dominio. A un desarrollador de aplicaciones se le permite modificar componentes de dominio, pero es responsable de dicho cambio. El cambio que se realiza podrá ser ofrecido "de vuelta" a los equipos de Ingeniería de Dominio. El equipo encargado de la gestión de la plataforma podrá aceptar el parche como parte de la propia plataforma, haciéndose responsable a partir de ese momento de dicho cambio.

Como consecuencia de esto, la implicación de los equipos de Ingeniería de Aplicación en lo que respecta al dominio se mejora. Esto tiene como efecto colateral que la plataforma se usa de manera más extensa que antes y se ajusta mejor a las necesidades de los grupos de Ingeniería de Aplicación. La metodología de fuente interna permitió la publicación de nuevos productos desde 2005 y la reducción

del tiempo de comercialización en al menos 3 meses.

Sin embargo este modelo no ayuda a introducir software de terceros en la línea de producto. La fase de colaboración se limita a la propia empresa. El control del software es en este caso muy claro y el desarrollo distribuido y el mantenimiento se mejora sobre la situación original. La metodología de fuente interna proporciona métodos de colaboración para el desarrollo de software sobre métodos tradicionales.

## 4.2. Usando herramientas de software libre

Esta es la forma más sencilla de introducir software libre en líneas de producto. Como el software libre no aparece en el producto final, es muy sencillo para el usuario cumplir con las reglas de la comunidad de software libre. Muchas herramientas están ya disponibles como software libre, aunque un factor en contra es que no existen muchas herramientas de software libre específicamente desarrolladas para la gestión de líneas de producto. Una lista de este tipo de herramientas puede encontrarse en las páginas de distribuidores de software para líneas de producto<sup>3</sup>. La mayoría de estas herramientas no son software libre. Algunas herramientas que se utilizan en el proyecto COSI para el desarrollo de líneas de producto y que son software libre son:

- *Stylebase para Eclipse:* Herramientas de software libre que sirven para compartir y reutilizar conocimiento sobre arquitecturas software.<sup>4</sup>
- *Subversion:* Proporciona gestión de versiones.<sup>5</sup>
- *Semantic MediaWiki:* Apoya el desarrollo colaborativo de documentos dentro de un entorno de desarrollo colaborativo.<sup>6</sup>

## 4.3. Usando componentes de software libre

El uso de componentes de software libre en líneas de producto no es un principio muy diferente al de usar cualquier otro software de terceros (COTS) en la línea de producto. Sin embargo existen diversos puntos sobre los que es necesario prestar especial atención. La planificación de la introducción de software de terceros en una línea de producto siempre se ve afectada por el hecho de que el ritmo de desarrollo de este software no está controlado por la propia compañía que lo utiliza: nuevas versiones de la aplicación que mejoran versiones anteriores de modos que no están controlados por la empresa que lo utiliza. Estas mejoras influyen en la línea de producto, ya que partes de este software de terceros se usan dentro de la Ingeniería de Dominio afectado a todos los productos. Por lo tanto esta nueva versión deberá ser incorporada y las partes *propietarias* del software deberán ser adaptadas. A menudo la nueva versión se basa en diferentes mecanismos de arquitectu-

ra e interfaces con respecto a los que están en uso en la línea de producto. Para solventar este problema la empresa que gestiona la línea de producto necesita saber con antelación cómo va a evolucionar el software de terceros. En el caso de COTS esto implica tener un contacto fluido con el proveedor. En el caso de un desarrollo de software libre la empresa deberá implicarse en la propia comunidad de desarrollo, o al menos estar conectada a las listas de correo en las que se discuten todos estos aspectos. Además, si alguno de los desarrolladores de la empresa se implica en la comunidad de desarrollo, la empresa podría influir en el modo en el que este software va a evolucionar, pudiendo incluso llevar a la incorporación de nuevas arquitecturas o estándares en el producto de software libre. En lugar de esperar a nuevas versiones del COTS y a mantener contacto con el proveedor, el software libre se adapta de manera mucho más continua. La posibilidad de obtener versiones tempranas del nuevo software (aunque pudieran aun tener algunos errores) sería de gran ayuda para comprobar que los nuevos cambios son conformes a la arquitectura de la línea de producto, así como para acelerar la incorporación de la primera versión estable. De modo recíproco el envío de informes de fallos y correcciones ayudará a mejorar la calidad del software que se desarrolla por parte de la comunidad de desarrollo, que si bien no es ya un elemento diferenciador para la empresa, es de vital importancia para la línea de producto. Se ha de decidir cuidadosamente qué partes (y qué versiones) de software libre se convertirán en parte de la plataforma, y qué software libre será únicamente usado en ciertas aplicaciones. De hecho, podría convertirse en una estrategia incorporar las últimas versiones de software libre únicamente en algunas aplicaciones *de prueba*, como es el caso del uso de COTS.

Otro problema en el desarrollo de software libre está relacionado con las licencias. Este problema podría afectar al modo en el que el software libre puede ser usado dentro de una línea de producto. Algunas licencias de software libre harían necesario publicar las fuentes del código adicional (el que realmente da valor añadido). Hay opciones técnicas para solventar este problema prestando especial atención a mantener un acoplamiento débil entre el software libre que se integra y el desarrollado de manera interna en la empresa. Sin embargo probablemente acabará siendo necesario que la empresa done alguna parte de su software a la comunidad de desarrollo para demostrar su buena voluntad hacia ella [14].

Es necesario tomar precauciones cuando se incorpora software de distintas fuentes en la línea de producto ya que las licencias particulares de cada componente podrían entrar en conflicto. Este aspecto deberá gestionarse

dentro de la propia organización por las siguientes razones:

- Ignorar las cuestiones relacionadas con las licencias en alguno de los departamentos podría llevar a la necesidad de liberar software de otros departamentos de alto valor competitivo para la empresa.
- El uso de software libre en el desarrollo de aplicación podría indicar un movimiento hacia la mercantilización, lo que a su vez originaría que ese software debería formar parte del software común. Hay que tener en cuenta que esto no es siempre lo que se pretende, ya que el software libre debería ser utilizado únicamente en código específico de aplicación para ciertos productos.
- Nuevas versiones de los componentes basados en software libre podrían provocar que algunos módulos de las aplicaciones quedaran obsoletos.
- El uso de software libre podrían llevar a estándares internos adaptados en la línea de producto. El único modo de gestionar este problema es a través del acuerdo de todas las partes implicadas en el desarrollo de la línea de producto.

En el contexto en el que la empresa se implica en el desarrollo de software libre, ésta podría introducir en la comunidad de desarrollo sus propias capacidades de gestión de la variabilidad. Esto facilitaría una solución para el problema de la gestión de la configuración en las comunidades de software libre. Sin embargo la propia comunidad podría no valorar esta aportación y descartar las capacidades de gestión de la variabilidad. En este caso, la empresa podría asociarse con otra comunidad que sí adopte estas herramientas. Esto podría asegurar el mantenimiento de los mecanismos que gestiona el aplicativo de software libre utilizado y garantizaría la aceptación por parte de la comunidad de software libre. No obstante es necesario tener en cuenta que el control sobre estos mecanismos podría perderse una vez son liberados. La empresa deberá mantener su implicación en la comunidad y por ello invertir en personal o económicamente en la comunidad de desarrollo con el objeto de mantener el estado operativo de las herramientas.

En caso de conflictos con estándares de arquitectura, una opción es usar el conocimiento adquirido para COTS. Crear coberturas propietarias o código de cohesión que conecte los módulos internos de código con los que se encuentran en el mundo del software libre podría implicar actualizaciones demasiado frecuentes debido a que ambos módulos evolucionan de manera independiente [14].

Otra opción está basada en que la empresa adopte la misma solución arquitectónica que se usa dentro de la comunidad de desarrollo de software libre. La principal desventaja es que tanto la Ingeniería de Aplicación como la

de Dominio deberán asumir esta solución arquitectónica, lo que podría suponer un esfuerzo demasiado grande. La empresa ha de asegurar en este caso que los mecanismos de la comunidad de software libre se adaptan a sus necesidades internas, y que a través de ellos pueden gestionar la variabilidad inherente a la línea de producto.

#### 4.4. Liberando los productos de la línea de producto

Muchas empresas han pasado por experiencias muy satisfactorias a la hora de liberar software mercantilizado. A pesar de que en primera instancia pueda parecer que se está regalando la propiedad intelectual de los módulos que se liberan, no suele ser el caso. El software que se encuentra en la parte inferior del diagrama de la **figura 3** no suele contener propiedad intelectual valiosa para la empresa y por esta razón puede ser liberado sin correr un riesgo excesivo. Un recurso más importante es el conocimiento sobre cómo adaptar el software propio a las necesidades de los clientes. Éste es un conocimiento que en ningún caso debería ser liberado. Los clientes son conscientes de que es mucho más sencillo y barato contratar a la empresa para instalar el software y adaptarlo que asumir por sí mismos la configuración completa.

Para Philips Healthcare la primera experiencia liberando software es la suite DVTK<sup>7</sup> [13]. Este software da soporte a la verificación de la conformidad de imágenes médicas con el estándar DICOM. Como los clientes conectarán equipamiento proporcionado por diferentes proveedores es importante que todas las compañías cumplan con el estándar. Sin embargo, DVTK es únicamente software de base sin valor comercial *per se* para la compañía. Por esta razón Philips decidió en 2005 liberarlo lo que permitió compartir los esfuerzos de desarrollo y mantenimiento a una escala mucho mayor. De manera específica llevó a un desarrollo y mantenimiento mucho más rápidos por parte de la compañía, especialmente en aquellas partes de DICOM que son genéricas y no específicas de las necesidades de Philips.

#### 4.5. Relaciones simbióticas

Finalmente, los módulos propios del software de base pueden ser liberados a la comunidad para asegurar que ésta los mantiene y apoya su posterior evolución. Esto lleva a una reducción drástica de los costes de mantenimiento de la empresa, ya que éstos pasa a compartirse con competidores y personas de otros dominios que también necesitan el software. Las mejoras y pruebas masivas del software serán realizadas por la comunidad y la propia compañía apoyará el desarrollo para asegurar que los problemas que aparezcan se solventan.

Esto puede ser particularmente efectivo para herramientas o componentes que han sido

obtenidos por otras vías. La implicación en la comunidad permite a la empresa asegurarse de que todos los requisitos internos se consideran y resuelven, y que el software sigue la línea evolutiva más beneficiosa para la empresa. Por ejemplo, Philips Healthcare es ahora parte de la comunidad Subversion<sup>8</sup>. El objetivo inicial era solventar los problemas que aparecían en la resolución de conflictos en el árbol de versiones. Este enfoque fue de gran éxito a través de la donación de una herramienta específica que solventaba este problema y que se incluyó como parte de la versión 1.6 de Subversion. Mientras tanto Philips se ha convertido en usuario experto y miembro activo de la comunidad Subversion influyendo realmente en la evolución de este proyecto.

### 5. Líneas de producto en desarrollo de software libre

#### 5.1. Prácticas de líneas de producto

Desde la perspectiva del software libre, algunas investigaciones han explorado proyectos de software libre con el objeto de analizar cómo ciertos principios y prácticas comunes de líneas de producto han sido adoptadas por éstos. Por ejemplo, un análisis del núcleo de Linux *"demuestra cómo el núcleo de Linux ha alcanzado algunos de los objetivos que las guías de las Líneas de Producto Software proponen"* [17]. Más allá, van Gurp [7] analizó las prácticas usadas en tres grandes proyectos de software libre (Eclipse, Mozilla y el *kernel* de Linux) desde una visión orientada a sugerir mejoras en el desarrollo de líneas de producto. Este estudio concluye con una sugerencia a los propietarios de líneas de producto software *"hay un conjunto de prácticas que pueden ser encontradas en gran cantidad de proyectos de software libre que han probado su funcionamiento satisfactorio al menos en este contexto"*.

Chastek et al. [4] han investigado los modelos de desarrollo de líneas de producto desde una visión orientada a analizar cómo han adoptado este modelo proyectos específicos de software libre. Han usado un *framework* para pruebas de software de líneas de producto<sup>9</sup> desarrollado por el Instituto de Ingeniería de la Universidad Carnegie Mellon<sup>®</sup> (SEI). En sus análisis del proyecto Eclipse y su comunidad, demostraron que *"Eclipse ha conseguido alcanzar un equilibrio estable entre la dirección planificada y las contribuciones individuales; también comparte el reto de la Ingeniería de Líneas de Producto de gestionar de manera efectiva la comunicación entre aquellos que desarrollan recursos clave y los que los utilizan"*. Resumiendo, a pesar de que algunos proyectos de software libre pueden ser analizados como una línea de producto (por ejemplo Eclipse), los principios y prácticas de línea de Producto no se usan de manera generalizada por parte de las comunidades de desarrollo de software libre.

#### 5.2. Arquitectura y variabilidad

Un recurso muy importante en las líneas de producto es la arquitectura de la plataforma, que define estándares para toda la línea de producto. Todos los sistemas han de ser conformes a esta arquitectura para asegurar el uso efectivo de la plataforma y el alcance de los objetivos de las líneas de producto para reducir el esfuerzo de desarrollo. El problema principal es que la arquitectura a menudo no está correctamente definida en las comunidades de software libre y por esta razón la conformidad con la misma es difícil de comprobar [8].

Las comunidades de software libre tienen a menudo otros mecanismos para gestionar la variabilidad muy diferentes de los utilizados dentro de una línea de producto. En muchos casos el software libre nunca fue concebido para ser usado en líneas de producto, y los problemas relacionados con los procesos de configuración compleja son abundantes dentro del mundo del software libre. Los puntos de variación y las variantes están (casi) ausentes en las metodologías de desarrollo de software libre. La variación espacial y temporal no se diferencia de manera clara. La variación se modela tradicionalmente a través del código o de directivas de compilación, lo cual no resulta muy efectivo para las líneas de producto.

No obstante las comunidades de software libre han desarrollado métodos efectivos de tratar con la configuración interna y la construcción del producto final.

#### 5.3. Dos procesos

Sin embargo dentro de las comunidades de software libre no existe una distinción clara entre la Ingeniería de Dominio y la Ingeniería de Aplicación, aunque ambas actividades se realizan dentro de las comunidades de software libre. Un grupo esencial de desarrolladores experimentados se ocupan de las actividades que podrían englobarse dentro de la Ingeniería de Dominio. El software principal que se usa en toda el aplicativo es desarrollado por estos desarrolladores principales. Los desarrolladores con menos experiencia o menos activos trabajan a menudo en aplicaciones específicas. Sin embargo, del mismo modo que en la Ingeniería de Líneas de Producto, la funcionalidad principal puede emerger también de estos desarrolladores. Promocionar un recurso desde una aplicación hacia el propio dominio es relativamente sencillo en el caso de desarrollo de software libre, y es también una de las ventajas del modelo de desarrollo de fuente interna.

El ámbito del software libre no está a menudo correctamente delimitado. Esto lo hace más flexible y amplía sus posibilidades de utilización. Sin embargo la falta de una arquitectura correctamente definida y el modelo de variabilidad utilizado hace difícil aplicar el soft-

ware en un entorno específico. Esto lleva a conflictos a la hora de decidir qué mecanismos se permiten y cuáles no, reduciendo así la consistencia de los diferentes módulos.

### 6. Conclusiones

Este artículo ha analizado el uso de software libre en líneas de producto. El modelo de desarrollo de software libre es inherentemente un modelo distribuido que constituye un modo muy atractivo y probado de construir software de gran calidad. Las empresas pueden usar las comunidades de software libre para producir y mantener software de base (sin alto valor diferenciador). Esto libera recursos de las compañías para producir software que sí tenga un alto valor específico.

Sin embargo la introducción de software libre en este ámbito no está libre de problemas, y aunque se solventen es necesario mantener ciertos recursos involucrados en la comunidad de software libre. Como el control y la propiedad del software no quedan (en la mayoría de los casos) en la empresa que lo usa, ésta ha de continuar invirtiendo recursos para aprovechar las ventajas del software libre. La empresa ha de considerar dentro de sus procesos de planificación cómo implicarse y como monitorizar la evolución del software libre que usa en sus líneas de producto, así como cuándo y cómo se introducen nuevas versiones. Deben tenerse también en cuenta los problemas legales que podrían aparecer si el grupo de Ingeniería de Dominio no tiene conocimiento de los recursos de la línea de producto que son software libre, así como de sus licencias. Problemas similares pueden aparecer si el equipo de Ingeniería de Aplicación acopla demasiado el código interno que aporta valor añadido a los módulos de software libre.

Podrían aparecer problemas si la arquitectura de los módulos de software libre es incompatible con alguno de los productos de la línea de producto. La empresa podría utilizar el poder adquirido a través de su implicación en la comunidad para proponer y llevar a cabo los cambios que adapten esta arquitectura a la suya o incorporar código que recubra estas características incompatibles.

Las empresas que usan líneas de producto cuentan a menudo con una gran organización distribuida. El desarrollo distribuido genera gran cantidad de problemas de coordinación, algunos de los cuales son solventados por el modelo de desarrollo de software libre. Con el objeto de solventar algunas de las desventajas del uso de software libre, puede utilizarse el modelo de desarrollo de fuente interna. Este último soluciona el problema del desarrollo distribuido, pero no ayuda a compartir el esfuerzo que se da cuando componentes de software libre se integran como parte de la línea de producto.

## Referencias

- [1] F. Bachmann, M. Goedicke, J. Leite, R. Nord, K. Pohl, B. Ramesh, A. Vilbig. "A Meta-Model for Representing Variability in Product Family Development", *Proceedings of the 5th International Workshop on Product Family Engineering (PFE-5)*, Siena, Italy, 2003, pp. 66-80.
- [2] J. Bosch, G. Florijn, D. Greefhorst, J. Kuusela, H. Obbink, K. Pohl. "Variability Issues in Software Product Lines". *Proceedings of the 4th International Workshop in Product Family Engineering (PFE-4)*, Bilbao, Spain, October 3-5, 2001, Springer, Berlin Heidelberg New York, LNCS 2290, 2002, pp. 13-21.
- [3] Jan Bosch. "The Challenges of Broadening the scope of Software Product Families". *Communications of the ACM*, December 2006, pp. 41-44.
- [4] G.J. Chastek, J.D. McGregor, L.M. Northrop. "Observations from Viewing Eclipse as a Product Line". En F. van der Linden, B. Lundell (Eds.) *OSSPL07 Asia, September 10, 2007*, Kyoto, Japan <<http://itea-cosi.org/modules/wikimod/index.php?page=OssPlas07>>.
- [5] COSI. Co-development using inner & Open source in Software Intensive systems, ITEA project 2005-2008, <<http://itea-cosi.org/> and <http://www.friprog.no/Laer-mer/Prosjekter/COSI-Library-of-Assets>>.
- [6] B. Fitzgerald. "The Transformation of open source Software". *MIS Quarterly*, Vol. 30, No. 3, 2006 pp. 587-598.
- [7] J. van Gorp. "OSS Product Family Engineering". *First International Workshop on open source Software and Product Lines*, Maryland, 2006 <[http://www.sei.cmu.edu/splc2006/Gorp\\_paper.pdf](http://www.sei.cmu.edu/splc2006/Gorp_paper.pdf)>.
- [8] I. Hammouda, T. Mikkonen. Open source Contributions as Platform Specialization Units. En F. van der Linden, B. Lundell, *Proceedings on the*

*Second International Workshop on OSSPL07 open source Software and Product Lines 2007, June 14, 2007*, Limerick, Ireland <<http://www.itea-cosi.org/modules/wikimod/index.php?page=OssPlas07>> paper #3>.

[9] A.A. Jilderda. *Inner Source Software Engineering at MIP fostering a meritocracy of peers*. Research Report, Philips Research, The Netherlands, 2004.

[10] K. Kang, S. Cohen, J.A. Hess, W.E. Novak, S.A. Peterson. "Feature-Oriented Domain Analysis (FODA) Feasibility Study", Technical Report, Software Engineering Institute, Carnegie-Mellon University, 1990.

[11] F. van der Linden, K. Schmid, E. Rommes. "Software Product Lines in Action", Springer Verlag, 2007

[12] F. van der Linden, B. Lundell. *Proceedings on the Third International Workshop on open source Software and Product Lines: OSSPL07 Asia, September 10, 2007*, Kyoto, Japan. <<http://itea-cosi.org/modules/wikimod/index.php?page=OssPlas07>>.

[13] F. van der Linden, B. Lundell, P. Marttiin. "Commodification of Industrial Software: A Case for Open Source", pendiente de publicar *IEEE Software July-August 2009*.

[14] J. Merilinna, M. Matinlassi. "Product Family Approach for Integration of In-house Software and open source Components". En F. van der Linden, B. Lundell, *Proceedings on the Second International Workshop on OSSPL07 open source Software and Product Lines 2007, June 14 2007*, Limerick, Ireland <<http://www.itea-cosi.org/modules/wikimod/index.php?page=OssPlas07>>.

[15] B. Perens. "The emerging economic paradigm of open source". *First Monday*, Vol. 10, No. 10, 2005, <<http://www.firstmonday.org/issues/>

[special10\\_10/perens/index.html](http://www.sei.cmu.edu/productlines/frame-work.html)>.

[16] K. Pohl, G. Böckle, F. van der Linden. "Software Product Line Engineering: Foundations, Principles, and Techniques". Springer, 2005. ISBN-10: 3-540-24372-0.

[17] J. Sincero, H. Schirmeier, W. Schröder-Preikschat, O. Spinczyk. "Is The Linux Kernel a Software Product Line?". En F. van der Linden, B. Lundell (Eds.), *Proceedings on the Third International Workshop on open source Software and Product Lines: OSSPL07 Asia, September 10, 2007*, Kyoto, Japan <<http://itea-cosi.org/modules/wikimod/index.php?page=OssPlas07>>.

[18] A. Stellman, J. Greene. *Beautiful Teams*. O'Reilly 2009 pp. 103-111. ISBN: 0596518021.

[19] J. Wesselius. "The Bazaar inside the Cathedral: Business Models for Internal Markets". *IEEE Software Vol. 25, No. 3*, May/June 2008 pp. 60-66.

## Notas

<sup>1</sup> TPPT Nuestra experiencia en la industria europea del software se alinea con la vision de Perens [15] quien estima que quizás el 90% del software en cualquier empresa no es diferenciador

<sup>2</sup> DVTK (Dicom Validation Toolkit), software libre que soporta el intercambio de imágenes médicas <<http://www.dvdk.org/>>.

<sup>3</sup> <<http://www.softwareproductlines.com/resources/vendors.html>>.

<sup>4</sup> <<http://stylebase.tigris.org>>.

<sup>5</sup> <<http://subversion.tigris.org>>.

<sup>6</sup> <[http://semantic-mediawiki.org/wiki/Semantic\\_MediaWiki](http://semantic-mediawiki.org/wiki/Semantic_MediaWiki)>.

<sup>7</sup> <<http://www.dvdk.org/>>.

<sup>8</sup> <<http://subversion.tigris.org>>.

<sup>9</sup> <<http://www.sei.cmu.edu/productlines/frame-work.html>>.

JORNADAS  
CIENTÍFICO  
TÉCNICAS  
EN SERVICIOS  
WEB Y SOA

Madrid, 30 de septiembre a 1 de octubre

# JSWEB'09

### Patrocinadores

#### Oro:



#### Plata:



### Objetivo

Esta quinta edición de las jornadas persigue consolidar el éxito de las JSWEB de ediciones anteriores, estableciéndose como un punto de referencia de profesionales, empresas e investigadores interesados en el uso y la adopción de servicios. Si bien las jornadas comenzaron centrándose en tecnología de Servicios Web, en la actualidad su ámbito se ha extendido al de la Ciencia e Ingeniería de Servicios, incluyendo también aspectos de tecnologías y plataformas (Servicios Web, Servicios Web Semánticos, Servicios Grid, etc.) así como Arquitecturas Orientadas a Servicios (SOA).

**Fecha de recepción de contribuciones: 15 de Junio de 2009**

Contacto: [jsweb09@jsweb.es](mailto:jsweb09@jsweb.es) - Web: <http://www.jsweb.es>

### Organizado por:



### Colaboradores:



Jan Henrik Ziesing  
Fraunhofer FOKUS, Centro de Competencia  
de Qualipso de Berlín (Alemania)

<jan.ziesing@focus.fraunhofer.de>

# Abordar las necesidades de la industria en Software Libre

**Traducción:** Miguel Angel Tinte García (GSyC/LibreSoft, Universidad Rey Juan Carlos)

El uso y desarrollo de software libre se han incrementado significativamente en los últimos años. Tomando como punto de partida la investigación y las actividades dirigidas por la comunidad, un nuevo segmento de negocio ha sido desarrollado recientemente alrededor del software libre. Mientras algunas de las compañías más importantes en el mundo del software libre (por ej. RedHat) hasta ahora se han convertido en los principales agentes del mercado e incluso algunas de las mayores compañías de software (por ej. IBM, Sun...) han desarrollado sus propios productos y estrategias en software libre, éste ha sido hasta el momento un punto estratégico especialmente para las pequeñas y medianas empresas. Debido a la participación de empresas orientadas a obtener beneficios económicos, podemos observar una profesionalización en los procesos de desarrollo y también en términos de mantenimiento de recursos y continuidad [1][2][3].

Aunque la profesionalización e industrialización del desarrollo de software libre es un fenómeno ya visible, existen aún prejuicios por parte de los usuarios y problemas para las empresas que impiden un uso aún más amplio y una mayor variedad de ofertas [4].

La agencia de innovación TSB Berlin GmbH ha publicado recientemente un estudio sobre el sector del software libre en Berlín que ofrece una visión general de la situación actual del mercado alemán en la región de la capital, mostrando las debilidades y las necesidades de las empresas y los agentes activos, así como el potencial de la región [4].

Según el estudio, las principales razones para el uso del denominado Software Libre por parte de la industria de Berlín son las ventajas económicas, las preferencias personales y razones de calidad, que son las tres mencionadas por más del 60% de los agentes industriales y claramente superan otras motivaciones de organización y de imagen, que son sólo mencionadas por menos del 30% (ver **figura 1**).

Las dos limitaciones más importantes para el éxito del software libre en el mercado (ambas mencionadas por cerca del 60% de los agentes más importantes relacionados con el software libre en Berlín) son el insuficiente apoyo político y los prejuicios existentes respecto a la calidad y sostenibilidad del software

**Resumen:** *el software libre está, además de cosechando un gran éxito, atacando los viejos prejuicios y las tradicionales dificultades del mercado. Para profundizar en estos temas, Fraunhofer Fokus está fusionando los resultados del proyecto europeo Qualipso con su demostrado modelo teórico sobre interoperabilidad en el sector público, orientándolo a la industria de las tecnologías de la información. De esta manera, podrá transferir el éxito obtenido y los modelos de negocio teóricos de interoperabilidad al mundo del software libre en la región de Berlín capital.*

**Palabras clave:** *interoperabilidad, necesidades de la industria, Qualipso, software libre.*

## Autor

**Jan Henrik Ziesing** trabaja como científico e investigador para el Instituto Fraunhofer en sistemas de comunicación abiertos en Berlín. Con una amplia experiencia en la transferencia de investigación aplicada, Jan se ha incorporado recientemente a las actividades del Centro de Competencia del proyecto Qualipso en Berlín.

libre (ver **figura 1**). Además, otros limitadores menos mencionados serían la dificultad de competir contra los agentes dominantes en el mercado, la falta de modelos de negocio y financiación (algo por encima del 40%), así como las cuestiones legales (casi el 25%).

El sector del software libre en Berlín se caracteriza, por un lado, por la alta demanda de soluciones desarrolladas con software libre y una fuerte relación entre software libre e I+D. Por otro lado, la industria del software libre en Berlín se encuentra muy dispersa y no suficientemente integrada en una red principal especializada en software libre [4].

Analizando estas fortalezas, debilidades, oportunidades y amenazas para el desarrollo de la industria del software libre en la región de Berlín capital es reseñable que las oportunidades más importantes para la industria coinciden con las principales debilidades y las mayores fortalezas con las amenazas (ver **figura 2**). Una estrategia razonable sería por lo tanto abordar las amenazas cuando sea posible y fortalecer la industria del software libre en particular cuando se encuentren las oportunidades.

Para la región de Berlín, este esfuerzo implica en especial abordar los prejuicios sobre la calidad, el fortalecimiento de la integración en la cadena de valor y el fomento de la transferencia de conocimientos de I+D para aplicaciones industriales.

Mejorar la percepción de la calidad requiere no sólo usar sofisticadas herramientas de

desarrollo orientadas a la calidad y a los procesos, sino también realizar pruebas del producto y su evaluación. Las evaluaciones de calidad deberían hacerse visibles mediante el marketing y la comunicación. Un paso importante hacia la mejora de la transferencia de conocimientos y la integración en la cadena de valor podría lograrse con una mejor conexión en red en la región de Berlín capital, que hasta ahora no ha tenido una estrategia clara ni ha sido suficientemente establecida.

Estos pasos, necesarios para el fortalecimiento de la industria del software libre en la región de Berlín, coinciden con el enfoque del centro del proyecto europeo Qualipso, que recientemente ha establecido centros de competencia de software libre en Madrid (España), Sao Carlos (Brasil) y Berlín (Alemania) y tiene previsto establecer más adelante nuevos Centros de Competencia en Italia, China, Polonia y Francia.

Estos centros de competencia prestarán servicios derivados de los resultados del proyecto Qualipso, que están dedicados a mejorar la calidad del software libre en los diferentes aspectos. Estos comprenden las herramientas y las mejores prácticas para una profesionalización e industrialización del proceso de desarrollo, pruebas de productos y etiquetado, medidas de la interoperabilidad, y evaluación jurídica, así como certificación y formación.

El segundo objetivo de los Centros de Competencia Qualipso es construir redes regionales. Los Centros de Competencia están habilitados para conectarse a través de la Red

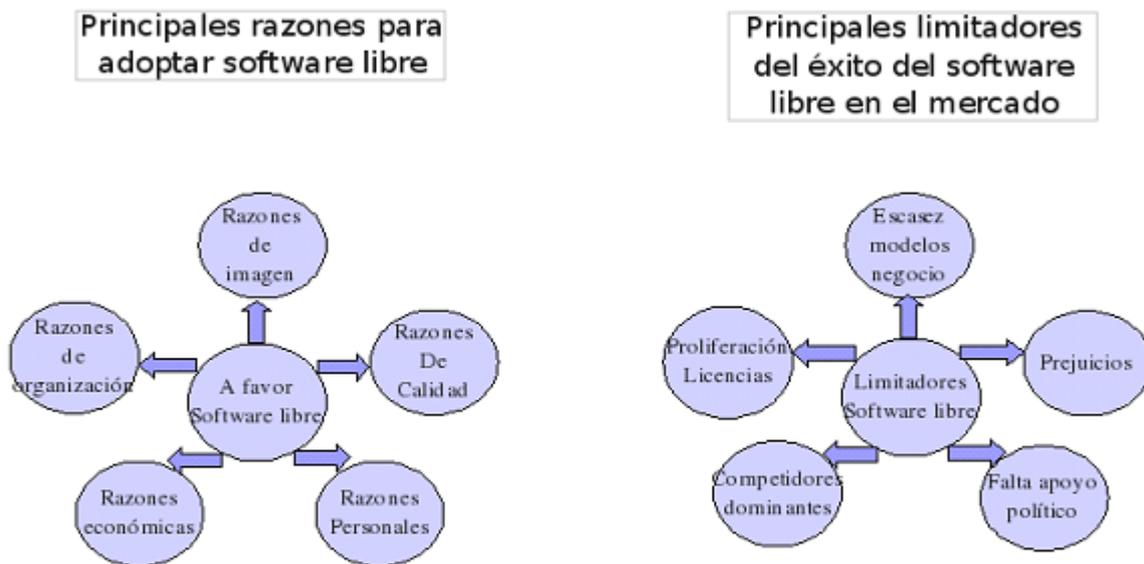


Figura 1. Principales razones y limitadores del software libre en la región de Berlín.

Qualipso, por lo que pueden ofrecer soporte a redes, apoyo y capacidad de intercambiar buenas prácticas no sólo en su región sino en todo el mundo.

Fraunhofer FOKUS (Fraunhofer Institute for Open Communication Systems), como impulsora del Centro de Competencia de Qualipso en Berlín, ha adquirido una amplia

experiencia en los últimos años en soluciones informáticas para los sectores públicos. Un tema importante ha sido y sigue siendo el paisaje heterogéneo existente en las tecnologías informáticas, lo que dificulta el intercambio de datos entre diferentes aplicaciones o entidades públicas.

Las aplicaciones utilizadas en el sector públi-

co en Alemania están, en general, altamente especializadas para su aplicación en su dominio y para ofrecer todas las funciones y servicios para este dominio en una única herramienta. La interoperabilidad con otras aplicaciones del mismo o de un dominio diferente no se consideró importante. Para el intercambio de datos entre diferentes aplicaciones era bastante común imprimir un documento en

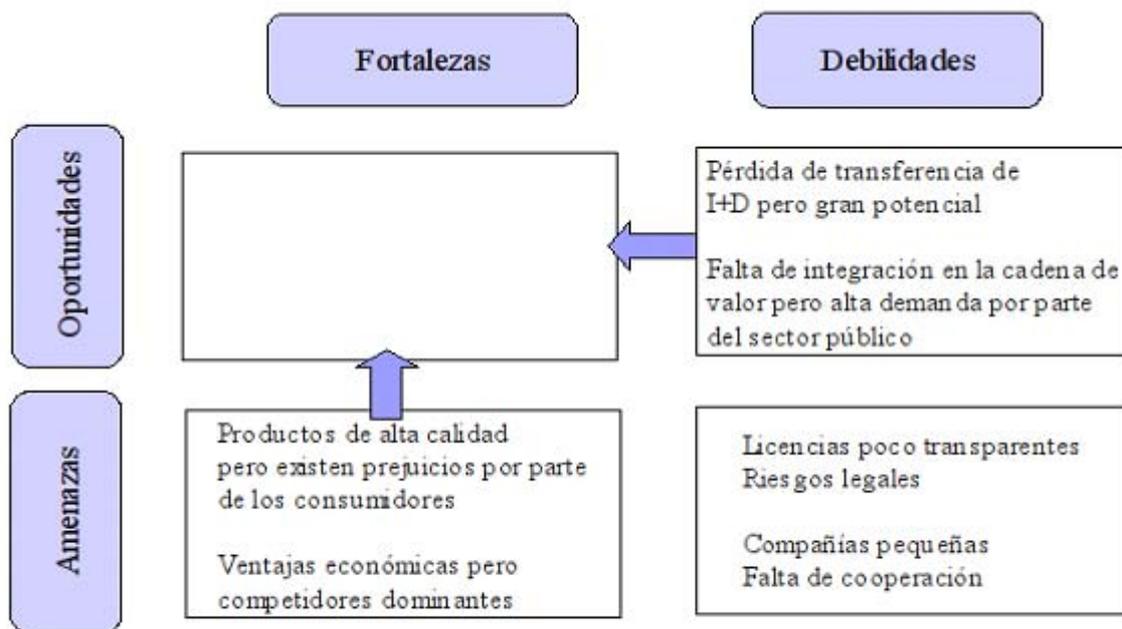


Figura 2. Análisis de la matriz DAFO.

una aplicación y escribirlo de nuevo a mano en otra aplicación.

Hoy en día, se hace evidente una tendencia hacia un eficiente y ergonómico proceso de administración electrónica basado en la conexión e integración de diferentes ámbitos de aplicación. Fraunhofer FOKUS se convirtió en uno de los principales impulsores de esta tendencia en Alemania mostrando en su Laboratorio de Administración Electrónica [5] las posibilidades y ventajas de una integración y orquestación de los diferentes dominios específicos de las aplicaciones software. El Laboratorio Administración Electrónica se dedica a investigar sobre interoperabilidad, a conectar a los principales actores de los sectores públicos y a comunicar sus logros.

Otra tendencia en la administración electrónica observada por Fraunhofer FOKUS es el desarrollo de servicios orientados a las arquitecturas en el sector público. Esto es especialmente interesante para la industria del software libre. La mayoría de las herramientas de software libre ofrecen determinados servicios para la solución de un cierto problema en numerosos ámbitos de aplicación, en lugar de ofrecer un conjunto de servicios integrados en

una aplicación de un dominio específico. Mediante la transferencia por parte del Fraunhofer FOKUS de su interoperabilidad y sus conocimientos sobre el sector, el Centro de Competencia de Qualipso podrá hacer frente al mercado del software libre en el sector público mediante nuevos escenarios de interoperabilidad. Estos escenarios ayudarán a la industria a desarrollar soluciones específicas de dominio complejo mediante la oferta de mayores facilidades para la integración y orquestación de herramientas interoperables de código abierto.

Además de centrarse en la interoperabilidad de los servicios para software libre, el recientemente inaugurado Centro de Competencia de Qualipso en Berlín fomentará el desarrollo de software libre, ofreciendo una forja de última generación, así como la creación de redes de comunicación y oportunidades para la industria del software libre. El Centro de Competencia podrá, por lo tanto, hacer frente a las necesidades más urgentes de la industria del software libre en la región de Berlín, transfiriendo un enfoque previamente experimentado de Fraunhofer FOKUS al sector del software libre.

**Referencias**

[1] J.-P. Laisné et al. 2020 FLOSS Roadmap, 2008.

[2] D. Kühni. *Kurze betriebswirtschaftliche Analyse von Open Source Software*. (i. GmbH, Hrsg.) Bern, Schweiz, 2006.

[3] J. Lerner, J. Tirole. *Some simple economics of open source*. *Journal of Industrial Economics*, 50 (2), 2002.

[4] D.M. Fornefeld, M. Gasper. *Potenzialanalyse im Technologiefeld Open Source in der Hauptstadtregion Berlin*. (T. I. GmbH, Hrsg.) Berlin, Deutschland, February 2009.

[5] Fraunhofer FOKUS. Fraunhofer FOKUS eGovernment-Lab. <[http://www.fokus.fraunhofer.de/en/fokus\\_testbeds/egov-lab/index.html](http://www.fokus.fraunhofer.de/en/fokus_testbeds/egov-lab/index.html)>.

## XIV Jornadas de Ingeniería del Software y Bases de Datos San Sebastián, 8-11 de Septiembre 2009

### CONFERENCIAS INVITADAS

Jean Bezin, Universidad de Nantes, Francia  
Don Batory, Universidad de Texas en Austin, USA  
Houari Sahraou, Universidad de Montreal, Canada

### TALLERES

ZOCO - Integración de Aplicaciones Web no Desmantelable  
ADIS - Apoyo a la Decisión en Ingeniería del Software  
PRIS - Pruebas en Ingeniería del Software  
PNIS - Procesos de Negocio e Ingeniería de Servicios  
DSDM - Desarrollo de Software Dirigido por Modelos  
WASELF - Autonomic and SELF-adaptive Systems

### TUTORIALES

**Análisis en líneas de productos: avances, desafíos y lecciones aprendidas**

David Benavides, Antonio Ruiz y Pablo Trinidad  
Univ. de Sevilla

**Herramientas Eclipse para el Desarrollo de Software Dirigido por Modelos**

Cristina Vicente y Diego Alonso  
Univ. Politécnica de Cartagena

Evento	Hasta 30-junio	Hasta 31-julio	Tardía
JISBD	350€	390€	430€
Taller inscritos JISBD	100€	115€	130€
Taller no inscritos en JISBD	130€	145€	160€
Tutorial no inscritos en JISBD	50€	50€	50€

### Presidente del Comité de Programa:

Antonio Vallecillo (Univ. Málaga)  
**Coordinadora de Talleres:**  
Coral Calero (Univ. Castilla-La Mancha)  
**Coordinador de Tutoriales:**  
Ernest Teniente (Univ. Polit. Cataluña)  
**Coordinadores de Publicidad:**  
Gentzane Aldekoa (Univ. Mondragón)  
José Raúl Romero (Univ. Córdoba)  
**Coordinadora de la Web:**  
Ana Altuna (Univ. Mondragón)

### Presidente del Comité Organizador:

Goiuria Sagardui (Univ. Mondragón)  
**Coordinador de Demostraciones:**  
Juan de Lara (Univ. Autónoma Madrid)  
**Coordinadora de Divulgación de Trabajos Relevantes ya Publicados:**  
Belén Vela (Univ. Rey Juan Carlos)  
**Coordinadora de actas:**  
Leire Etxeberria (Univ. Mondragón)



<http://www.mondragon.edu/jisbd2009/>

Gabriele Ruffatti

Director de Arquitecturas y Consultoría, Grupo Engineering I+D, Italia

&lt;gabriele.ruffatti@eng.it&gt;

# SpagoWorld, la iniciativa de software libre de Engineering

Traducción: Israel Herraiz Tabernero (Universidad Complutense de Madrid)

## 1. Introducción

Engineering<sup>1</sup> es una empresa informática global, y el mayor operador informático de Italia en el sector servicios, proveedor líder de ofertas completas e integradas en toda la cadena de valor del software: diseño, desarrollo, servicios externalizados, productos y soluciones privativas verticales, consultoría estratégica e informática... ajustadas a los modelos de negocio de nuestros clientes en todos los mercados.

Hace ya más de cinco años desde que Engineering decidiera desarrollar y gestionar directamente proyectos de software libre, en lugar de simplemente colaborar con varias comunidades de software libre, o usar directamente sus resultados. SpagoWorld<sup>2</sup>, la principal iniciativa de software libre de Engineering, constituye un ecosistema que reúne empresas, integradores, vendedores, instituciones y usuarios operando de una manera efectiva para añadir valor al dominio del software libre.

La construcción de un ecosistema de software libre efectivo requiere algunos aspectos clave:

- Colaboración con las comunidades de software libre a nivel internacional.
- Realización de proyectos efectivos en el ámbito de las tecnologías actuales más estratégicas, presentando al mismo tiempo una visión innovadora.
- Éxito comercial.

En la actualidad, SpagoWorld es un ecosistema de software libre basado en proyectos que se enfocan en inteligencia de negocios y dominios de gestión de procesos de negocios (SOA, Service-Oriented Architecture), los cuales han sido identificados por Engineering como los más prometedores en el mundo del software libre. Los proyectos se alojan en la comunidad global OW2<sup>3</sup> y añaden un enfoque original a las capacidades estándar de sus dominios específicos, que se caracteriza por una especial atención a las necesidades de los usuarios, siendo ésta la razón del incremento de su rendimiento comercial.

La iniciativa SpagoWorld no es sólo cooperación tecnológica; su objetivo principal es promocionar las comunidades de los proyectos, a la vez que se asegura la libertad del software desarrollado a tiempo y de las cola-

**Resumen:** este artículo muestra un caso de estudio basado en la experiencia de una empresa informática en el desarrollo y la gestión de proyectos de software libre. En la primera parte, presentamos la compañía y su papel en actividades relacionadas con software libre. Después, el artículo presenta las razones y la estrategia corporativa para entrar en el mundo del software libre, y el modelo de negocio de SpagoWorld, la principal iniciativa sobre software libre de Engineering. Finalmente, se investiga el doble papel que juega en actividades comerciales y creación y gestión de comunidades, con énfasis en la sostenibilidad del modelo.

**Palabras clave:** empresas informáticas, estrategias corporativas, software libre.

### Autor

**Gabriele Ruffatti** es director de la unidad de Arquitecturas y Consultoría en la división de I+D de Engineering. Con más de 27 de años de experiencia en el ámbito de la informática, ha ejercido varios puestos de gestión en diferentes áreas dentro del Grupo Engineering, como por ejemplo el desarrollo de productos complejos y de proyectos para importantes firmas y para el sector público, en procesos de mejora de calidad de software y de aseguramiento de calidad, además de en la definición de soluciones innovadoras. Además, contribuyó al inicio y desarrollo del sistema corporativo de calidad, y ayudó a conseguir las certificaciones CMMI e ISO 9001:2000. En 2004 lanzó la iniciativa de software libre SpagoWorld. Además, Gabriele coordina las actividades de software libre en Engineering. Es miembro del Consejo de SpagoWorld y del consorcio OW2. De 2006 a 2008, participó como profesor asociado, impartiendo clases sobre software libre, en el Departamento de Matemáticas, Física y Ciencias Naturales de la Universidad de Padua, dentro del programa de Máster en Informática.

boraciones con empresas, de manera que se consolida un modelo de negocio diversificado y basado en un enfoque comercial del software libre. Los aspectos más destacados de este enfoque son: desarrollo de soluciones adaptables con énfasis en los requisitos de los usuarios, desarrollo de soluciones a nivel empresarial, soporte a los servicios ofrecidos, atención prestada a las demandas de las comunidades de software libre y la comunidad académica, y, finalmente, todo lo que concierne al crecimiento de un ecosistema que crea nuevo valor para todos sus participantes.

SpagoWorld es un ejemplo efectivo de ecosistema de negocios que actúa con un modelo de negocio específico, basado en el desarrollo y la promoción de soluciones particulares (mediante la venta de soporte a servicios y proyectos de software relacionados) y al mismo tiempo sostiene el sistema entero en un verdadero entorno de *cooperación*<sup>4</sup>. El dominio del software libre es particularmente apropiado para el desarrollo de una estrategia colectiva que incrementa el valor en un contexto que puede ser definido como "ecológico", donde las ganancias no monetarias y difícilmente cuantificables pueden ser más beneficiosas que las monetarias, porque son beneficios a largo plazo en un contexto de sostenibilidad.

## 2. Engineering y el software libre

Engineering lleva a cabo la definición de soluciones con arquitecturas innovadoras, y la realización de proyectos complejos para la administración pública, instituciones financieras y grandes empresas, usando tanto software libre como soluciones privativas.

Después de una primera fase centrada en el apoyo a iniciativas de software libre y la investigación de comunidades, desde el año 2004 Engineering ha definido una estrategia enfocada en el desarrollo, implementación y entrega de soluciones basadas en software libre como una oportunidad tecnológica y de negocio en los servicios que ofrece a sus clientes, siendo sus objetivos principales los siguientes:

- Uso industrial del software libre en las actividades de integración de sistemas, y provisión de un soporte completo a los servicios ofrecidos.
- Contribución activa en varios proyectos de software libre.
- Presencia activa en comunidades internacionales.
- Desarrollo y apoyo de iniciativas de software libre.

En la actualidad, la compañía cree que el software libre es un factor acelerador para la

consecución de objetivos estratégicos críticos en la empresa.

La compañía proporciona a sus clientes su experiencia y conocimiento para la selección, integración, validación y soporte de los mejores componentes de software libre, incluyendo sus propias soluciones, ofreciéndoles los beneficios completos de un mayor ratio de valor monetario, gracias a atractivos esquemas de licenciamiento de software libre, y a la robustez como resultado de un intenso escrutinio del código fuente.

El enfoque sobre el software libre es racional desde un punto de vista ingenieril: analizar las ventajas e inconvenientes, y tomar decisiones adecuadas mejor que ideológicas. Engineering está convencida de que un uso adecuado del software libre puede ayudar a generar altos retornos para sus clientes, a la vez que se mejora la calidad de los sistemas.

### 3. El proceso de liberación de software de Engineering: razones y estrategia

En los últimos 30 años, Engineering ha operado en los mercados de desarrollo de software e integración de sistemas. Hace seis años, la compañía comenzó a pensar que el software libre era útil para los negocios. En aquella época, el software libre no estaba tan extendido en el mercado como lo está ahora, y la cuestión era si el software libre tenía el potencial para proporcionar beneficios en el mercado de integración de sistemas. Para responder a esta cuestión, la compañía analizó las características particulares del software libre, y determinó que, por varias razones, era posible elaborar un modelo de negocio válido para los integradores de sistemas, que podían aprovecharse de estas características.

Resumiendo, la completa ausencia de costes de licencia puede liberar recursos financieros que el cliente puede dedicar a la adquisición de servicios y de soluciones a medida, aspectos fundamentales ambos del negocio de un integrador de sistemas. La disponibilidad del código fuente permite al integrador de sistemas extender su oferta en las áreas de soporte y servicios al mantenimiento de soluciones de software libre. Este mercado potencial tiene una barrera de entrada muy baja, de modo que no existen posiciones dominantes.

Además, hay que tener en cuenta que la principal característica de un profesional informático, que tiene que encontrar la mejor solución para su propio cliente, está representada por su competencia adquirida gracias a la experiencia. Por tanto, es inevitable que la preparación en el dominio del software libre se haga *haciendo*, es decir mediante la gestión y participación activa en proyectos de software libre.

Orazio Viele, gerente de I+D en Engineering, escribió en 2005: *"El software libre representa más una oportunidad que una amenaza para un integrador de sistemas. En la actualidad, no podemos establecer cuál es el valor potencial de mercado estimulado por el software libre. Sin embargo, las características de este fenómeno son tales que podemos predecir un crecimiento progresivo en los próximos años. El reto para un integrador de sistemas consiste en una excelente preparación, porque ésta es la única manera de tomar ventaja de los mejores resultados provocados por esta revolución"*. [1].

Tras examinar estas razones, se hizo necesario definir cómo enfocar esta oportunidad y definir una estrategia para posicionar con éxito a Engineering en este mercado.

Durante estos últimos seis años, Engineering ha mejorado esta estrategia, basándose en los siguientes principios:

- Tomar una posición líder en el mundo del software libre, desarrollando sus propias soluciones y creando un ecosistema alrededor de ellas. Esto diferencia a la compañía de otros integradores de sistemas, que sólo usan soluciones desarrolladas por otros. La iniciativa SpagoWorld es una acción concreta, para llevar a cabo esta estrategia. Todo esto hizo posible que Engineering fuera percibido como un productor de soluciones de software libre, especialmente fuera de Italia.

- Ser parte y contribuir a las comunidades internacionales de software libre, para crear una red colaborativa que enriquezca su oferta al mercado de soluciones y servicios. El ser miembro del consorcio OW2 y las contribuciones a diferentes comunidades son pruebas claras de esta estrategia.

- Seleccionar soluciones de software libre de alta fiabilidad, para ofrecerlas a sus clientes y satisfacer sus demandas. Dos herramientas adoptadas para alcanzar este objetivo son la creación de un centro de competencia dedicado en exclusiva a explorar soluciones, y la definición de una metodología para evaluar software libre.

La estrategia adoptada por Engineering respecto al software libre está centrada en dos aspectos más:

- Desarrollo de *software libre y gratuito*, teniendo en cuenta que Engineering no adopta un esquema de licenciamiento dual, sino que desarrolla y gestiona software liberado únicamente con licencias libres (como la licencia Apache o la GNU GPL/LGPL<sup>5</sup>), evitando cualquier política que asemeje el software libre al modelo privativo<sup>6</sup>.

- Un modelo de negocio *centrado en el proyecto*, donde el proyecto es más importante que las soluciones tecnológicas que se adopten, y donde se explota la capacidad para diseñar y desarrollar sistemas complejos capaces de satisfacer las demandas de los clien-

tes. En este modelo, el uso de software libre se convierte en una herramienta para mejorar las habilidades de diseño y el *know-how* técnico de un integrador de sistemas.

Esta estrategia no se planeó *a priori*, sino que deriva de un análisis de resultados, y de su crecimiento y adaptación en el tiempo. Así, tras algunos años de actividad, podemos decir que hemos vivido la experiencia de integrarnos en un entorno ecológico competitivo, donde el concepto estratégico ha cambiado respecto al previamente conocido. *"Para el desarrollo de una estrategia, normalmente es importante fijar objetivos a corto, medio y largo plazo, y definir un plan, llamado estratégico, a través del cual podamos alcanzar esos objetivos, definiendo los medios más adecuados, y suponiendo que siempre serán insuficientes. Adaptar un enfoque ecológico del valor supone tanto abandonar la parte de certeza que se deriva de la posesión de un plan definido para desarrollar un cierto conjunto de objetivos, como cambiarlos debido a nuestra capacidad de exploración continua de nuestra existencia en el mundo y al sentido que tenemos que asignarle a este hecho. El objetivo es mejorar la calidad de nuestra condición, experimentando todos los posibles potenciales distintos."* [2].

### 4. El modelo de negocio de software libre de Engineering

Engineering considera en su modelo de negocio que en la actualidad el software libre no es simplemente un modelo de desarrollo y distribución de software, sino que está correlacionado con la naturaleza de las comunicaciones, con su evolución en el tiempo junto a los modelos de negocio, y con su estructura ecológica (Véase [3] para una comparación entre ecosistemas de negocios y ecosistemas biológicos).

En la actualidad, Engineering es un agente informático global, aunque es principalmente un integrador de sistemas con características particulares, cuyos resultados se valoran en el mercado como los resultados de una empresa informática con rendimientos efectivos de desarrollo, comenzando por:

- La capacidad "genética" para el diseño.
- Un énfasis importante en la adquisición de conocimientos de primera mano (primero "testea", luego propón a los clientes y finalmente desarrolla), crecientes en el tiempo mediante experiencia y evaluación.
- Independencia de la solución elegida y adoptada.
- Flexibilidad con respecto a las diferentes situaciones, y una actitud natural para adaptar cada solución a distintos contextos, de modo que se generen soluciones y productos con un alto nivel de adaptación al cliente.
- Un enfoque industrial que hace énfasis en la adopción de los resultados en el mundo "real", de una manera efectiva.

Estas características también se aplican al caso del software libre, especialmente en el diseño y desarrollo de soluciones, con una actitud independiente y un enfoque en la utilidad industrial.

Como resultado, el modelo de software libre de Engineering es un modelo *empresarial de software libre*, en lo que respecta al desarrollo de proyectos y soluciones abiertas, y un modelo *profesional de software libre* en lo que respecta a la certificación de competencias y servicios de soporte a soluciones abiertas.

Más en detalle, el modelo de desarrollo de software libre puede calificarse como centrado en el proyecto, porque "el alcance del desarrollo de una solución específica comprende la realización de proyectos de software bajo las demandas de diferentes clientes, que se pueden beneficiar de una solución de software libre, ofreciendo mejores características que una solución propietario en términos de disponibilidad, apertura, modificabilidad, modularidad, integración, adaptabilidad, reusabilidad y escalabilidad. Resumiendo, el proyecto tiene más valor que la solución adoptada" [4].

Según este modelo, las soluciones basadas en software libre hacen posible el desarrollo de proyectos; estos proyectos, de hecho, posibilitan el crecimiento del software libre. Por otro lado, el software libre favorece la consecución de los requisitos de los clientes, de modo que se logra la mejor solución adaptada a sus necesidades; por otro lado también, la comunidad se beneficia de estos nuevos requisitos, nuevo código, testeo, realimentación y contribuciones externas.

En este contexto, el integrador de sistemas es el facilitador que activa las relaciones sinérgicas, y uno de los principales actores, gracias a su capacidad "genética" para trabajar de esta manera.

## 5. La iniciativa SpagoWorld: un poco de historia

La iniciativa SpagoWorld es un ejemplo de cómo Engineering dirige su modelo de negocio de software libre centrado en proyectos.

Surgida en 2004, la iniciativa cuenta ahora con cuatro proyectos principales:

- *SpagoBI*: la plataforma para *business intelligence* (BI).
- *Spagic*: la plataforma empresarial de integración SOA/BPM.
- *Spago4Q*: especialización de SpagoBI para calidad de software.
- *Spago*: una plataforma empresarial Java.

Todos estos proyectos adoptan el mismo modelo de licenciamiento (se distribuyen bajo una licencia GNU LGPL, sin versiones "profesionales" o "enterprise" por las que es nece-

sario pagar), y se alojan en el consorcio OW2, que proporciona un soporte independiente y a largo plazo gracias a una comunidad global.

Los proyectos comparten una visión común, basada en:

- *Soluciones flexibles*: integrando componentes ya existentes y desarrollando nuevos módulos, con un enfoque de integración en la plataforma, de modo que se identifican las soluciones más adecuadas para las necesidades de los usuarios.

- *Nivel empresarial*: las soluciones resultan de la experiencia en proyectos a nivel empresarial, donde las aplicaciones son críticas, y hay que garantizar la funcionalidad y un alto rendimiento.

- *Enfoque en el desarrollo de proyectos*: la capacidad para entender las necesidades reales de los usuarios, además de los requisitos de los diferentes proyectos, convierten a las soluciones de SpagoWorld en la mejor opción para comenzar el desarrollo de un nuevo proyecto de software.

- *Uso comercial*: las soluciones de software libre adoptan un esquema libre de licenciamiento, que permite el uso de las soluciones de SpagoWorld en diferentes categorías de productos y servicios.

- *Servicios de soporte*: cada solución es entregada, bajo demanda, incluyendo una solución completa de servicios de soporte.

- *Enfoque en la comunidad y la investigación*: todos los proyectos tienen un alto compromiso con las necesidades de la comunidad, e incorporan resultados innovadores de investigación.

- *Creación de un ecosistema de valor*: la iniciativa SpagoWorld participa en la creación de un ecosistema de valor añadido para todos los miembros de la comunidad: empresa, desarrolladores, investigadores y usuarios.

## 6. SpagoWorld y el mercado comercial de software libre

Las soluciones SpagoWorld son descargadas desde diferentes países alrededor del mundo. Esto confirma que son conocidas (y probablemente usadas<sup>7</sup>) en un contexto global, sin ceñirse a un mercado geográfico específico.

Aunque el éxito comercial significa otra cosa diferente. Este aspecto debe considerar dos puntos preliminares, uno particular a cada proyecto y otro que tiene que ver con la solución como un todo.

La premisa específica consiste en el hecho de que cada solución tiene que enfrentarse a competidores específicos en un contexto de software libre, donde la reputación otorgada por la comunidad es crucial (por ej., Spago, el *framework* Java, apenas puede competir con otras *frameworks* Java más conocidos;

Spago4Q es una solución nueva y única, sin competidores reales, pero necesita alcanzar una reputación suficiente; Spagic, una solución bien recibida por la comunidad, necesita algo de tiempo para demostrar su eficacia en el uso en casos reales alrededor del mundo; SpagoBI, la más popular, es reconocida sobre todo como un producto software libre).

La premisa general se refiere a diferentes aspectos:

- Engineering, desarrollador de todas las soluciones, es un integrador de sistemas que actúa en los mercados y dominios informáticos, y que está siguiendo una trayectoria internacional todavía sin finalizar. Los competidores de software libre son empresas que actúan en un dominio específico (como Pentaho, JasperSoft y Actuate en el dominio de *business intelligence*, o Intalio y MuleSource para soluciones SOA/BPM) ofreciendo soporte a nivel mundial. Contrariamente a un integrador de sistemas de propósito general, el mercado informático percibe a estas empresas como más eficaces centrándose en un único dominio que es crítico para su éxito, e invirtiendo cantidades significativas en actividades de marketing.

- Especialmente en Europa, los grandes integradores de sistemas que dirigen el mercado informático prefieren no usar las soluciones de otro integrador de manera explícita, para invadir "intrusismos";

- La falta de soporte constante y universal en regiones como Europa, EE.UU., Asia y Latinoamérica no facilita la inserción de las soluciones de SpagoWorld en las listas "top", a pesar de sus funcionales y sus innovadoras características.

No obstante, tanto SpagoBI como Spagic gozan de una buena reputación (su reciente inclusión en Gartner Research [5][6] confirma este punto), y han obtenido pronto ciertos resultados comerciales, de modo que se han convertido en dos activos de software libre del grupo Engineering: Spagic está ganando impulso en el desarrollo de proyectos SOA/BPM para clientes de Engineering, algunos de ellos multinacionales; las actividades de desarrollo de SpagoBI se sostienen gracias a los beneficios que proporcionan las ventas de servicios de formación y soporte, principalmente en Francia y mercados relacionados. Además, los servicios de soporte de SpagoBI se ofrecen tanto en Latinoamérica como en Asia.

## 7. SpagoWorld y la comunidad de software libre: lecciones aprendidas

El primer resultado significativo de Engineering en el software libre proviene de sus actividades en algunos proyectos de investigación en Italia y Europa.

Las colaboraciones en este campo con em-

presas, universidades e institutos de investigación han proporcionado oportunidades para desarrollar soluciones y componentes libres, entre los que podemos mencionar *bxModeller* para el modelado de procesos de negocio, que proviene de los resultados de los proyectos de investigación DISCoRSO, X@Work y TEKNE; las soluciones del proyecto Bricks para bibliotecas de herencia cultural digitales; SeCSE para sistemas centrados en servicios y ETICS, para el desarrollo de software y calidad en entornos *grid*.

Cuando Engineering decidió comenzar a desarrollar y gestionar proyectos de software libre gratuitos a nivel industrial, se tomaron dos importantes decisiones:

- La elección de no hacerlo solos, sino conectados a una comunidad internacional ya existente.
- Ser consecuentes con el hecho de que en las comunidades de software libre los tiempos han cambiado.

De hecho, mientras que la naturaleza del software libre ha cambiado en el tiempo, desde el campo de las infraestructuras hacia el *middleware* primero, y hoy en día hacia un entorno aplicativo, se han creado nuevas comunidades con diferentes connotaciones (por ejemplo, las Fundaciones de Linux, Apache y Eclipse, el consorcio OW2, etc.).

Desde aquellas primeras comunidades de individuos movidos por la *ética hacker* nos hemos desplazado a la tercera generación de comunidades de software libre que aglutinan diferentes entidades legales, compañías federadas, vendedores, clientes, administraciones públicas e individuos. El modelo de negocio de estas meta-organizaciones es colectivo: el núcleo de su valor se orienta hacia el incremento del valor de la organización como un todo, y consiste en la estimulación de la colaboración entre sus miembros para alcanzar diversas metas que son útiles para todos los miembros.

La naturaleza cambiante de estas comunidades está influyendo también en el desarrollo de los actuales proyectos de investigación de Engineering. Algunos ejemplos de estas comunidades son:

- *Qualipso*<sup>8</sup>, una alianza internacional de empresas informáticas, PYMEs, investigadores e instituciones públicas y académicas, cuyo rol consiste en ayudar a la industria y a los gobiernos en el fomento de la innovación y la competitividad con software libre.
- El Grupo de Trabajo sobre Software Libre de NESSI<sup>9</sup> que da soporte a la *Networked European Software and Services Initiative* para construir una plataforma tecnológica europea dedicada al software y a los servicios. Apoya a NESSI en la definición de una estrategia global sobre software libre, enfocada a empresas que quieran implementar o adoptar

software libre, además de a comunidades de software libre que quieran colaborar y participar en NESSI, proporcionando el apoyo, la ayuda y la información necesarios para posicionar al software libre como el canal principal de comunicación de los logros de NESSI, asegurando niveles adecuados de calidad, dependencia y seguridad.

Incluso cuando Engineering estaba buscando una comunidad de referencia para sus proyectos de software libre, eligió una comunidad que prestaba atención especial a los requisitos de los usuarios finales y las empresas del ecosistema, sin ignorar el papel de los individuos y los desarrolladores en la comunidad. La adhesión al consorcio ObjectWeb en 2005 ha representado la elección particular de entrar en una comunidad caracterizada por una identidad precisa que persigue promocionar y afirmar un ecosistema sostenible y durable basado en soluciones software libre. Estas características pueden ayudar a las empresas del consorcio a conseguir sus objetivos de negocio y a las administraciones y a los usuarios a satisfacer sus demandas.

La participación en ObjectWeb también ha representado la oportunidad de compartir su transformación en el consorcio OW2, un nuevo consorcio internacional más fuerte, guiado por una filosofía abierta y que persigue la promoción de soluciones tecnológicas y una nueva manera de hacer negocios. Nacido en enero de 2007 como una consecuencia de la integración de las comunidades europea ObjectWeb y china OrientWare, OW2 es hoy un consorcio industrial independiente dedicado al fomento de un vibrante ecosistema de negocio, que cuenta con más de 100 organizaciones y 6.000 desarrolladores distribuidos en Europa, Asia y América, además de albergar más de cien proyectos tecnológicos.

La adhesión a OW2 ha sido una muestra de la estrategia de software libre de Engineering, cuyo objetivo es compartir proyectos con la comunidad e integrarlos con otras soluciones, buscando de manera continua nuevas oportunidades. El software SpagoWorld está alojado en la forja de OW2, de modo que permite a la comunidad participar usando herramientas específicas (listas de correo, foros, repositorios, área de descargas), y proporcionando una gestión independiente del software publicado. Esta colaboración va más allá de un apoyo notorio al crecimiento de toda la gama completa de software de OW2, y representa la participación en uno de los ejemplos más exitosos de comunidad de software libre de tercera generación.

En la actualidad, Engineering es co-fundador y miembro estratégico del consorcio OW2. Además, es particularmente activo en la vida del consorcio, siendo miembro del Consejo Rector, de todos los consejos de apoyo a los

ecosistemas (Consejo del Ecosistema, Consejo Tecnológico y Oficina de Gestión), y permanece activo en muchas iniciativas (es el líder de la iniciativa de Business Intelligence, y forma parte de las actividades del Capítulo Local Europeo).

Con el tiempo, la colaboración se ha extendido también a otras comunidades. Gracias principalmente al desarrollo de la plataforma Spagic, Engineering participa en la comunidad Eclipse en el desarrollo del proyecto Eclipse STP/Intermediate Model, junto al instituto francés de investigación INRIA, y en la comunidad Apache como contribuidora al proyecto ServiceMix.

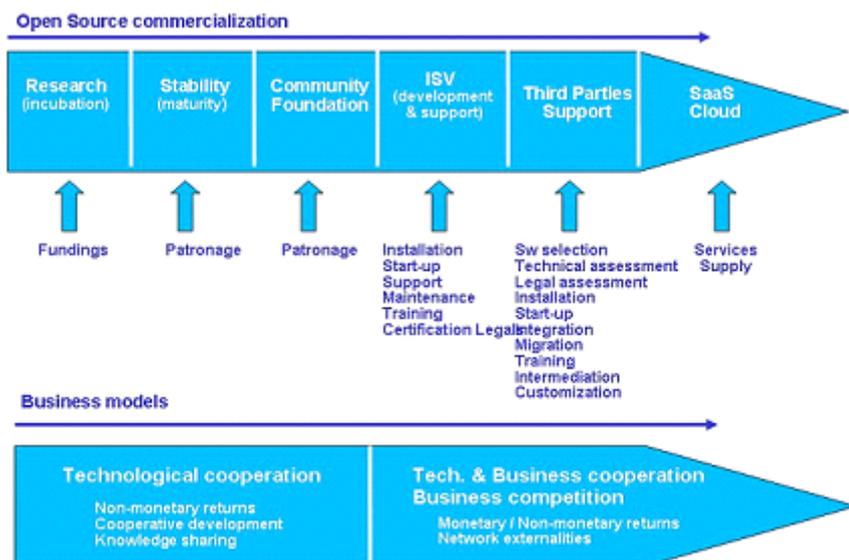
Engineering usa sus propias competencias en gestión de software libre, incluso usando soluciones software libre de terceros como componentes importantes de sus propios proyectos, o cuando necesita extenderlos, adaptarlos o resolver sus defectos de software. En este caso, Engineering proporciona las mejoras que realiza a la comunidad, obteniendo un doble beneficio:

- El soporte (de) y la participación activa en el ecosistema de software libre.
- La protección de la inversión del cliente para el que se ha realizado el proyecto. De hecho, si la contribución ofrecida es aceptada como parte de una nueva versión de la solución de software libre, hay claros beneficios en términos de mantenimiento y evolución futura.

Entre estas contribuciones, en el caso de desarrollos en Java, arquitecturas SOA y Business Intelligence, podemos nombrar los proyectos eXo Platform, Cimero, Jpivot, Harvard JHOVE, además de los mencionados previamente Eclipse STP y Apache ServiceMix.

La historia de las relaciones de Engineering con las comunidades muestra que:

- No sólo la *participación en la comunidad*, sino también la *"construcción de comunidad"* son cruciales, tanto en las comunidades ya existentes como en la creación de nuevas comunidades. Esto permite extender el "efecto red", facilitar la reputación global, intercambiar conocimiento y experiencias, y supone la aportación de contribuciones reales.
- La *gestión de la comunidad*, en la cual la empresa juega un papel líder en la gobernanza del proyecto, representa un elemento crítico y superable, especialmente cuando el proyecto presenta características especiales en términos de novedad y efectividad. Incluso en la experiencia de Engineering, es difícil construir una comunidad formada por actores que son activos en el soporte y la gestión del crecimiento del proyecto. El problema de las relaciones de confianza en la comunidad, entre la empresa y firmas de diversos tamaños, necesita todavía más investigación, puesto que es un



**Figura 1.** De la colaboración a la coopección: Situando el software libre a nivel de negocio.

aspecto fundamental en la construcción de una verdadera ecología de valor.

■ *Las relaciones con las comunidades* son procesos de creación de valor respaldados por redes complejas de relaciones sociales y de negocio, de valores compartidos, y de interdependencias entre los diferentes miembros. El valor obtenido es alto, aunque principalmente no cuantitativo (por ej., no monetario). Es una colaboración en una red industrial basada en el conocimiento, que comparte esfuerzos promocionales y un fuerte compromiso en las decisiones cruciales, en el fomento de la innovación, en la competición abierta y en la libertad para formar parte de las diferentes actividades.

**8. Sostenibilidad del modelo: una aproximación ecológica**

La iniciativa SpagoWorld proporciona un contexto donde su comunidad y los diferentes actores (empresas, vendedores, integradores, consultores en BI, instituciones públicas, clientes, académicos e individuos) cooperan para desarrollar una infraestructura de soluciones madura y fiable, y compiten para conseguir sus objetivos, creando un entorno activo y estimulante. De este modo, todos aquellos que adopten SpagoWorld pueden encontrar un entorno de referencia para su estrategia de adquisición de software libre, y una oportunidad para contribuir al crecimiento de una estrategia colectiva que pretende incrementar valor en un contexto ecológico. La cooperación (en asuntos no-monetarios) y competición (en el mismo mercado) simultáneas posibilitan relaciones complejas que fomentan el ecosistema (ver **figura 1**).

Desde este punto de vista, la iniciativa SpagoWorld es un ejemplo efectivo de pro-

moción de un ecosistema de negocios que actúa como el modelo de negocio específico de Engineering, y que se basa en el desarrollo y promoción de sus soluciones (vendiendo servicios de soporte y los proyectos de software correlacionados), y que al mismo tiempo apoya el sistema completo en un entorno de verdadera *coopección*. En el contexto del software libre, los retornos indirectos y no monetarios pueden ser de más valor que los monetarios porque suponen beneficios en un contexto sostenible a largo plazo.

**Referencias**

[1] O. Viele. Software Libero, opportunità o minaccia per i system integrator?, *Homepage Engineering Ingegneria Informatica Anno VII n.8*, Roma 2005 (en italiano)

[2] A. Ganzaroli, L. Pilotti. En R. Fiocca "Rileggere l'Impresa" (Capitolo 15 «Quali reti oltre il networking»), ETAS, 2007 (en italiano).

[3] M. Iansity, R. Levien. Keystones and Dominators: Framing Operating and Technology Strategy in a Business Ecosystem. *Harvard Business School, Working Paper #03-061*, 2004.

[4] G. Ruffatti. Ecosistemi di business in azione. Il caso Engineering. *Quaderni di management - n. 33, E.G.V. Edizioni, May-June 2008*, pp. 37-48 (en italiano). Disponible también en: <http://www.spagoworld.org/ecm/faces/public/guest/home/community/resources>.

[5] A. Bitterer. *Who's Who in Open-Source Business Intelligence*. Gartner Research, abril 2008.

[6] M. Pezzini, L.F. Kenney, J. Lennard. *Cool Vendors in Platform and Integration Middleware*. Gartner Research, marzo 2008.

**Notas**

<sup>1</sup> Engineering <http://www.eng.it/>.

<sup>2</sup> SpagoWorld Initiative <http://www.spagoworld.org>.

<sup>3</sup> OW2 Consortium <http://www.ow2.org/>.

<sup>4</sup> El término *coopección* significa la presencia contemporánea de relaciones de cooperación y competición.

<sup>5</sup> En la iniciativa SpagoWorld, que está basada en la estrategia comercial sobre software libre de Engineering, se requiere que el software se publique bajo la licencia GNU LGPL.

<sup>6</sup> El licenciamiento dual es un modelo híbrido, que incluye tanto una licencia libre (generalmente de la familia de licencias GNU) y su venta, o la venta de extensiones, bajo una licencia privativa siguiendo un esquema EULA (*End User License Agreement*). Algunos elementos de este modelo parecen más cercanos al modelo propietario que al del software libre. Por ejemplo, la presencia de inversores institucionales en las empresas que lo producen, el casi total control de la empresa que desarrolla la solución, y las características mismas de la solución, generalmente más típicas de un producto que de una plataforma.

<sup>7</sup> El número de descargas (esto es, el número de veces que un fichero es descargado por un usuario desde la web hasta su ordenador) se considera normalmente un número correlacionado con el éxito de un proyecto de software libre. Sin embargo, no tiene demasiado significado, porque existen muchos factores que pueden influir en este fenómeno, y tampoco dice mucho acerca del número de descargas que realmente se instalan y usan.

<sup>8</sup> Qualipso <http://www.qualipso.org/>.

<sup>9</sup> NESSI Open Source Working Group <http://www.nessi-europe.com/Nessi/WorkingGroups/AdoptionWorkingGroups/NESSIOpenSourceSoftwareWorkingGroup/tabid/269/Default.aspx>

Susana Muñoz Hernández,  
Jesús Martínez Mateo  
Facultad de Informática, Universidad  
Politécnica de Madrid

<{susana,jmartinez}@fi.upm.es>

# Una oportunidad para las empresas de software libre: mercado emergente en los países en vías de desarrollo

## 1. El software libre en los mercados occidentales

### ¿Por qué luchar contra el mercado?

Es bien conocido que el software libre y de código abierto<sup>1</sup> tiene que ganar una dura batalla contra el software propietario para encontrar un espacio en el mercado occidental. Algunos de estos problemas, quizá todos ellos, son específicos de este mercado donde el negocio del software tiene una historia consolidada. Por otra parte, existe un escenario donde puede ser evitada esta cultura tribal; en los países en vías de desarrollo muchos de estos problemas simplemente no existen. En general, el desarrollo y el despliegue del software en los países en vías de desarrollo tienen una corta tradición. Esto es una considerable ventaja para la expansión del software libre en aquellos mercados con una concepción más moderna en relación con el software [1].

### La bondad de los mercados

Existen varios casos en los que una nueva tecnología ha tenido menos problemas para establecerse en un mercado emergente que en los ya consolidados. Por ejemplo, a finales del siglo diecinueve era más sencillo abrir el mercado a las locomotoras eléctricas en aquellos lugares donde los trenes de vapor no eran aún habituales. Esto es similar, hoy día, donde muchos lugares aislados de países en vías de desarrollo han comenzado a utilizar directamente comunicaciones móviles sin haber tenido cualquier infraestructura previa de telefonía fija. En estos países se ha producido una brecha tecnológica en algunas generaciones en relación con la revolución industrial, el hardware, o las comunicaciones entre otras; entonces... ¿por qué no también con el software? Aunque quizá una pregunta más apropiada sería: ¿puede ser el software libre la tecnología del futuro en el desarrollo de software? Probablemente sí, en cuyo caso la historia podría volver a repetirse de una forma similar para ignorar el software propietario en beneficio del software libre en los países en vías de desarrollo. Bajo este supuesto es posible que los países en vías de desarrollo se conviertan en un serio candidato para adoptar una estrategia nacional a nivel de gobierno en relación con el software libre, sin pasar por la fase previa de la era del software privativo. Actualmente, siguiendo esta sospecha, algunas comunidades de software libre [2] se han dado cuenta de que existe un crecimiento y una oportunidad de negocio para ellos en estos países.

**Resumen:** durante los últimos años el negocio del software ha cambiado en muchos aspectos. Las compañías de software han evolucionado desde producir aplicaciones software y vender su código propietario, a proporcionar aplicaciones de código abierto y centrar su negocio en ofrecer servicios relativos a su adaptación, instalación, mantenimiento y formación de usuarios. Las compañías de software libre emplean un modelo de negocio que proporciona nuevas oportunidades en mercados donde el software privativo es inviable. En este artículo describimos una interesante oportunidad de negocio que está surgiendo para estas compañías: las necesidades de software de los países en vías de desarrollo. Estos países están creando un mercado emergente para el desarrollo de software especialmente interesante porque si sus necesidades y limitaciones especiales son tenidas en cuenta son un escenario perfecto para el enfoque del software libre. Finalmente, será también discutida la influencia de algunos catalizadores que están realmente actuando en este mercado objetivo.

**Palabras clave:** mercados emergentes, oportunidades de negocio, países en vías de desarrollo, software libre.

### Autores

**Susana Muñoz Hernández** es Doctor en Informática por la Universidad Politécnica de Madrid (UPM), tiene un máster en Gestión de las Tecnologías de la Información por la universidad Ramón Llull de Barcelona y un grado en Relaciones Internacionales por la Sociedad de Estudios Internacionales de Madrid. Ganó el primer premio en el concurso nacional para jóvenes talentos de la Universidad de La Salle de Madrid en 2003. Tras varios años de experiencia profesional trabajando en empresas privadas, comenzó a trabajar como profesora asociada en la Escuela de Informática de la Universidad Politécnica de Madrid en 1998 donde lleva a cabo su actividad investigadora en el grupo BABEL. Es responsable de difusión y comunicación de la plataforma nacional española de software y servicios, INES. Ha establecido el grupo de cooperación TEDECO (*Technology for Development and Cooperation*) en 2006 y ejerce como directora desde entonces. Desde TEDECO, lidera la iniciativa Morfeo-CODE para la aplicación de software libre a la COoperación para el DESarrollo en la comunidad Morfeo.

**Jesús Martínez Mateo** es estudiante de doctorado en Informática en la Universidad Politécnica de Madrid (UPM). Tiene un máster en Matemáticas de la Computación por la UPM. Desarrolla su actividad investigadora en el laboratorio de informática en la sección departamental de Análisis Numérico como integrante del grupo de investigación en Información Cuántica y Computación. Contribuye a varios proyectos, entre otros el proyecto Cenit Segur@ financiado por el Ministerio de Comercio e Industria de España para el estudio y diseño de una Red de Distribución Clave Cuántica Metropolitana. Compagina su trabajo investigador con el desarrollo de software libre (ej. Lan Core) y la participación en el grupo TEDECO (*Technology for Development and Cooperation*), siendo un miembro activo que ha participado en dos misiones de campo en Burundi.

### ¿Tradicición o PCI (Progreso, Cambio e Innovación)?

El problema más importante del software libre en las grandes compañías es que éstas ya tienen la mayoría de sus aplicaciones desarrolladas utilizando software propietario. La tradición es una poderosa desventaja puesto que las migraciones de software pueden implicar inversiones a corto plazo y las compañías raramente confían en las ventajas que a medio y largo plazo les puede suponer una migración a software libre, u otros productos. Por el contrario, encontramos algunas instituciones, organizaciones y compañías en

los países en vías de desarrollo que están comenzando a generalizar el uso de aplicaciones software [3]. En estos casos, y por razones evidentes al tratarse de países en vías de desarrollo, el presupuesto para la compra de software es considerablemente reducido, de tal forma que la adquisición de software libre no es una opción sino una necesidad. En cualquier caso, el software libre es mucho más económico que la compra de una aplicación propietaria convencional, por lo que las desventajas tradicionales son oportunidades en los mercados de software de los países en vías de desarrollo.

**Comercio justo**

El comercio justo es un movimiento social organizado y un enfoque del mercado que tiene como objetivo ayudar a los productores en los países en vías de desarrollo al mismo tiempo que promueven la sostenibilidad. El comercio justo no es tan sólo prestigioso desde el punto de vista moral, sino también por su rentabilidad. En 2007, las mercancías etiquetadas como comercio justo tuvieron un nivel de ventas por valor de aproximadamente 2,3 billones de euros en todo el mundo (3,62 billones de dólares americanos), con un incremento del 47% de un año para otro.

Ha existido un debate en un foro especializado acerca de las similitudes entre el software libre y el comercio justo [4]. El origen de esta discusión fue una presentación de Matthew Edmonson del proyecto gubernamental *Open IT Up* que dio en el Reino Unido unas conferencias para proveedores de asistencia tecnológica<sup>2</sup>. En realidad, existen bastantes semejanzas teniendo en cuenta que ambos implican una elección y una preocupación por las personas. James Davis dice que como usuarios de software libre esperamos que se nos conceda la libertad de hacer lo que nosotros queramos con nuestro software, siempre y cuando no se restrinja lo que los demás pueden hacer con nuestro software modificado.

A pesar de que no estemos de acuerdo con esta declaración (puesto que podemos encontrar algunas diferencias sustanciales entre el software libre y el comercio justo [5]) este es un punto muy positivo para el software libre, el ser comparado en un debate público con tan prestigiosa empresa como es el comercio justo.

**2. Catálisis del software libre en los países en vías de desarrollo**

**Las TIC en los países en vías de desarrollo**

Lejos del ancestral concepto de "cooperación", mucho más cercano a la caridad, en los últimos cinco años los proyectos de cooperación están orientados al desarrollo de los países receptores. Aparte de los campos tradicionales de cooperación (salud pública, educación, o el suministro de alimentos entre otros) existen nuevas áreas donde la cooperación está jugando un rol decisivo, y las relacionadas con las Tecnologías de la Información y la Comunicación (TIC) son probablemente las más importantes. De hecho, la cooperación en TIC está directamente aplicada al resto de áreas de cooperación tradicionales.

**Software libre para el fortalecimiento institucional**

La mayoría de los proyectos de cooperación que incluyen desarrollo de software han optado por el uso de software libre. Esto es muy interesante si tenemos en cuenta que este software va a ser parte de muchas instituciones clave (Administración o Educación entre otras) de los países en vías de desarrollo [6]. El e-Gobierno está siendo especialmente promovido y muchos gobiernos de países en vías de desarrollo están apostando por el software libre siguiendo el consejo de consultores externos independientes y Organizaciones No Gubernamentales (ONG) [7][8]. Es evidente la influencia que este hecho tendrá en el desarrollo de software en el resto de los sectores de estos países.

**Siguiente generación de programadores de primera categoría**

En la actualidad, sin otra opción, las tendencias actuales en software libre y propietario en los países en vías de desarrollo vienen de países extranjeros. Sin embargo, se espera que en un futuro cercano sus primeras generacio-

nes de ingenieros informáticos sean los responsables del desarrollo, pero sobre todo del mantenimiento y uso del software. Este es otro punto a favor del software libre. Los esfuerzos en educación están concentrados en el campo del *e-learning* y el desarrollo de la educación superior. La influencia del software libre en la educación de las nuevas generaciones de programadores y profesionales de los países en vías de desarrollo está siendo reforzada por el trabajo de las ONGs y la cooperación de las universidades occidentales [9].

**3. Países en vías de desarrollo: mercados de software libre emergentes**

El software libre es una alternativa consolidada para el desarrollo de software profesional que es altamente recomendada para los países en vías de desarrollo. Durante los últimos años existen estudios y esfuerzos desde estos países que apuestan por el software abierto y los modelos de software libre [10]. Unos quince países africanos ya han publicado recomendaciones de software libre frente a soluciones de software propietario (por ej, Angola, Benin, Kenia, Senegal, Sudáfrica, Tanzania, Uganda y Zambia entre otros). Esto puede ser aplicado a muchas áreas pero es especialmente importante para tres: Educación, enfocado principalmente en las bibliotecas digitales [11], desarrolladores de software, o soporte TIC para la educación general; Administraciones Públicas, con especial interés en el despliegue del e-gobierno; y Salud, no sólo proveyendo software para instituciones de salud sino también para desarrollar el crecimiento de la e-Salud y su utilización en áreas aisladas [12].

Aparte de estas áreas en las que la introducción del software libre ya ha comenzado, el sector privado está dispuesto a desarrollar un

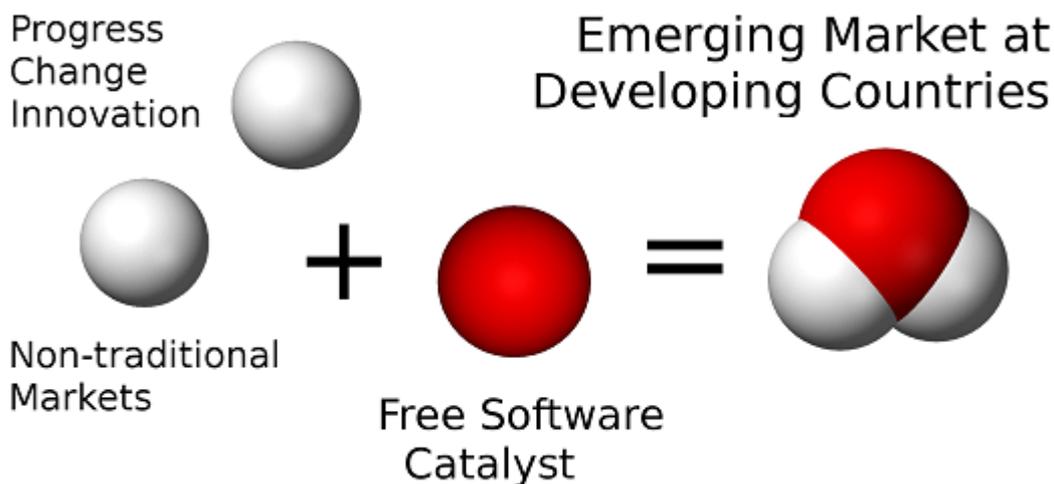


Figura 1. Proceso de catálisis del software libre en los países en vías de desarrollo.

software de apoyo confiable y seguro, y sus decisiones al elegir el tipo de software que van a utilizar está muy condicionada por el criterio de la Administración Pública y los sectores públicos en general.

Teniendo en cuenta las características especiales de los países en vías de desarrollo, su falta de reticencias a la hora de usar el software libre, el efecto catalizador de la cooperación y el previsible crecimiento de la industria de software en estos países (ver **figura 1**), podemos concluir diciendo que la aparición del mercado de software en los países en vías de desarrollo representa una oportunidad de negocio para las empresas de software libre.

## Referencias

- [1] **J. Lerner, J. Tirole.** "The Simple Economics of Open Source". *National Bureau of Economic Research, Inc., NBER Working Papers, No. 7600, 2000.* <<http://ideas.repec.org/s/nbr/nberwo.html>>.
- [2] **Morfeo-code.** Iniciativa para el uso de software libre en la cooperación para el desarrollo, <<http://tedeco.morfeo-project.org/>>.
- [3] **G. Camara, F. Fonseca.** "Information Policies

and Open Source Software in Developing Countries". *Journal of the American Society for Information Science and Technology, 2007.*

[4] **D. Wilcox.** *Is Open Source Software Fair Trade for Nonprofits?* Designing for Civil Society. <[http://partnerships.typepad.com/civic/2007/02/is\\_open\\_source\\_.html](http://partnerships.typepad.com/civic/2007/02/is_open_source_.html)>.

[5] **J. Davis.** *How Free Software is not Fair Trade for non-profits.* Free web hosting for UK charities and non-profit organizations. <<http://www.freecharity.org.uk/2007/02/08/how-free-software-is-not-fair-trade-for-non-profits/>>.

[6] **S. Weber.** "Open Source Software in Developing Economies". Social Science Research Council, USA. <[http://www.ssrc.org/programs/itic/publications/ITST\\_materials/webernote2.pdf](http://www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf)>.

[7] **United Nations Economic y Social Commission for Asia and the Pacific (ESCAP).** "ICT Policy database," <<http://www.unescap.org/ictstd/policy/db/>>.

[8] Programa para el Desarrollo de las Naciones Unidas (United Nations Development Programme, UNDP) y el Centro Internacional de Investigación para el Desarrollo (International Development Research Centre, IDRC) de Canadá. *International Open Source Network (IOSN) – Software Freedom For All,* <<http://www.iosn.net/>>.

[9] **TEDECO: Tecnología para el Desarrollo y la Cooperación.** <<http://tedeco.fi.upm.es/>>.

[10] **Bridges.org, CIPESA.** *Free/Open Source Software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners,* <[http://](http://www.cipesa.org/files/FOSSPolicyToolkit.pdf)

[www.cipesa.org/files/FOSSPolicyToolkit.pdf](http://www.cipesa.org/files/FOSSPolicyToolkit.pdf)>.

[11] **I. Witten, M. Loots, M. Trujillo, D. Bainbridge.** "The Promise of Digital Libraries in Developing Countries" *Communications of the ACM, 44 (5), pp. 82-85, 2001.*

[12] **Fundación EHAS: Enlace Hispano Americano de Salud.** <<http://www.ehas.org/>>.

Fecha de revisión de los enlaces web: mayo de 2009.

## Notas

<sup>1</sup> En este artículo, hemos decidido utilizar el término *software libre* siguiendo la definición de la Free Software Foundation (FSF) en lugar de otros términos comunes, y seguramente más populares, como FOSS, del inglés *Free and Open Source Software*, o OSS, del inglés *Open Source Software*.

<sup>2</sup> **Nota del editor:** En la referencia aportada por los autores se comenta que Matthew Edmonson dio una conferencia a "circuit riders" que dan soporte tecnológico a empresas sin ánimo de lucro. Este término (también se puede usar "eRiders" en su lugar), tiene su origen en el movimiento metodista y se refiere a proveedores de asistencia tecnológica que atienden a pequeñas organizaciones sin ánimo de lucro de sectores económicos concretos para la resolución de problemas o el soporte a necesidades tecnológicas particulares de dichas organizaciones <[http://en.wikipedia.org/wiki/Circuit\\_rider\\_\(technology\)](http://en.wikipedia.org/wiki/Circuit_rider_(technology))>.

September 23-25, 2009

DISC2009

23rd INTERNATIONAL SYMPOSIUM ON  
DISTRIBUTED COMPUTING  
ELCHE/ELX ■ SPAIN

<http://disc2009.gsync.es>

The program this year will feature:

- \* keynote lectures by Lorenzo Alvisi, Nir Shavit, and Willy Zwaenepoel
- \* five exciting associated workshops (on September 22nd and 26th, 2009)
- \* a tutorial on cloud computing
- \* 60th birthday celebration of Michel Raynal and Shmuel Zaks
- \* the 2009 Edsger W. Dijkstra Prize in Distributed Computing ceremony

## Program Committee

- \* Ittai Abraham, Microsoft Research SCV
- \* Yehuda Afek, Tel-Aviv University
- \* Marcos K. Aguilera, Microsoft Research SCV
- \* James Aspnes, Yale
- \* Christian Cachin, IBM Zurich Research Laboratory
- \* Gregory V. Chockler, IBM Haifa Research Laboratory
- \* Carole Delporte-Gallet, University of Paris Diderot
- \* Pascal Felber, University of Neuchatel
- \* Seth Gilbert, École Polytechnique Fédérale de Lausanne
- \* Danny Hendler, Ben-Gurion University
- \* Ricardo Jiménez-Peris, Universidad Politécnica de Madrid
- \* Idit Keidar, Technion (Chair)
- \* Zvi Lotker, Ben-Gurion University
- \* Thomas Moscibroda, Microsoft Research
- \* David Peleg, Weizmann Institute
- \* Eric Ruppert, York University
- \* Elad M. Schiller, Chalmers U. of Technology and Gothemburg U.
- \* Mark R. Tuttle, Intel
- \* Robbert van Renesse, Cornell University
- \* Jay J. Wylie, HP Labs
- \* Lidong Zhou, Microsoft Research Asia

## Organizing Committee Chairs

- \* Vicent Cholvi Juan, Universitat Jaume I
- \* Antonio Fernández Anta, Universidad Rey Juan Carlos



GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA E INNOVACIÓN

Microsoft

Research



Universidad del País Vasco Euskal Herriko Unibertsitatea

Ajuntament d'ELX

turisme d'ELX



Universidad Rey Juan Carlos

Agustín Cernuda del Río<sup>1</sup>,  
Manuel Quintela Pumares<sup>2</sup>,  
Miguel Riesco Albizu<sup>1</sup>

<sup>1</sup>Dpto. de Informática, Universidad de Oviedo; <sup>2</sup>alumno de la Escuela de Ingeniería Informática de Oviedo

<guti.albizu@uniovi.es>,  
<manuel.quintela@gmail.com>

## 1. Introducción

El paso a un enfoque pedagógico basado en competencias profesionales exige una reflexión sobre métodos y herramientas docentes. Existen diversas definiciones de lo que es una competencia en este ámbito, no todas coincidentes. Cabe aceptar que las competencias tienen un componente de interiorización y preparación personal para reaccionar ante circunstancias concretas, no tan explícito cuando se habla de conocimientos.

Según de Miguel [10], la competencia es *una potencial de conductas adaptadas a una situación*. Se dice que "si la confrontación con el ejercicio real no se produce, la competencia no es perceptible o no se pone a prueba", que "sólo existe la competencia si se vincula a un objeto o una situación", y se justifica que "debe colocarse al estudiante ante *diversas* situaciones de estudio y trabajo similares a las que puede encontrar en la práctica de su profesión" (la cursiva aparece en el original).

En este artículo se reflexiona sobre la utilización de un juego de rol como herramienta docente sobre el ejercicio de la profesión informática. Se describe una experiencia de construcción de un juego desarrollado con este fin, sus características, las experiencias realizadas y las conclusiones extraídas hasta el momento.

Aquí se utilizará el término *role-play* (por estar extendido su uso en castellano en el contexto educativo) para aludir a ejercicios prácticos en los que el alumno desempeña un papel. El término *juego de rol* no se utilizará como traducción del anterior, sino para aludir a juegos ofrecidos por la industria del entretenimiento (en inglés, *role-playing games* o RPG).

## 2. Simulación y role-play como método didáctico

La idea de colocar al alumno en una situación que imite a la real, con fines formativos, no es nueva. Se ha hecho por medio de juegos, simulaciones y ejercicios de *role-play* de maneras muy diversas.

En ámbitos universitarios es tradicional el uso del *role-play* en aspectos sociales (como la negociación [1]); en la informática también se ha utilizado para tareas exploratorias, ejerciendo el alumno como elemento de un sistema para llegar a una mejor comprensión

# Un juego de rol para la enseñanza de la profesión informática

Este artículo fue seleccionado para su publicación en **Novática** entre las ponencias presentadas a las XIV Jornadas de Enseñanza Universitaria de la Informática (JENUI 2008) celebradas en Granada de las que ATI fue entidad colaboradora.

**Resumen:** *la transformación de las enseñanzas universitarias para hacer hincapié en las competencias profesionales (no sólo en los conocimientos) lleva a plantearse qué métodos de enseñanza pueden facilitar que el alumno adquiera tales competencias "transversales". Muchas están relacionadas con el comportamiento en situaciones concretas, ya sea como efecto (reacciones) o causa (toma de decisiones). Suelen exigir el manejo de patrones de interacción personal que tienen consecuencias futuras. Reflexionando sobre la profesión, cabe recordar también que el público en general tiene una imagen difusa de la ingeniería en informática y no siempre le asocia las competencias mencionadas. Los juegos de rol pueden ser un elemento útil, aún poco explotado, en el entrenamiento de los alumnos para el ejercicio de esas competencias profesionales y en la difusión de la profesión informática entre el público en general.*

**Palabras clave:** *competencias profesionales, ingeniería en informática, juego de rol, simulación*

del mismo, como en programas concurrentes, sistemas dinámicos u orientación a objetos. Fuera de la enseñanza, esto es habitual también en métodos de análisis como las fichas CRC. No obstante, no parece haberse explotado en la misma medida el uso de juegos de rol (en sentido estricto) para estos fines (hay alguna experiencia con juegos de cartas [2], producto relativamente afín a los RPG).

Por supuesto, cierta imagen negativa y sensacionalista que se ha venido asociando a los juegos de rol no tiene base científica alguna. Diversos estudios [12] afirman que jugar a rol no está relacionado con rasgos particulares de personalidad, o que incluso puede haber correlación con rasgos positivos como una mayor inclinación a la experimentación [5]. Parece probado que los juegos de rol [7] promueven el ejercicio de la empatía, socialización, tolerancia (a lo extraño, indefinido o impreciso), aceptación de la diversidad, etc. y existe literatura sobre su uso como método terapéutico para tratar dificultades de socialización o problemas mentales [4].

## 3. Los juegos de rol

### 3.1. Generalidades

Un juego de rol (*role-playing game* o RPG) es básicamente un juego de tablero, en el que varias personas se reúnen en torno a una mesa para representar una historia. La historia está abierta y los participantes elaboran su papel improvisando y decidiendo el curso de la misma. El objetivo no suele ser ganar, sino simplemente desenvolverse en un marco más o menos imaginario. Esto implica un comportamiento flexible, dinámico y (muy frecuentemente) cooperativo.

Uno de los jugadores tiene una misión espe-

cial; es el *director*. Describirá la situación inicial (y cualquier otra que se vaya produciendo) y hará de árbitro. Los demás jugadores, en cada nueva situación, pretenderán realizar acciones, que pueden tener o no éxito; para ayudar a decidirlo, el juego de rol tendrá una serie de reglas. A partir de ciertos valores o habilidades de cada personaje (reflejados en su *ficha de personaje*), y con un ingrediente de probabilidad (mediante dados), la aplicación de las reglas permite dilucidar qué ocurre ante una propuesta de un jugador.

El director de juego es quien aplica las reglas y las adapta o reinventa para resolver las (frecuentes) situaciones imprevistas.

### 3.2. Elementos

El manual de un juego de rol suele incluir:

- Un *desarrollo descriptivo* de todo un mundo en el que tiene lugar la partida: lugares, fechas, sucesos, razas, tribus, útiles, personalidades, costumbres... Es toda una obra literaria de ficción en la que participan artistas (guionistas, dibujantes...) y los buenos aficionados llegan a mostrar un conocimiento sorprendente de algo tan vasto.
- Una plantilla llamada *ficha de personaje*, con ciertos *atributos* (rasgos de personalidad, habilidades, niveles de salud o fatiga, etc.). Para crear cada personaje se da valor numérico a los atributos.
- Unas reglas del juego en las que se describe lo que un personaje puede hacer en función de esos atributos, o cómo evolucionan dichos atributos como consecuencia de sus actos. Normalmente se utilizan ponderaciones de atributos combinadas con las tiradas de dados.

Además de esto, es el director quien prepara una partida, describiendo una situación concreta en la que los personajes desarrollan la

acción. Una partida puede durar algunas horas, o extenderse a lo largo de varias sesiones.

Aunque en ámbitos periodísticos se habla de intervenciones de los jugadores en el mundo real, eso es excepcional. Algunos jugadores se esmeran con ambientación o disfraces, pero es raro; y más aún jugar fuera de la sala o interactuar con terceros. Cualquier comportamiento patológico es tan incomprensible para los aficionados como para el público.

Así, un juego de rol en su versión de mesa es prácticamente un libro (el manual) y no tiene por intervenir ningún dispositivo informático, aunque también existe esa opción.

### 3.3. Los juegos de rol comerciales

La industria de los juegos de rol tiene ya tres décadas y cuenta con muchos seguidores. Un juego de rol comercial puede requerir años de desarrollo. Si tiene éxito se suelen publicar sucesivos *módulos* adicionales de juego, que amplían las descripciones iniciales, plantean situaciones o partidas prefabricadas, etc.

### 4. Objetivos

En este proyecto se pretende aprovechar el planteamiento de los juegos de rol comercia-

les en conexión con el aprendizaje de competencias, implicando al alumno de dos maneras: como jugador y como desarrollador del juego.

Un segmento significativo de los consumidores de RPG encaja en el rango de edad de los estudiantes de ingeniería y, además, los jugadores habituales de rol están acostumbrados a participar en estas simulaciones de forma natural. Por otra parte, la creación o perfeccionamiento de un RPG sobre la profesión mueve al alumno a investigar y reflexionar sobre diversas facetas de la misma.

Respecto a la divulgación, un RPG puede ser un medio para llegar a personas ajenas a los estudios de ingeniería en informática.

Se planteó como un primer objetivo el desarrollo de un RPG como sigue:

- Debe permitir practicar diversas habilidades y competencias si es usado con tal fin.
- Las reglas del sistema de juego deben ir respaldadas por fuentes académicas o criterios didácticos.
- Debe ser plenamente funcional (comparable en su uso a los juegos de rol comerciales).
- Las reglas (reflejando siempre la realidad) deben adaptarse para que el juego sea divertido y ágil, y no difícil de jugar.

De conseguirse estos objetivos, en futuras fases del proyecto cabría estudiar empíricamente las diversas formas de aplicación que la literatura describe para estas técnicas (en clase, de forma voluntaria, en aprendizaje por proyectos, etc.); cosa que dista de ser trivial [11].

### 5. El juego

A continuación se ofrece una descripción del juego, necesariamente breve (al fin y al cabo, el propio manual no es más que una descripción detallada).

#### 5.1. Cómo se construyó

Para asegurar una buena conexión con el perfil de los usuarios de RPG comerciales (y aprovechar la vertiente formativa de construir el juego), se ha intentado implicar a alumnos jugadores de rol. La primera versión se desarrolló como un Proyecto Fin de Carrera de Ingeniería Técnica en Informática. Se dio al juego el nombre de *¡Hola, mundo!* [8].

En fases iniciales, se contactó con un autor de juegos profesional, lo que permitió confirmar que no existe una práctica común para el desarrollo de estos productos; el proceso suele ser artesanal, creativo, y propio de cada empresa. En nuestro caso se adoptó un ciclo de vida en espiral; se desarrollaría un prototipo muy sencillo, se jugaría una partida de prueba para depurar las reglas y se iteraría hasta llegar a un punto satisfactorio.

El juego tendría una estructura típica: una descripción general del universo, pautas para crear personajes y reglas de desarrollo del juego. Tendrían gran importancia las historias o partidas predefinidas para que las utilizase el director:

- En la descripción del universo reside parte del valor divulgativo acerca de la profesión para el público en general.
- En las historias concretas reside el grueso de la estrategia docente del juego.

Se ha acudido a un marco de referencia de los existentes para la creación de juegos de rol, concretamente Fudge [6]. Se trata de una estructura básica sobre la que los desarrolladores de RPG pueden montar sus juegos, definiendo los aspectos concretos.

Respecto a las historias predefinidas, que el docente puede crear para ejercitar competencias concretas, la secuencia de creación es:

- Elegir y mencionar las competencias que se desea ejercitar, citando explícitamente la fuente: *computing curricula*, mapas de competencias de la titulación, textos sobre ingeniería del software, etc.
- Elegidas las competencias, describir el objetivo didáctico de la historia, los elementos a incorporar y las pautas a seguir.
- Finalmente, redactar la historia propiamente dicha, basándose en el análisis precedente. Esta historia es lo que realmente utilizan director y jugadores.

#### 5.2. Personajes

Un personaje se define mediante una *ficha de personaje*. La ficha contiene *atributos, habilidades, conocimientos técnicos* y variables de estado. Cada uno de ellos tiene asignado un Valor Fudge (VF), que es un valor entre -3 y +3; 0 representa el grado medio, -3 sería un grado "terriblemente malo" y +3 "espectacular". En la **figura 1** puede verse una lista de todos los elementos mencionados; un personaje concreto tendría un VF para cada parámetro ahí mencionado.

- Los *atributos* son características genéricas intrínsecas del personaje, que "no se estudian". Un atributo se relaciona con habilidades (reflejadas con siglas).
- Las *habilidades* son características que un personaje ha aprendido.
- Los *conocimientos técnicos* son conceptos teóricos y/o prácticos específicos.
- El *estado del personaje* engloba características que fluctúan en el tiempo, dependiendo de lo que le va ocurriendo al personaje, y que evolucionan más rápidamente que las demás características.

Las reglas de creación producen personajes no muy experimentados, que no llevan mucho tiempo en el mercado laboral<sup>1</sup> (pero pueden modificarse si se desea). Bajo esta premisa, para crear un personaje se asigna a cada atributo un VF entre -3 y +2, con la condición

<u>Atributos</u>	<u>Habilidades</u>
Análisis y Comprensión (AC)	Persuasión (RI)
Síntesis y Aplicación (SA)	Engañar (RI)
Evaluación (E)	Expresión (RI)
Relaciones interpersonales (RI)	Etiqueta (RI)
Aplomo (A)	Reunir información (AC)
Salud (S)	Cultura general (SA)
<u>Conocimientos técnicos</u>	Análisis (AC) (SA)
Algorítmica	Diseño (AC) (SA)
Administración de sistemas	Construcción (AC) (SA)
Programación	Implantación (AC) (SA)
Hardware dispositivos	Prueba (E)
Comunicac. persona-máquina	Gestión de proyectos (AC)
Sistemas información	Gestión de la calidad (E)
Bases de datos	Estimación (E)
Sistemas inteligentes	<u>Estado del personaje</u>
Redes y comunicaciones	Nivel de salud
Desarrollo web	Nivel de moral

Figura 1. Elementos de la ficha de personaje.



Figura 2. Imagen de la tercera partida de prueba.

de que la suma de todos los VF de los atributos debe ser 0. Es decir, un personaje no puede sobresalir en todas las facetas (de ahí la suma 0) y además ningún personaje parte con un nivel de excelencia (+3) en ninguna de ellas (un nivel así siempre ha de conseguirse con la experiencia, durante la partida).

Los criterios para dar VF a las habilidades son los mismos. Para los conocimientos la suma inicial debe ser -4 en lugar de 0 (los conocimientos pueden adquirirse con más rapidez durante el juego).

El nivel de salud inicial será el mismo que el del atributo *salud* y el nivel del moral el mismo que el del atributo *aplomo* (partiendo con un valor mínimo de 0 en todo caso).

Durante la partida, el director otorga puntos de experiencia, que pueden canjearse (según ciertas tablas) para mejorar en atributos, habilidades o conocimientos.

### 5.3. Reglas

Durante el juego, los personajes intentarán realizar acciones. Su éxito depende de la capacidad del personaje, de las circunstancias y del azar. Es un juego de probabilidades.

En una tirada Fudge (TF) intervienen 4 dados Fudge (DF). Un DF da un valor entre -1 y +1, así que cada tirada está entre -4 y +4. Además de los dados, interviene una característica del personaje (dependiendo de qué pretenda hacer) y diversos modificadores, dependiendo de las circunstancias, que se explican en el manual. Una acción tendrá asociado un grado de dificultad, que será un VF. La diferencia entre ese VF y la TF dará el grado de éxito de la acción; cualquier resultado negativo representa el fracaso.

En algunas tiradas sólo intervienen las características de un personaje (por ejemplo, encontrar un error en un programa), pero en otras se oponen las tiradas de dos personajes (por ejemplo, cuando un analista intenta convencer a un cliente entran en juego la capacidad de persuasión del analista y la dureza negociadora del cliente).

En el manual del juego se describen con detalle muchas otras pautas específicas para resol-

ver situaciones, hacer evolucionar los personajes, etc.

### 5.4. Entorno del juego

Una parte importante del manual de un RPG es la descripción del mundo en el que se desarrolla. Se describen (en algunos casos de forma casi literaria) entornos, profesiones, personajes, actitudes, personalidades, eventos históricos y todo aquello que complete la ficción. En la versión actual se describe el desarrollo de proyectos de informática y las características de las empresas de informática.

Se introducen diversas definiciones (proceso, tarea, producto...), se describen las actividades ligadas al desarrollo de un proyecto y se ofrecen elementos suficientes para que un director de juego (quizás ajeno a la informática) pueda manejar partidas en estos escenarios. Por supuesto, el ámbito del juego no tiene por qué limitarse al desarrollo de software (véase el apartado "Campos de aplicación"), pero se utilizó como un primer caso de uso.

Para desarrollar las historias, se ofrecen también tablas (que podemos considerar reglas adicionales) basadas en la literatura sobre ingeniería del software que ofrecen parámetros numéricos, tales como recursos necesarios para el desarrollo según la complejidad de un proyecto.

### 5.5. Pruebas con usuarios

Para refinar las reglas, se organizaron tres partidas reales. En dos se trabajó con jugadores de rol habituales no informáticos y en otra con estudiantes de informática, tanto jugadores de rol como debutantes. Tras cada partida se depuraba el juego.

La primera partida reunió al director de juego (el alumno coautor del mismo) con 4 jugadores (con hasta 10 años de experiencia con RPG), todos ajenos a la informática, para comprobar la validez divulgativa del juego.

No se dio a los jugadores información específica sobre la profesión o lo que *debe hacer* un ingeniero en informática; sólo se les explicó el juego. Antes y después de la partida se les pasó un cuestionario sobre las tareas que realiza

un ingeniero en informática; se comprobó que con una sola partida había cambiado su forma de ver la profesión. Por ejemplo, antes se consideraba tareas relevantes de un ingeniero en informática "programar, revisar los ordenadores e instalar programas, mantener los equipos informáticos", etc. Después, las respuestas eran del tipo "Entrevistarse con los clientes e identificar lo que necesitan, organizar el tiempo para cumplir las fechas de entrega, controlar cómo evoluciona el proyecto para evitar tener que 'volver atrás'", etc. Con esta prueba se verificó que el juego era utilizable (incluso por no informáticos) y que cumplía los objetivos divulgativos.

En una segunda partida se reunieron dos estudiantes de I. T. en Informática (con poca o ninguna experiencia como jugadores de rol) con el director de juego. Aunque el desarrollo fue distinto (por ejemplo, siendo más conscientes de los problemas de la estimación negociaron características y coste del producto de forma mucho más ventajosa) las conclusiones son análogas y básicamente positivas.

Hubo una tercera partida de prueba con el primer grupo, que se grabó en vídeo para su posterior análisis (ver figura 2).

## 6. Aplicación pedagógica

### 6.1. Aspectos de aprendizaje

Según Giménez [7], un beneficio de los RPG aplicados a la educación es convertir en significativo un aprendizaje que de otro modo no lo sería. Por otra parte, la comunicación y expresión oral son elementos constantes. Además, al desarrollar estrategias y sopesar los efectos de las decisiones, los jugadores frecuentemente toman notas, organizan información, realizan mapas mentales y discuten esquemas de actuación con compañeros de juego. Los RPG promueven la lectura; los jugadores se aprenden prácticamente de memoria enormes manuales, y frecuentemente redactan otros que son notables ejercicios literarios y creativos.

El efecto en actitudes y competencias sociales es quizá el más estudiado [4][5][12]. "Meterse en los zapatos de otro" desarrolla la empatía; se han utilizado frecuentemente los RPG como herramienta de socialización. En una partida surgen diversas formas de colaboración y trabajo en equipo, ya sea por las situaciones representadas o porque constantemente se necesita consenso para el propio desarrollo del juego. Los enfrentamientos están claramente despersonalizados, ya que el jugador siempre es consciente de que el otro interpreta un papel.

### 6.2. Campos de aplicación

En un primer contacto con los RPG resulta chocante su carácter extremadamente abierto

e indefinido. Se trata de juegos con un puñado de normas, que el director puede modificar o aplicar a su manera y en los que *no se sabe qué va a pasar* realmente, ni en qué consiste ganar la partida.

Esta ambigüedad es una ventaja. Nada impide preparar partidas con arreglo a muy diversos enfoques:

- Desarrollar historias "de acción" basadas en el propio argumento, centrándose en decisiones, interacción humana, negociación, flujo de información, cooperación, competición...
- Crear historias altamente especializadas, en las que el jugador tenga realmente que resolver problemas técnicos de gran dificultad y aplicar sus conocimientos. En lugar de tirar dados para dilucidar si una acción tiene éxito o no (que puede ser imprescindible para simular una lucha entre vampiros) el jugador puede conseguir o no normalizar determinado esquema relacional, o elegir determinado algoritmo cuya eficiencia decida el curso de la historia en el futuro.
- Otra área de gran interés puede ser la deontología profesional [9]. Colocar al jugador en dilemas de tipo deontológico suele resultar poco académico, e incluso secundario, en un proyecto de ingeniería del software donde el alumno está aprendiendo a documentar o gestionar. Pero las posibilidades son infinitas en el marco de una partida de rol.
- Otro grado de libertad es la temática de la historia; más allá del desarrollo de software, se puede reflexionar sobre administración de sistemas, seguridad, *hacking*, auditoría informática, peritajes y actuaciones judiciales, responsabilidad civil, mantenimiento, documentación, informática forense, formación, protección de datos de carácter personal, firma electrónica...

### 7. Conclusiones

Aunque el *role-playing* es una técnica pedagógica conocida, no parece existir tradición de aplicar los llamados juegos de rol a la enseñanza de la profesión informática. Sin embargo, se ha podido comprobar que es perfectamente posible.

Además, se ha podido demostrar que el juego de rol construido es un producto utilizable en la práctica, y que ha resultado satisfactorio para jugadores habituales (exigentes, por tanto) de rol.

Respecto a los aspectos educativos, se ha proporcionado un enfoque metódico que permite ampliarlo de manera controlada y documentada para cubrir nuevos objetivos didácticos y facilitar a terceros el análisis, mejora y adaptación de las historias desarrolladas.

Además de utilizarlo para que los jugadores reflexionen sobre los temas elegidos, la faceta de desarrollador del juego también resulta

altamente instructiva si la realizan alumnos.

Respecto a su finalidad divulgativa, se ha demostrado también que pueden utilizarlo personas ajenas a la informática que, de manera natural, ven mejorar de manera notable su conocimiento sobre la profesión.

En un futuro próximo se pretende *crear comunidad* en torno al juego, perfeccionar su núcleo y sobre todo crear muchas más historias predefinidas que permitan explorar los campos de aplicación mencionados.

### Agradecimientos

Los autores desean hacer constar su agradecimiento a los voluntarios que participaron en las partidas de prueba.

### Referencias

- [1] Patrick D'Aquino, Olivier Barreteau, et al. The role playing games in an ABM participatory modeling process: outcomes from five different experiments carried out in the last five years. En «*Integrated assessment and decision support*». iEMSs. 1st biennial meeting of the International Environmental Modelling and Software Society», Lugano, junio 2002.
- [2] Alex Baker, Emily Oh Navarro, Emily, André van der Hoek. An Experimental Card Game for Teaching Software Engineering. *Proceedings of the 16th Conference on Software Engineering Education and Training. IEEE Computer Society*, 2003.
- [3] Olivier Barreteau. The joint use of role-playing games and models regarding negotiation processes: characterization of associations. *Journal of Artificial Societies and Social Simulation vol. 6, no. 2*, 2003.
- [4] Wayne D. Blackmon. Dungeons and Dragons: The Use of a Fantasy Game in the Psychotherapeutic Treatment of a Young Adult. *American Journal of Psychotherapy*, 48 (4), págs. 624-632.
- [5] James L. Carroll, Paul M. Carolin. Relationship between Game Playing and Personality. *Psychological Reports*, junio de 1989, págs. 705-706.
- [6] Grey Ghost Press, Inc. *Fudge – A Roleplaying Game*, <<http://www.fudgerpg.com>>.
- [7] Pablo Giménez. *Los Juegos de Rol: Hacia una propuesta pedagógica*. <<http://dreamers.com/defensadelrol/articulos/propuesta.htm>>.
- [8] ¡Hola, mundo! Sitio web preliminar. <<http://www.agustincernuda.info/holamundo.html>>.
- [9] Michael C. Loui. Educational Technologies and the Teaching of Ethics in Science and Engineering. *Science and Engineering Ethics*, nº 11 (2005), págs. 435-446.
- [10] Mario de Miguel Díaz (Dir.). *Modalidades de enseñanza centradas en el desarrollo de competencias - Orientaciones para promover el cambio metodológico en el espacio europeo de educación superior*. Programa de estudios y análisis destinado a la mejora de la calidad de la Enseñanza Superior y de la actividad del profesorado universitario, proyecto EA22000055-00111188, diciembre de 2005.
- [11] Emily Oh Navarro, Emily, André van der Hoek. Comprehensive Evaluation of an Educational Software Engineering Simulation Environment. *20th Conference on Software Engineering, Education and Training (CSEE&T)*, julio de 2007, Dublín, Irlanda. IEEE Computer Society.
- [12] *Studies About Fantasy Role-Playing Games*, <<http://www.rpgstudies.net>>.

### Nota

<sup>1</sup> Nótese en la descripción de Dir. que incorporan, en sí mismas, un claro "autorretrato" de los estudiantes.

Joan Baiget Solé

Profesor de Teoría y Gestión del Conocimiento en la Universidad Autónoma de Barcelona; Socio Senior de ATI

<jbaiget@uoc.edu>

# Datos, conocimiento, información... en este orden

Estamos acostumbrados a oír hablar de datos, información y conocimiento como una secuencia lógica, difícilmente intercambiable. Y ello desde más de una perspectiva, no exentas de fundamento y de lógica. Pero no es necesariamente ésta la única manera de afrontar la relación entre dichos términos. Existen otras posibilidades como intentaremos argumentar a continuación.

Datos, Conocimiento e Información, en este orden. ¿Por qué?

Los *datos* son las cosas que suceden en el mundo real. *Datum* viene del latín, "lo que nos es dado", es decir, los hechos, lo que ha sucedido. Con nuestra capacidad de experimentar la realidad lo que hacemos es tener *conocimiento* de lo que ha sucedido, de lo que "nos es dado", de los hechos. Y es sólo después de conocer esos datos que poseemos la capacidad de *informar* (del latín *informare*), es decir, de darles forma (ver **figura 1**).

El conocimiento es la manera como accedemos a la realidad de las cosas. Conocer es acceder a una parte de la realidad (¿o es que podemos "conocer" algo que no existe?). Pero, ¿cómo se realiza el acto de conocer? Pues bien, algo sucede y somos capaces, a través de nuestros sentidos, de experimentarlo, puesto que no podemos experimentar algo que no ha sucedido ni algo que esté fuera del alcance de nuestros sentidos.

De manera que el "mundo real" queda grabado en nuestras mentes por la experiencia y representa un conocimiento de esa realidad que sólo nosotros poseemos, y que podemos complementar con el pensamiento. Pensando (razonando por ejemplo) podemos acceder a entender otros aspectos de la realidad que la experiencia aún no nos ha revelado, para luego, con nuestras hipótesis en mano, llevar

**Resumen:** frente al discurso predominante de situar datos-información-conocimiento como una secuencia lógica, contraponemos una aproximación de datos-conocimiento-información, en este orden, que justifica el entendimiento de los datos como elementos del "mundo real", el conocimiento como característica esencial de un "sujeto pensante" que lo percibe y la información como elemento principal de un "mundo simbólico" que es creación del sujeto pensante y que está compuesto por documentos que representan, escenifican o proyectan, el mundo real.

**Palabras clave:** conocimiento, conocimiento como criterio, conocimiento como stock, datos, documentación, escenarios del conocimiento, Gestión del Conocimiento, información, mundo real, mundo simbólico, sujeto pensante.

a cabo un proceso experimental que nos confirmará (o no) si ese presunto conocimiento de cómo es la realidad (inferido a través de la razón) es o no verdadero.

En términos de *input-output*, "experimentar" es un *output* del mundo real y un *input* en nuestra mente (sujeto pensante), algo que nos llega del exterior. *Pensar*, en cambio, es un *output* y un *input* "de" y "en" nosotros mismos, ya que el pensamiento empieza y acaba en nuestra mente, y nada sale de nosotros. Cuando queremos comunicar nuestros pensamientos al "exterior", decir cómo vemos el mundo, lo que hacemos es *enseñar* (en sentido amplio), es decir, mostramos lo que hay en nuestro interior, mediante el lenguaje, los signos, la palabra escrita, una grabación, una pintura, etc., mediante un documento en definitiva. El documento pasa a formar parte de un mundo simbólico que representa al mundo real. Y hacemos esto (el "enseñar") para que alguien pueda *aprender* (en sentido amplio) desde ese mundo simbólico que hemos creado, sin necesidad de pasar por los caminos de la experiencia (mundo real) y del pensamiento (sujeto pensante) que nosotros ya hemos recorrido. Finalmente, podemos *aplicar* nuestros conocimientos, de nuevo, al mundo real, transformándolo y dejando en él la huella de éstos.

Estas 5 fases (Experimentar, Pensar, Enseñar, Aprender, Aplicar) representan los "Escenarios del Conocimiento"<sup>1</sup> (ver **figura 2**) y sirven para indicar que el *mundo real* está representado (en parte) en cada *sujeto pensante*, de manera individual e intransferible, en su mente (conocimiento orgánico) y de manera colectiva (también en parte) en un *mundo simbólico* (conocimiento inorgánico) de conocimientos explicitados en soportes (libros, imágenes, grabaciones...).

Los hechos suceden (datos), se conocen (conocimiento) y se informan (información). De manera que el ámbito de los datos, corresponde al "mundo real". El ámbito del conocimiento corresponde a las personas ("sujeto pensante") y el ámbito de la información corresponde a los documentos ("mundo simbólico").

Así, si me encuentro en una estación y he visto salir un tren a las 10h, tengo el conocimiento directo de un dato, de algo que ha sucedido (mundo real). Si en esa misma estación leo un panel informativo donde se dice que a las 10,30h saldrá otro tren, tengo el conocimiento de una información que alguien (con conocimiento de otros datos) ha informado (mundo simbólico). Luego puedo pensar ("sujeto pensante") que saldrá otro tren en el



Figura 1. Secuencia lógica del enfoque expuesto en este artículo.

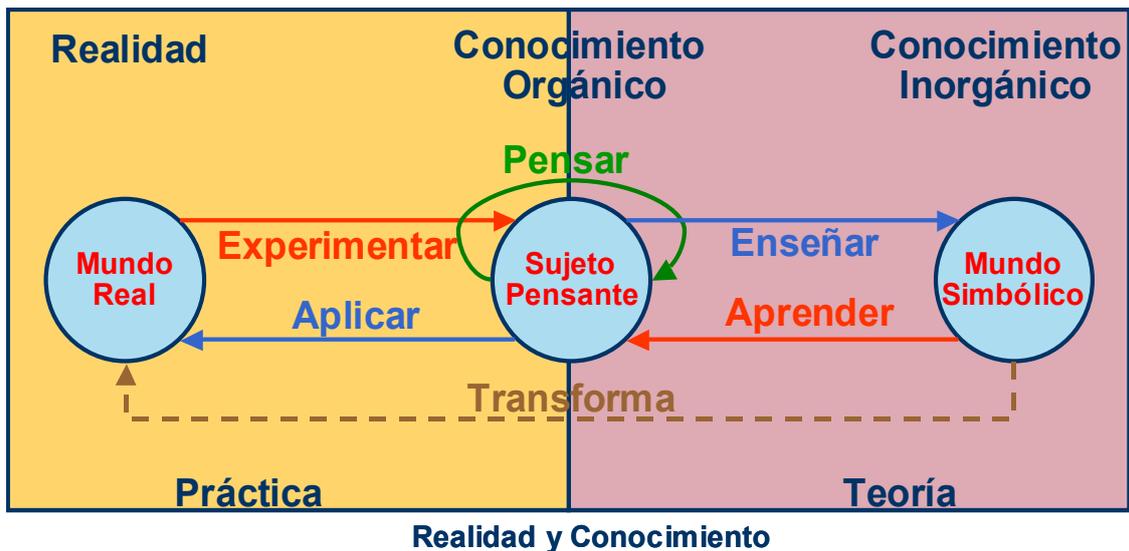


Figura 2. Las cinco fases que representan los escenarios del conocimiento.

mismo intervalo de media hora, a las 11h, y entonces tengo un conocimiento hipotético. El primero es un conocimiento cierto, el segundo y el tercero son sólo probables, porque aún no han sucedido.

En general, podemos decir que nuestro *conocimiento como stock* es el conjunto de nuestro saber directo del mundo ("experimentado") más nuestro saber indirecto del mundo ("aprendido"), junto a nuestro saber inferido ("pensado"), y nuestro *conocimiento como criterios* es el resultado de un número indefinido de factores (biológicos, genéticos, psicológicos, físicos, sociales...) que interactúan con nuestro stock de conocimientos y que, entre ambos, justifican nuestras decisiones. Un gran stock de conocimiento no implica actuar con mejor criterio y hay que ser consciente que parte de nuestro conocimiento "aprendido" y "pensado" puede no ser verdadero.

**La información es el conocimiento "de los otros".** En el Mundo Simbólico a donde acudimos para "aprender" ("informarnos"), todo aquello que nos informa de que algo ha sucedido, o va a suceder, es el resultado de que "otro" ha conocido unos datos y los ha informado, es decir, les ha dado forma gracias a su conocimiento.

Por ello, **la verdad depende de la correcta propagación del conocimiento.** Que la manera en que se informen los datos se corresponda con la realidad o no, dependerá que se extienda, o no, la mentira (desinformación) en nuestra sociedad y que nuestro conocimiento resultante sea, o no, verdadero.

**Nota**

<sup>1</sup> Ver artículo "Escenarios del Conocimiento", Joan Baiget 2005. <<http://eprints.rclis.org/archive/00008474/>>.

Ricardo Navarro Moral, Francisco R. Feito Higuera, Rafael J. Segura Sánchez, Ángel L. García Fernández  
 Universidad de Jaén

<ricardonavarromoral@gmail.com>  
 <{ffeito, rsegura, algarcia}@ujaen.es>

### 1. Avances en geoinformación

Si observamos los últimos avances en el mundo de la Informática, los Sistemas de Información Geográfica (SIG) y demás herramientas informáticas aplicadas al tratamiento de la geoinformación son uno de los temas más punteros y de mayor actualidad.

En un principio, estas herramientas trabajaban sobre elementos bidimensionales para presentar la geoinformación (planos y capas 2D). Con el paso del tiempo están empezando a adquirir elementos más complejos como puede ser la combinación de modelos 3D muy realistas que integran imágenes del entorno real y/o grabaciones. Entre este tipo de herramientas destacan fundamentalmente las de los dos gigantes de la informática: Google Earth y Microsoft Virtual Earth. Tanto una aplicación como la otra están integrando cada vez más elementos para que su representación virtual de terrenos sea más realista.

A continuación pasamos a describir algunos de ellos:

# Tratamiento de geoinformación a nivel peatonal: animación 3D a partir de metainformación Exif

**Resumen:** en la actualidad, las aplicaciones informáticas que tratan información geográfica están teniendo un gran éxito y acogida en el mercado. Algunas de las más conocidas, como por ejemplo Google Earth o Microsoft Virtual Earth, ofrecen un gran potencial a la hora de realizar sobrevuelos y vistas generales. El problema de estas aplicaciones aparece a la hora de tratar los datos a nivel peatonal, ya que surgen problemas tales como la lentitud en la carga de texturas y edificios, falta de realismo (proporciones inapropiadas de los edificios con respecto al peatón), detección de colisiones, etc. En este artículo se pretende dar mayor realismo al tratamiento de la geoinformación a nivel peatonal. Para ello se ha desarrollado una aplicación que, a partir de fotos georreferenciadas, permite generar en un modelo 3D (que represente al entorno) trayectorias y/o videos que siguen aproximadamente el mismo camino que se ha utilizado para capturar dichas fotos. La aplicación se basa en tres pilares fundamentales: animación 3D, modelos 3D y metainformación Exif.

**Palabras clave:** animación 3D, Exif, geoinformación, metainformación peatonal, modelo 3D.

#### ■ Fotos reales georreferenciadas en Panoramio

Panoramio es un sitio web dedicado a exhibir las fotografías de paisajes que los propios usuarios crean y georreferencian. Las imágenes que cumplen ciertos requisitos pueden ser vistas a través Google Earth. El objetivo de Panoramio es permitir a los usuarios de esta aplicación aprender más sobre una zona espe-

cífica del mundo, observando las fotografías que otros usuarios han tomado ahí (ver **figura 1**).

■ Bird's Eye de Microsoft Virtual Earth  
 Esta herramienta permite navegar por imágenes aéreas estáticas del mundo real, ofreciendo la posibilidad de cambiar el punto de vista (ver **figura 2**).

■ Modelos 3D de Google Earth



Figura 1. Fotografía encontrada en Panoramio.

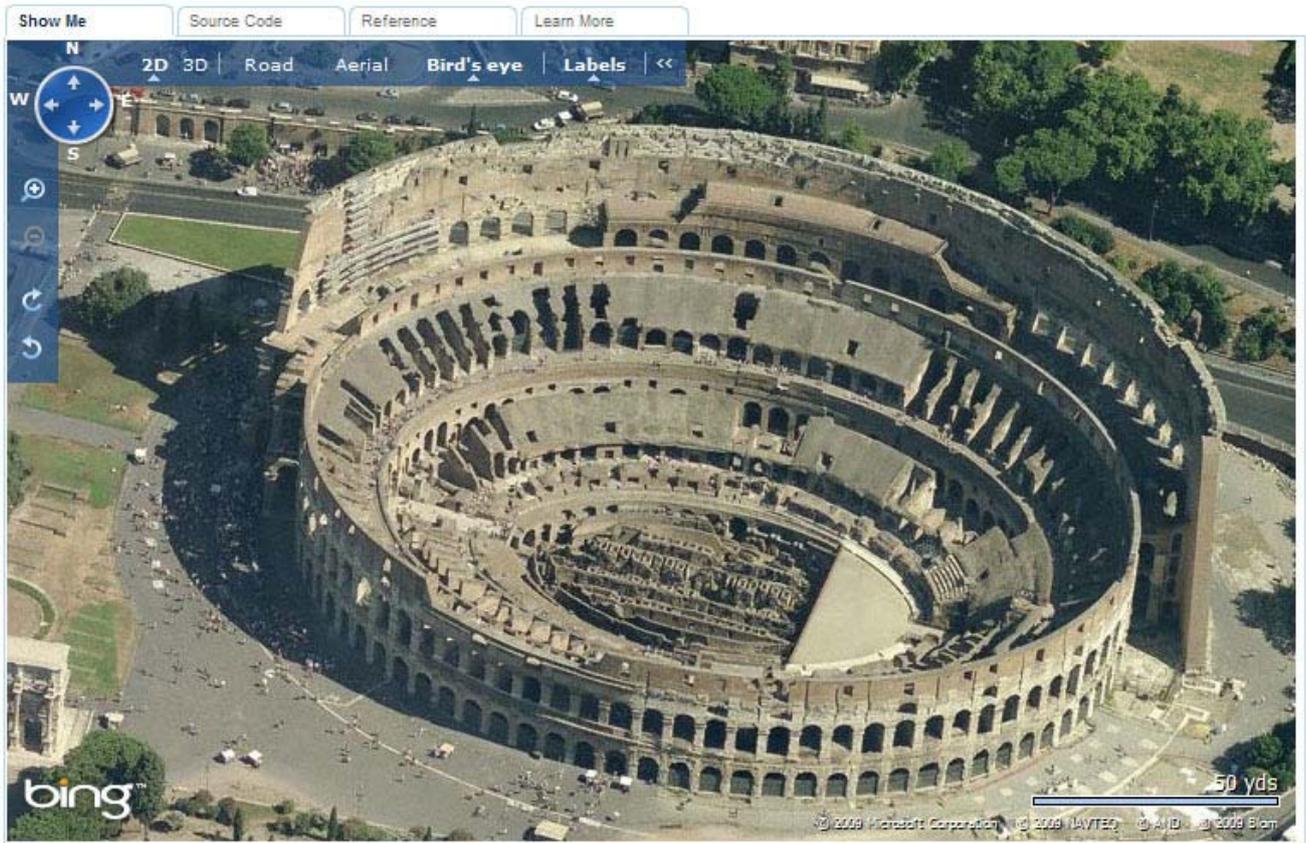


Figura 2. Imagen aérea en Microsoft Virtual Earth.

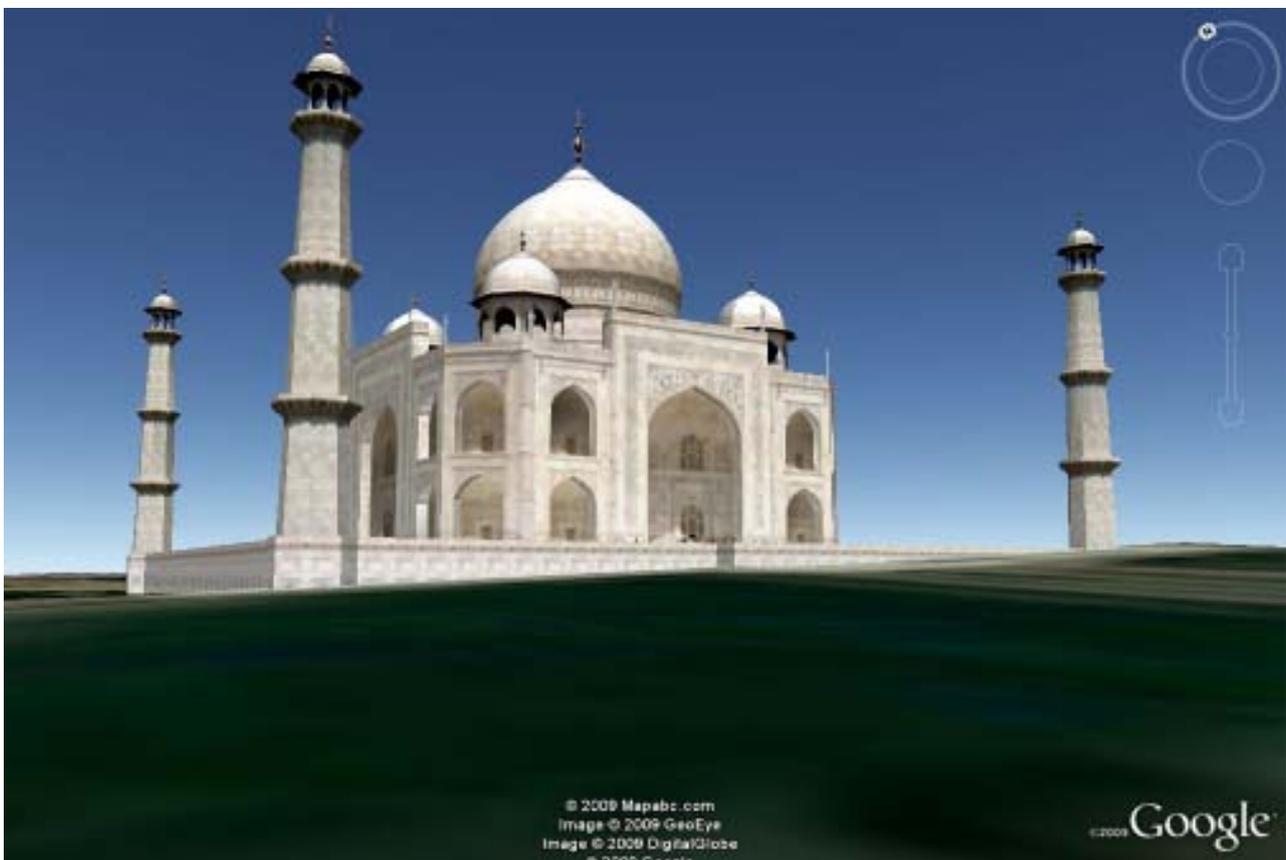


Figura 3. Modelo 3D en Google Earth.



Figura 4. Navegación a través de vistas panorámicas reales.



Figura 5. Información EXIF de una imagen tomada con una cámara GPS.

Google Earth integra modelos en tres dimensiones de edificaciones, monumentos, museos, etc. Estos modelos se desarrollan mediante una herramienta gratuita (aunque también está disponible una versión de pago con mayores funcionalidades) y de uso sencillo denominada SketchUP. De este modo, se permite al usuario desarrollar sus propios modelos 3D, junto con la posibilidad de compartir modelos y georreferenciarlos, lo cual dota a Google Earth de mayor realismo (ver figura 3).

■ Google Street View

Google Street View permite navegar a través de vistas panorámicas reales de 360 grados tomadas a intervalos regulares por las calles de una ciudad (ver figura 4).

A pesar de todos estos avances y de las nuevas herramientas que van apareciendo, no se consiguen buenos resultados a pie de calle, es decir, a nivel peatonal. Algunos de los temas que presentan mayor problema en este ámbito son la lentitud de carga de la información visual (sobre todo con las texturas de edificios, terrenos...) y el realismo a la hora de la navegación a nivel peatonal (edificios y entornos con proporciones reales, colisiones con los edificios, interacción con los objetos del entorno...). También es cierto que herramientas como Google Earth y Microsoft Virtual Earth no tienen como prioridad obtener buenos resultados a nivel peatonal, sino que centran su objetivo en obtener vistas aéreas más generales, realistas y optimizadas.

### 2. Imágenes georreferenciadas y el formato Exif

Hace unos años, una de las herramientas básicas del fotógrafo era su cuaderno en el que apuntaba donde se había realizado cada foto, con qué objetivo, qué tipo de película, velocidad de disparo, distancia focal etc. Con la llegada de las cámaras digitales se pierde esa costumbre gracias a la aparición del formato Exif.

*Exchangeable Image File Format* (abreviatura oficial Exif, no EXIF) [1] es una especificación para formatos de archivo de imagen usados por las cámaras digitales. Fue creado por la *Japan Electronic Industry Development Association* (JEIDA).

El formato Exif permite almacenar dentro del fichero, junto con la imagen capturada, información relativa a ésta como puede ser el nombre del fichero, el tipo de cámara y entre otras cosas la fecha de disparo. Esta información almacenada recibe el nombre de metadatos ó metainformación [2].

De este modo, si se dispone de una cámara compatible con el formato Exif y equipada con GPS (*Global Positioning System*), las imágenes capturadas podrán almacenar los datos correspondientes a la localización del lugar de captura de la imagen: longitud, latitud, altitud..., es decir se georreferencian las imágenes capturadas.

En la **figura 5** se muestra un ejemplo de la información Exif almacenada en una imagen tomada con una cámara con GPS.

### 3. Propuesta de aplicación

Supongamos que se dispone de una cámara con GPS y que se han capturado con ella fotos siguiendo una ruta. Supongamos también que se tiene un modelo 3D del entorno en el que se han capturado dichas fotos. A partir de los conceptos de animación [3], modelos 3D e imágenes georreferenciadas surge la idea de crear una aplicación que, a partir de fotos georreferenciadas, sea capaz de generar animaciones a través de un modelo 3D del entorno que sigan una ruta aproximada a la que se ha utilizado para capturar dichas imágenes. Dichas imágenes han de ser capturadas a ras de suelo ya que el objetivo es reproducir rutas peatonales.



Figura 6. Fotografía del edificio A4 de la Universidad de Jaén.

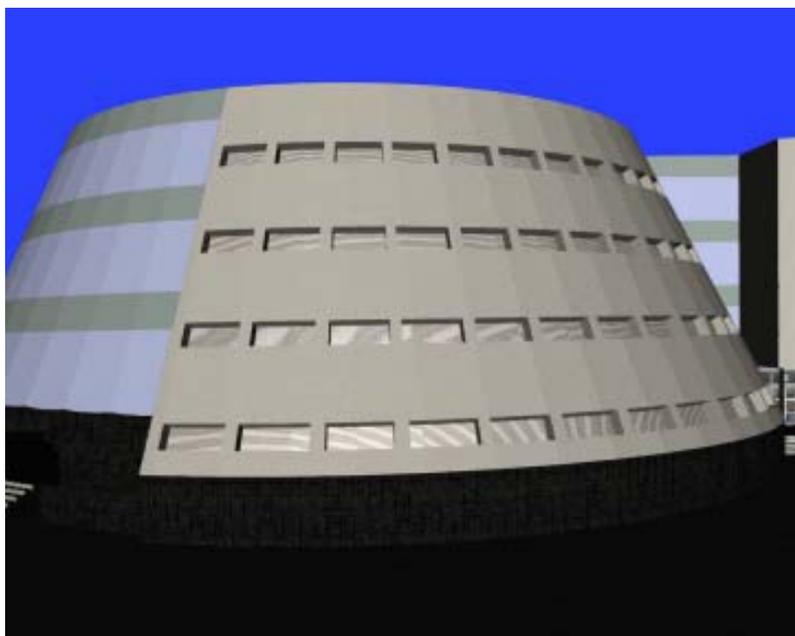


Figura 7. Modelo 3D del edificio A4 de la Universidad de Jaén.

Recientemente, junto con Google Earth 5.0, Google ha añadido una extensión al lenguaje KML (*Keyhole Markup Language*) para la creación de scripts de animación [4]. La aplicación desarrollada ofrece la posibilidad de generar un fichero KML compatible con Google Earth. A partir de este fichero,

Google Earth será capaz de generar la animación de la ruta seguida en la aplicación desarrollada.

En las siguientes secciones se explicará más detalladamente cómo se ha creado dicha aplicación.



Figura 8. Navegación a través de modelos 3D.

**3.1. Creando un mundo virtual**

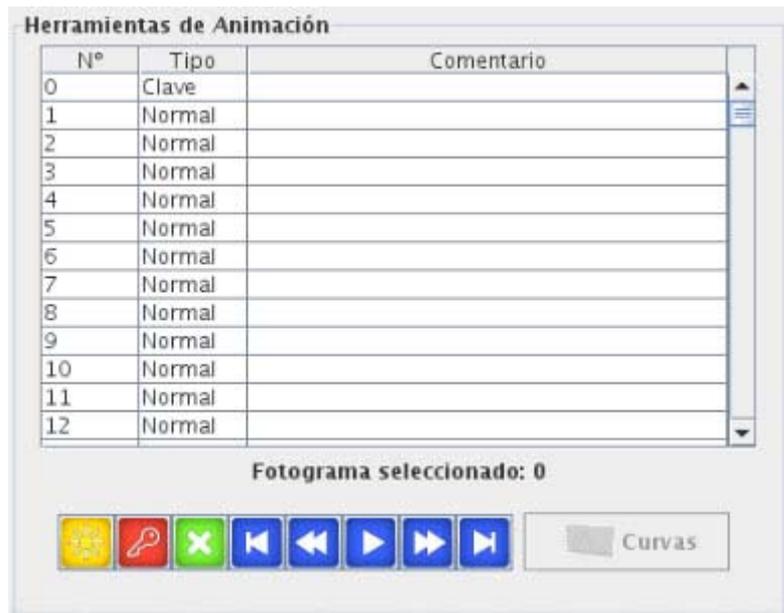
Para utilizar la aplicación en cuestión es fundamental disponer de un modelo 3D virtual que simule el entorno donde se han tomado las fotografías. En la red se pueden encontrar muchos modelos 3D ya creados de edificios de ciudades reales [5]. Otra opción es modelar el entorno de manera manual mediante herramientas tales como 3DS, Maya, Blender o SketchUp.

La **figura 6** muestra un ejemplo de una foto del mundo real (edificio A4 de la Universidad de Jaén) mientras que la **figura 7** muestra un modelo 3D de la misma imagen.

Los datos del modelo deben estar georreferenciados. El sistema de referencia de las fotos (sistema de coordenadas de mundo) tiene que ser igual o fácilmente convertible al sistema usado para georreferenciar el modelo (sistema de coordenadas de escena). En el caso concreto de esta aplicación, el modelo tiene su propio sistema de coordenadas, de manera que a la hora de posicionarnos en un punto del modelo a partir de una imagen se realiza un cambio de coordenadas de mundo a coordenadas de escena.

**3.2. Movimiento y animación a través del mundo virtual**

Una vez se tiene el modelo 3D es necesario integrarlo en un entorno gráfico que permita navegar a través de él, es decir, se necesita una cámara virtual [6] a través de la cual se verá el mundo 3D en un dispositivo de visualización adecuado.



**Figura 9.** Creación de animaciones a partir de fotogramas clave.

Este entorno gráfico se ha desarrollado utilizando el lenguaje de programación Java y su librería Java3D así como una librería de clases para animación 3D de autoría propia. La cámara tiene la posibilidad de moverse libremente a través del modelo 3D y por tanto obtener diferentes imágenes del mismo (ver **figura 8**).

También se ha añadido una paleta de animación que permite establecer como fotogramas clave posiciones de la cámara, es decir, instantáneas del modelo 3D en un momento deter-

minado. La aplicación es capaz de crear animaciones de cámara a partir de estos fotogramas clave (calculando los *inbetweens* o fotogramas intermedios). Los fotogramas intermedios se obtienen mediante interpolación, la cual es calculada haciendo uso del interpolador KBRotPosScaleSplinePathInterpolator [7] de Java 3D que permite realizar interpolaciones creando un *spline* a partir de una serie de puntos clave (ver **figura 9**).

**3.3. Integrando imágenes y su metainformación Exif**

También ha sido necesario añadir al entorno gráfico alguna manera de interactuar con las imágenes capturadas mediante la cámara con GPS y la metainformación de las mismas (ver **figura 10**). Para ello se ha acoplado al entorno gráfico la capacidad de abrir y visualizar imágenes, así como mostrar la metainformación Exif de cada una de ellas (latitud, longitud, nombre, dimensiones, etc.).

Una función fundamental para nuestro propósito, es la funcionalidad de situar la cámara virtual (a través de la cual visualizamos el modelo 3D) en función de la información GPS que nos proporciona la metainformación Exif de las imágenes capturadas. Para ello la aplicación es capaz de convertir las coordenadas obtenidas por el GPS de la cámara (utilizando el *datum* WGS84) a coordenadas UTM [8], ya que normalmente los modelos 3D están expresados en coordenadas cartesianas (en este caso se utilizará el *datum* ED50). Esta conversión es realizada por la aplicación de manera transparente para el usuario, aunque es necesario señalar que se pueden producir pequeñas imprecisiones en la conversión derivadas de la propia naturaleza del *datum* ED50 (este aspecto se solucionará



**Figura 10.** Interacción del entorno gráfico con las imágenes capturadas.

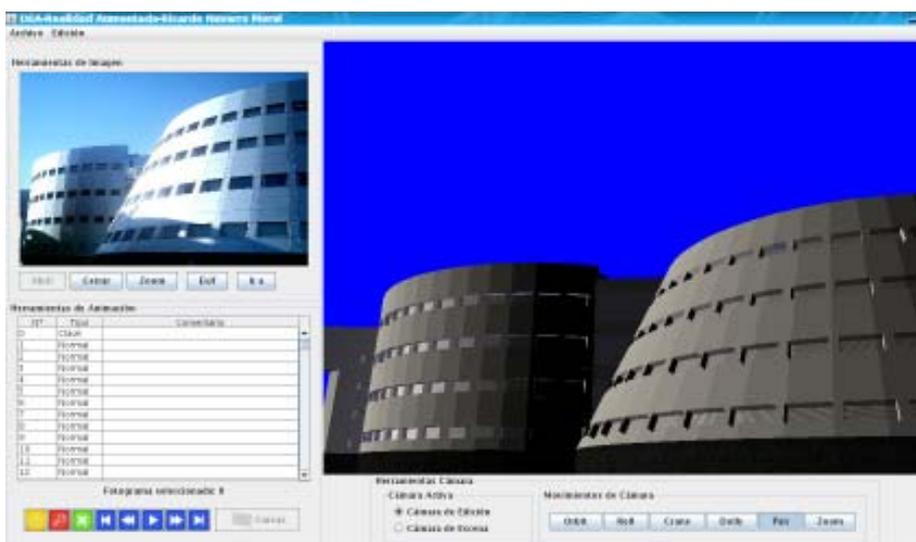


Figura 11. Visión general de la aplicación desarrollada.

con la adaptación en España del datum ETRS89) [9].

Una vez que se tiene la cámara posicionada, el usuario será el encargado de orientarla gracias a los movimientos de cámara proporcionados por la aplicación, ya que el formato Exif no incluye los datos de orientación, sino solo los de posición.

### 3.4. Fundamentos de la aplicación

Para crear la animación bastará con seguir unos sencillos pasos para cada imagen capturada:

- Abrir una imagen (que ha sido capturada con la cámara con GPS).
- Posicionar la cámara virtual en función de la metainformación Exif de la imagen abierta en ese momento (esto se hace con la pulsación de un botón).
- Orientar la cámara virtual (ya posicionada) mediante las herramientas proporcionadas por el entorno gráfico.
- Marcar como fotograma clave el estado actual de la cámara.
- Una vez marcados todos los fotogramas claves se puede previsualizar la animación.
- Una vez creada la animación, se puede crear un fichero KML con sus parámetros para posteriormente poder ser generada y visualizada en Google Earth.

En la figura 11 mostramos una visión general de la aplicación desarrollada.

### 4. Conclusión y posibles mejoras

Se ha presentado una aplicación que, a partir de fotos georreferenciadas, es capaz de generar animaciones a través de un modelo 3D del entorno que sigan una ruta similar a la que se ha utilizado para capturar dichas imágenes.

Mediante la aplicación desarrollada se intenta aportar mayor realismo en el nivel peatonal a herramientas que tratan la geoinformación. Seguramente, con el paso del tiempo, herra-

mientas como Google Earth o Microsoft Virtual Earth integrarán novedades similares a las desarrolladas en esta aplicación.

Una de las grandes mejoras que se podría integrar en esta aplicación sería la posibilidad de obtener en el modelo 3D la orientación de la imagen capturada. Esto se podría conseguir mediante el uso de alguna de las siguientes técnicas: búsquedas en bases de datos de imágenes ó técnicas para el uso de dispositivos con acelerómetro [10] y/o giroscopio [11] para la toma de imágenes.

La primera técnica consistiría en comparar la imagen capturada con las imágenes almacenadas en bases de datos que tuvieran la misma metainformación Exif de posición. De esta manera, la imagen almacenada que más se aproximase a la capturada determinaría la orientación de esta última.

Por otro lado, la segunda opción es más compleja. Algunas cámaras integran acelerómetros y/o giroscopios, los cuales detectan y corrigen los efectos de los ligeros movimientos de la cámara, que pueden dar lugar a fotos movidas. Trabajando sobre estos dispositivos y la detección de estos movimientos se podría obtener la orientación de la cámara en el momento de la captura de la imagen y así mismo incorporarlo a la metainformación Exif.

Otra posible mejora podría consistir en incorporar al modelo una iluminación que dependiera de la fecha y hora de la toma de la imagen. De este modo, las imágenes capturadas y la visión equivalente obtenida en el modelo tendrían mayor similitud y por tanto más realismo.

## Referencias

- [1] Exif.org. <<http://www.exif.org>>.
- [2] DMCI Usage Board. *Dcmi Metadata Terms* (2006). <<http://dublincore.org/documents/dcmi-terms/>>.
- [3] J. Lasseter. Principles of traditional animation applied to 3D computer animation. *Computer Graphics*, 4 (21), pp: 35-44, 1987.
- [4] Google code. *Touring in KML*. <<http://code.google.com/intl/en/apis/kml/documentation/touring.html>>.
- [5] Google Galería 3D. <<http://sketchup.google.com/3dwarehouse/>>.
- [6] J. Foley, A. Van Dam, S. Feiner, J. Hughes. *Computer Graphics: Principles and Practice in C*. Addison-Wesley Professional, 1995. ISBN-10: 0201848406.
- [7] java.net. *Class KBRotPosScaleSpline PathInterpolator* <<http://download.java.net/media/java3d/javadoc/1.3.2/com/sun/j3d/utills/behaviors/interpolators/KBRotPosScaleSplinePathInterpolator.html>>.
- [8] Steven Dutch. *Converting UTM to Latitude and Longitude (or Vice Versa)*. Universidad de Wisconsin, Departamento de Ciencias Naturales y Aplicadas. <<http://www.uwgb.edu/dutchs/UsefulData/UTMFormulas.htm>>.
- [9] F.J. González-Matesanz, R. Quirós, J.A. Sánchez Sobrino, A. Dalda. El cambio de Datum ED50-ETRS89: Métodos y Resultados. *Topografía y Cartografía. Volumen XXII, Nº 130 Septiembre-October 2005*, pp 3-10. Colegio Oficial de Ingenieros Técnicos en Topografía, Madrid.
- [10] James J. Brophy. *Electrónica fundamental para científicos*. Editorial Reverte S.A., 1979.
- [11] Ferdinand P. Beer. *Mecánica vectorial para ingenieros*. McGraw Hill, 2005.

Fecha de revisión de los enlaces web: 17/2/2009.

Ángel Hernández García,  
Santiago Iglesias Pradas,  
Julián Chaparro Peláez,  
Félix Pascual Miguel  
*Universidad Politécnica de Madrid*

<{angel.hernandez,s.iglesias,julian.chaparro,  
felixjose.pascual}@upm.es>

# Influencia en el rendimiento académico de la interacción en línea de los alumnos: estudio y análisis comparativo entre diferentes modalidades de enseñanza

## 1. Introducción

En los últimos años se ha producido a escala mundial una gran expansión de la educación online, entendida ésta como el proceso docente de teleformación realizado a través de una plataforma tecnológica basada en Tecnologías de la Información y Comunicaciones (TIC), propiciada fundamentalmente por las ventajas que ésta ofrece, como la independencia de la ubicación espacial y temporal (en este último caso salvo en entornos de participación síncronos), la variedad de fuentes de conocimiento al alcance de profesores y alumnos, la reducción de los costes asociados al proceso formativo y la potenciación de la participación por parte de todos los agentes que intervienen en el proceso docente [1].

No obstante, también existen una serie de inconvenientes asociados a la teleformación online, como la necesidad de poseer un ordenador con conexión a Internet y las habilidades necesarias para utilizar las herramientas tecnológicas disponibles, así como la curva de aprendizaje de cada plataforma en concreto. Además, el estudio en solitario requiere de unos hábitos de estudio y un grado de motivación para la realización de los cursos adecuados, así como de la interacción con el cuerpo docente para evitar que el proceso de aprendizaje resulte un fracaso.

Así pues, si se desea el éxito del proceso docente en e-learning uno de los principales objetivos de cualquier curso en línea, partiendo de un cuidado y claro diseño, planificación e implementación del mismo debe pasar por su uso continuado y la interacción por parte de los alumnos con la plataforma, potenciando todos los aspectos relacionados con esta interacción tanto entre los alumnos como con los contenidos y el profesorado.

## 2. Marco teórico

### 2.1. Interacción en entornos de teleformación online

Dentro de la enseñanza online se puede apreciar una evolución que tiende a centrar el proceso a través de la plataforma en el estudiante [2], lo cual se ha visto reflejado en la naturaleza de las plataformas de teleformación, que han pasado de ser un mero repositorio de información a introducir me-

Este artículo ha sido seleccionado entre los mejores trabajos presentados en X Simposio Internacional de Informática Educativa (SIIE' 2008) que se celebró en Salamanca en el mes de Octubre de 2008.

**Resumen:** *a lo largo de los últimos años, se ha producido un aumento en el uso de sistemas de teleformación para la impartición de cursos a distancia, reflejado tanto en la aparición de universidades virtuales como en la virtualización de materias en las universidades tradicionales. Una de las mayores preocupaciones a la hora de impartir cursos a través de una plataforma de teleformación es conseguir una experiencia docente que resulte en un mayor aprovechamiento del curso por parte de los alumnos, para lo cual se han de determinar los factores que influyen en el rendimiento académico. En el presente estudio se plantea esta pregunta en base a factores objetivos de interacción entre el alumno y el sistema (en concreto, el número de páginas y recursos visitados, el número de visitas e intervenciones en foros, y las intervenciones en chats), llegando a la conclusión de que no existe una relación significativa entre estos parámetros y el rendimiento académico, medido como la nota final alcanzada en el curso. Además, se plantea un análisis comparativo entre cursos similares llevados a cabo en modalidad en línea y modalidad presencial, como medio para determinar la variación entre ambos tipos, con independencia de los contenidos docentes, tanto de la influencia de los factores de estudio como del rendimiento académico final por parte del alumno. En este análisis, la conclusión alcanzada es que los cursos online permiten a los alumnos un mejor aprovechamiento del curso en términos de los resultados finales alcanzados, no existiendo relación de los factores de interacción con el rendimiento académico en ninguna de las dos modalidades de enseñanza.*

**Palabras clave:** *e-learning, indicadores de interacción en línea, rendimiento académico.*

canismos de participación y seguimiento activo del desarrollo de los cursos [3] con el fin de obtener una mejora sustancial de la calidad de la enseñanza, los conocimientos obtenidos y la interacción (tanto alumno-compañeros como alumno-plataforma, alumno-contenidos docentes y alumno-profesorado).

En este contexto, tradicionalmente ha existido una gran preocupación sobre los factores que influyen en el rendimiento académico en entornos de enseñanza telemática a distancia, de forma que tanto para los cuerpos docentes como para los creadores de plataformas de teleformación fuera posible potenciar los aspectos con una influencia positiva en el rendimiento académico, y minimizar (o incluso neutralizar) las variables con efecto negativo.

Si bien existe un gran número de factores subjetivos de índole psicológica o social, como la motivación del alumno para la realización del curso [4], o la utilidad futura del mismo para el desempeño profesional, o bien otros factores de difícil cuantificación como la calidad de los contenidos docentes, a nivel objetivo uno de los principales factores señalados reside en el grado de participación e

implicación de los alumnos en el curso [5][6], generalmente medido a través de cuantificadores objetivos como el número de correos electrónicos intercambiados, el número de participaciones en los foros de discusión o el tiempo de conexión del alumno a la plataforma, o a través de cuantificadores subjetivos como la calidad de estas interacciones [7]. En los estudios empíricos realizados hasta la fecha no se ha detectado una gran influencia de la interacción vía correo electrónico aunque sí se ha destacado la importancia del tiempo y hábitos de dedicación al curso [8], el tiempo de uso [9] y el número de accesos al sistema [10] [11] así como la doble relevancia de las intervenciones en foros de discusión: por un lado, como instrumento para asentar y ampliar los conocimientos adquiridos, y por otra parte como pilar en la creación de comunidades de aprendizaje [12].

Pese a tratarse de un tema de estudio relativamente novedoso, todos los estudios existentes parecen coincidir en la idoneidad de emplear las herramientas de registro propias de las plataformas de teleformación en lugar del empleo de programas de registro de acceso web generales a la hora de obtener registros

apropiados para el análisis de la influencia de las interacciones en línea con el rendimiento académico [13][14][15][16].

No obstante lo anterior, algunos autores alegan la idoneidad de acompañar estos análisis con mecanismos de realimentación para los alumnos como medio para corregir deficiencias derivadas de las características de aislamiento y planificación personal propias de estos sistemas de teleformación. Así, por ejemplo, se considera adecuado el incluir comentarios sobre la realización de las tareas por parte de los alumnos, principalmente cuando éstos tienden a reforzar los puntos en los que se considera que se ha realizado un aprendizaje efectivo, más que los que tienen naturaleza puramente correctiva [17]. Estas interacciones del alumno con el sistema y el profesorado, por tanto, son consideradas como los mayores indicadores del resultado del aprendizaje efectivo por parte del alumno, sin despreciar las mejoras que introducen en la formación de comunidades de aprendizaje (o interacción entre alumnos) [13].

Finalmente, entre los factores que pueden influir en este tipo de análisis se han señalado como más importantes la experiencia previa en el uso de estos sistemas, como elemento positivo cara al rendimiento final [18], pudiendo en algunos casos condicionar los resultados (se ha detectado que los usuarios más experimentados necesitan de menos interacciones con la plataforma [19]); también existe un fenómeno, reseñado en [20] y [21] y conocido como el de “testigo de aprendizaje”, que puede introducir problemas de análisis a la hora de medir la interacción en foros. Este fenómeno consiste en la existencia de una serie de alumnos con gran capacidad de aprendizaje y gran rendimiento, y que sin embargo apenas realizan aportaciones en los foros, limitándose a cumplir con los mínimos establecidos y a leer las aportaciones de los compañeros.

### 2.2. Enseñanza en línea vs. enseñanza presencial

Otra de las preocupaciones fundamentales a la hora de tomar la decisión de impartir un curso de forma presencial o en línea viene determinado por la adecuación de las plataformas de teleformación para la enseñanza de estos cursos, así como las mejoras que puede introducir el uso de estos sistemas de cara al aprendizaje final por parte de los alumnos.

Sin embargo, en la literatura revisada no parecen existir elementos de consenso que permitan llegar a ninguna conclusión válida, ya que si bien en [22] se indica que los resultados son mejores en los cursos online, [18][23][24] y [25] revelan que no hay diferencias significativas, aunque el grado de interacción es superior en entornos presenciales [18] y existe una mayor tasa de abandonos

en los entornos de teleformación online [25]. Otros autores indican que los sistemas de teleformación no pueden sustituir pero sí complementar a los entornos presenciales tradicionales [26]; en este sentido, si parece existir un acuerdo en la conveniencia de emplear herramientas de teleformación apropiadas (LMS, o *Learning Management Systems*) en lugar de ofrecer simplemente los contenidos docentes en red [27].

### 3. Metodología de estudio

#### 3.1. Descripción de los cursos

Para la realización de este estudio se han empleado datos de cinco cursos de distinta naturaleza: uno de ellos, un curso de grado de “Creación de empresas” de libre elección de la Escuela Técnica de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid realizado íntegramente en línea; dos cursos troncales pertenecientes al “Máster de Domótica Online” de la Universidad Politécnica de Madrid también impartidos de forma completa en línea; y los dos mismos cursos troncales correspondientes al “Máster de Domótica” impartidos en su modalidad presencial.

Se ha de notar que en el caso del máster presencial, pese a que la documentación se reparta impresa a los alumnos, también se encuentra disponible en la plataforma de teleformación creada a tal efecto. Asimismo, tanto las tareas como los tests son en los cinco cursos contestadas a través de la plataforma de teleformación.

La plataforma elegida para la impartición de los cursos es Moodle, debido tanto a su extensión y facilidad de uso (a todos los alumnos y profesores se les ha entregado un manual de uso antes de comenzar el curso, y se han establecido canales de resolución de dudas técnicas tanto vía correo electrónico como a través de foros dentro de la plataforma), como a los mecanismos de recogida de estadísticas de interacción que ofrece la plataforma.

En total se han recogido datos de 63 alumnos de los cinco cursos, graduados y posgraduados con una elevada formación en sistemas de información aunque neófitos en sistemas de teleformación, durante 8 semanas para el caso del segundo curso troncal de los máster y el curso de libre elección, y durante 2 semanas para el primer curso troncal de máster. La planificación de los cursos en todos los casos ha sido idéntica, planeando un calendario de activación de los contenidos docentes para las sucesivas unidades didácticas, e incluyendo la realización de tests, tareas individuales y casos en equipo. Además, para los cursos online se ha implementado un foro de noticias general (presente también en los cursos presenciales) y foros específicos en cada unidad, tanto de debates planteados por los

profesores (con impacto en la nota global de la unidad) como de tutoría y de dudas técnicas para realizar tareas de soporte a los alumnos.

Como herramienta de comunicación adicional, en los cursos de máster online se han programado sesiones de chat para proceder a la resolución de dudas, aclaración de conceptos relativos a cada unidad y discusión de los casos en equipo entregados.

Una vez realizadas las entregas de las tareas, se ha procedido a la corrección de las mismas por parte de los docentes, incluyendo los comentarios pertinentes relativos a las respuestas específicas de cada alumno.

Finalmente, se han realizado tareas de seguimiento consistentes en contactos vía telefónica o a través del correo electrónico con los alumnos en los que se han detectado anomalías de comportamiento (retrasos en entregas, ausencias de conexión prolongadas...) a fin de favorecer el seguimiento del curso.

#### 3.2. Objetivos del estudio

El objetivo principal del presente estudio es el análisis de las relaciones entre las interacciones de los alumnos con la plataforma de teleformación y el resultado académico final del alumno.

Las interacciones se han medido mediante el empleo de las herramientas de registro de la plataforma utilizada (Moodle), y son las referidas a las siguientes acciones:

- Número completo de páginas visitadas.
- Número de visitas (consultas e intervenciones) en los foros.
- Número de intervenciones en foros.
- Número de intervenciones en chats (si el chat está disponible).
- Número de lecturas de chats (informes, visitas pasivas).

La medida del rendimiento está determinada por la nota final del alumno.

Adicionalmente, y con el fin de poder determinar más precisamente la influencia del uso de sistemas de teleformación en línea en el rendimiento académico, independientemente de los contenidos docentes, se establece una comparación entre cursos impartidos completamente en línea a través del sistema de teleformación y sus equivalentes impartidos de manera presencial haciendo asimismo uso del sistema. Esta comparación busca discernir si existen factores de interacción cuya influencia se ve alterada al ofrecerse los contenidos en modo en línea, así como si el rendimiento académico final se ve afectado por la modalidad de enseñanza del curso.

#### 3.3. Metodología de análisis

Al igual que en estudios previos [8][10], se han aplicado técnicas de regresión lineal con el objeto de determinar la contribución de

cada una de las variables al resultado final. El método empleado ha sido el de regresión lineal múltiple hacia atrás, con el fin de poder determinar cuáles de los factores influyen significativamente en el rendimiento académico final del alumno. Este método permite establecer la importancia relativa de las variables independientes de forma predictiva de tal modo que la varianza de la variable dependiente se puede estimar con un nivel de significancia como la combinación lineal de la contribución específica (ponderada por un factor para cada componente) de cada una de las variables independientes [28].

De esta forma, se han empleado las 63 muestras recogidas, incluyendo las 2 correspondientes a alumnos que han comenzado el curso sin llegar a concluirlo, y se ha procedido a su análisis mediante el paquete estadístico de software SPSS 14.0.

#### 4. Análisis de resultados

En primer lugar, a fin de poder realizar la regresión lineal múltiple, se han extraído los coeficientes de Pearson para poder determinar si las variables independientes estaban relacionadas significativamente con la variable dependiente. El resultado, para cada uno de los cursos ha ofrecido por lo general valores dentro de los límites aceptables, por lo que se ha procedido a realizar el análisis de regresión múltiple hacia atrás. No obstante, se han detectado valores muy elevados, del orden de 0.85 o superiores entre varios parámetros, como las intervenciones en chats y visitas al chat (como puede parecer obvio a priori), por lo que se ha optado por agruparlos para el análisis a fin de evitar problemas de multicolinealidad.

Para el máster online, además, se ha detectado esta misma relación entre el número de páginas visitadas y las visitas al foro, lo que implica la posibilidad de que ambos factores estén explicando el mismo fenómeno y que puede ser indicativo de la adopción de una rutina sistemática en las visitas de los alumnos, que incluya repaso de actividades, visita a noticias, y visitas a foros, así como accesos específicos y limitados para completar las tareas.

En los cursos en que se han obtenido valores

del estadístico de Durbin-Watson, este valor ha estado situado en torno a 2 (2.064), por lo que se ha desechado la posibilidad de autocorrelación entre las variables independientes.

Posteriormente se ha procedido a la realización de un análisis de la varianza (ANOVA), a fin de determinar la existencia o no de una relación lineal entre las variables independientes y la dependiente de forma global, desestimando la hipótesis de la existencia de esta relación para un nivel de significancia de 0.05. Una vez realizado el análisis sobre el modelo propuesto se han encontrado niveles de significancia muy elevados de todos los factores para cada uno de los cursos, por lo que se ha procedido a la eliminación de variables y definición de nuevos modelos en pasos sucesivos hasta obtener valores que permitirían la explicación de los modelos propuestos mediante la combinación lineal de los factores de influencia.

Tras proceder así, no se han encontrado relaciones significativas entre las variables implicadas en el modelo, por lo que se ha procedido a limitar el análisis al número total de interacciones con la plataforma, pero de nuevo no se han descubierto relaciones significativas al nivel aceptable de 0.05, lo que contradice las hipótesis de partida.

Por lo que se refiere a la comparación entre el rendimiento final de los alumnos entre los cursos online y presencial, en la **tabla 1** y teniendo en cuenta que el profesorado y contenidos docentes son idénticos para ambas modalidades y que los criterios de evaluación y mecanismos de entrega son similares, se puede observar una mejora en el rendimiento de los alumnos de la modalidad online, para la que se aprecia además una menor dispersión de la nota final.

#### 5. Conclusiones y consideraciones finales

A lo largo de este estudio se ha realizado un análisis basado en indicadores puramente objetivos y no intrusivos ofrecidos por el sistema LMS con respecto a la actividad en línea de alumnos en cursos de grado y posgrado, buscando la influencia de los mismos en el rendimiento académico final, ba-

sándonos en estudios previos. Sin embargo, los resultados del análisis no permiten establecer ninguna conclusión significativa a este respecto; es decir, no se han podido establecer relaciones significativas entre el número de "hits", el número de intervenciones o lecturas de foros, o de intervenciones en chats con la nota final de los alumnos.

Este hallazgo puede deberse a dos causas: la existencia de una serie de factores objetivos no medibles a través del sistema de registro de la plataforma (como, por ejemplo, el número de horas reales dedicadas al estudio fuera de la plataforma) u otros factores subjetivos; o bien, el reducido tamaño de la muestra.

Pese a que el estudio no ha confirmado la hipótesis de partida, sí que ha permitido distinguir ciertas rutinas de uso de estos sistemas, como la detección de creación de hábitos de acceso a la plataforma, que permitirían diferenciar entre una actuación de acceso a contenidos docentes de amplia duración y otra de resolución de tareas de corta duración.

Por otra parte, se ha estudiado la influencia de los factores de interacción entre alumnos y sistemas de información tanto para entornos de aprendizaje en línea como presencial, no hallándose ninguna diferencia en lo que respecta a la influencia de los factores analizados con respecto al rendimiento académico (nula en ambos casos) pero apreciable en lo referente al rendimiento final en sí, ya que se ha observado un mejor desempeño de los alumnos de la modalidad online, pese a que no se esperaban diferencias significativas. Dada la igualdad a efectos de evaluación de ambas, la explicación a este fenómeno podría darse por varias razones: por una parte, la interacción en enseñanza online tiene un carácter de anonimato superior, por lo que resulta más fácil tomar parte en mecanismos de interacción para los alumnos tanto con otros compañeros como con el profesorado, además de poder contrastar las aportaciones de cada uno, poder consultarlas de nuevo y reforzar conocimientos; por otra parte, la facilidad de distribuir el tiempo más eficientemente (siempre que se disponga de cierta disciplina en el estudio) y el trabajar en

Curso	N	Media	Desv. Estándar
Curso Troncal 1 (online)	8	8,8175	,30410
Curso Troncal 1 (presencial)	13	8,2615	1,00626
Curso Troncal 2 (online)	8	8,6013	,43043
Curso Troncal 2 (presencial)	13	7,8962	,76540

**Tabla 1.** Descriptivos de rendimiento final en cursos de máster.

un entorno con acceso a información complementaria (ya sea guiada o autodidacta) a través de Internet puede permitir un mejor aprovechamiento del estudio.

Como consideración final, indicar la conveniencia de considerar la medida del rendimiento en base a otros parámetros adicionales de índole subjetiva, como la percepción de aprendizaje del propio alumno, o relacionados con las teorías de comportamiento o de adopción tecnológica, como la facilidad de uso del sistema, utilidad percibida, motivación del alumno, calidad percibida de los contenidos, percepción de sentimiento de comunidad, experiencia previa, factores normativos sociales, etc., lo que permitiría realizar análisis más precisos tanto para la influencia de los factores como para la comparación entre diferentes modalidades de enseñanza (presencial vs. online). La importancia de estos factores será tenida en cuenta por los autores en futuros estudios.

Asimismo consideramos de interés la posibilidad de introducir para ambas modalidades tanto mecanismos dinamizadores (revisión entre pares por parte de los alumnos) como correctores (revisión más detallada de comportamientos de acceso y corrección de conductas anómalas) para observar su influencia en el rendimiento de los alumnos.

La principal limitación del estudio la encontramos en el tamaño de la muestra disponible. Aunque el número de muestras seleccionadas para la investigación en conjunto puede considerarse aceptable, es ligeramente inferior a los tamaños recomendados [28]. No obstante, una vez identificada la correlación entre variables que pueden estar explicando el mismo fenómeno, unido a la inexistencia de chat o inactividad de foros en el caso de los cursos de máster presenciales, el número de muestras seleccionado se acerca a los valores exigibles para el cálculo de los coeficientes  $by R^2$ . Sin embargo, consideramos fundamental la necesidad de realizar estudios posteriores que permitan contrastar de forma concluyente los resultados obtenidos con un espacio muestral mayor. El aumento en el número de muestras, por otra parte, permitiría la realización de segmentaciones en base a factores psicosociales o estudios longitudinales para determinar la posibilidad de haber contado para la muestra con grupos excepcionalmente dotados o no para los cursos.

### Referencias

- [1] **Bloomsburg University of Pennsylvania's Department of Instructional Technology.** *E-Learning Concepts and Techniques*. <[http://iit.bloomu.edu/Spring2006\\_eBook\\_files/index.htm](http://iit.bloomu.edu/Spring2006_eBook_files/index.htm)> [Consultado: 18 marzo 2008].
- [2] **J. Kurhila, M. Miettinen, P. Nokelainen, H. Tirri.** The Role of the Learning Platform in Student-Centered E-Learning. *Proceedings of the IEEE international Conference on Advanced Learning Technologies (ICALT'04) - Volume 00* (August 30 - September 01, 2004). ICALT. IEEE Computer Society, Washington, DC, 540-544.
- [3] **J.B. Williams, M. Goldberg.** The evolution of e-learning. *Proceedings of the 22nd ASCILITE Conference*, 2005.
- [4] **A.Y. Wang, M.H. Newlin.** Predictors of web-student performance: the role of self-efficacy and reasons for taking an on-line class. *Computers in Human Behavior 18* (2002), pp. 151-163.
- [5] **B. Dimai, M.Ebner.** Community without a vision won't work. *1st Global Conference, Interactive Convergence: Research in Multimedia*, 2003.
- [6] **Y. Downing, L. Chun-Hsiung, S. Pei-Chen.** The analysis of learning records and learning effect in blended e-Learning. *Journal of Information Science and Engineering 21*, 2005, pp. 973-984.
- [7] **M.K. Barbour, M.A.J. Collins.** Online Writing as an Indicator of Student Performance. *Journal of Instructional Technology and Distance Learning, Vol. 2. No.8.* August 2005.
- [8] **C. Nian-Shing, L. Kan-Min.** Factors affecting e-learning for achievement. *IEEE International Conference on Advanced Learning Technologies (ICALT 2002) Proceedings*.
- [9] **J. Kickul, G. Kickul.** New pathways in e-learning: the role of student proactivity and technology utilization. *Midwest Academy of Management Proceedings 2002*.
- [10] **C. Ramos, E. Yudko.** "Hits" (not "Discussion Posts") predict student success in online courses: A double cross-validation study. *Computers & Education 50* (2008), pp. 1174-1182.
- [11] **P. Moreira, V. Teixeira, H. Valente et al.** E-learning in human nutrition: a pilot study. *EUNIS 2005 Proceedings*.
- [12] **R. Palloff, K. Pratt.** Online learning communities revisited. *21st Annual Conference on Distance Teaching and Learning*, 2005.
- [13] **J.B. Arbaugh, R. Benbunam-Fich.** The importance of participant interaction in online environments. *Decision Support Systems 43* (2007), pp. 853-865.
- [14] **E.W. Black, K. Dawson, J. Priem.** Data for free: Using LMS activity logs to measure community in online courses. *The Internet and Higher Education Volume 11 Issue 2* (2008), doi: 10.1016/j.iheduc.2008.03.002.
- [15] **L.V. Morris, C. Finnegan, S.S. Wu.** Tracking student behavior, persistence, and achievement in online courses. *Internet and Higher Education 8* (2005) pp. 221-231.
- [16] **S. Rafaeli, G.Ravid.** OnLine, Web Based Learning Environment for an Information Systems course: Access logs, Linearity and Performance. *ISECON '97*, Orlando, Florida, pp. 92-99.
- [17] **S.C. Tseng, C.C. Tsai.** On-line peer assessment and the role of the peer feedback: A study of high school computer course. *Computers & Education 49* (2007) pp. 1161-1174.
- [18] **M.K. Tallent-Runnels et al.** Teaching Courses Online: A Review of the Research. Review of Educational Research. *Spring 2006, Vol. 76, No. 1*, pp. 93-135.
- [19] **P.C. Shi, D. Muñoz, F. Sánchez.** The effect of previous experience with information and communication technologies on performance in a Web-based learning program. *Computers in Human Behavior 22* (2006) pp. 962-970.
- [20] **H. Fritsch.** Host contacted, waiting for reply. Final report and documentation of the virtual seminar for professional development in distance education (pp. 355-378). Oldenburg: *Bibliotheks und Informations- systems der Universitat Oldenburg* (Virtual seminar held January-March), 1997.
- [21] **M.F. Beaudoin.** Learning or lurking? Tracking the "invisible" online student. *Internet and Higher Education 5* (2002) pp. 147-155.
- [22] **T.M. Connolly, E. MacArthur, M. Stansfield, E. McLellan.** A quasi-experimental study of three online learning courses in computing. *Computers & Education 49* (2007) pp. 345-359.
- [23] **M.D. Roblyer, J.Freeman, M.B. Donaldson, M. Maddox.** A comparison of outcomes of virtual school courses offered in synchronous and asynchronous formats. *Internet and Higher Education 10* (2007), pp. 261-268.
- [24] **F. Spooner, L. Jordan, B. Algozzine, M. Spooner.** Student Ratings of Instruction in Distance Learning and On-Campus Classes. *The Journal of Educational Research, Vol. 92*, 1999.
- [25] **D.A. Sapp, J. Simon.** Comparing grades in online and face-to-face writing courses: Interpersonal accountability and institutional commitment. *Computers and Composition 22* (2005), pp. 471-489.
- [26] **J. Gal-Ezer, D. Lupo.** Integrating internet tools into traditional CS distance education: students' attitudes. *Computers & Education 38* (2002) 319-329.
- [27] **A.D. Carswell, V. Venkatesh.** Learner outcomes in an asynchronous distance education environment. *Int. J. Human-Computer Studies 56* (2002), pp. 475-494. doi:10.1006/ijhc.1004
- [28] **G.D. Garson.** *Statnotes: Topics in Multivariate Analysis*. <<http://www2.chass.ncsu.edu/garson/pa765/statnote.htm>> [Consultado: 20 de abril 2008]

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 "Acceso y recuperación de información" (José María Gómez Hidalgo, Manuel J. Maña López)

**Tema:** conferencia

El éxito de las Redes Sociales no solo las ha puesto en el candelero de los medios de comunicación, sino también en el de los científicos. Durante años, uno de los trabajos más complejos para un ingeniero de Minería de Datos [1] era la recolección de los datos sobre los que poner a trabajar sus algoritmos. ¿Cómo conseguir que un equipo de personas haga el esfuerzo de etiquetar datos para el Aprendizaje Automático? Las Redes Sociales se caracterizan porque el contenido es generado por sus usuarios, de manera gratuita, y mejor aún, porque describe sus propias relaciones sociales.

Las Redes Sociales son en consecuencia uno de los campos potencialmente más fructíferos para la utilización de las tecnologías de Minería de Datos, con potenciales resultados en forma de Marketing Personalizado, conocimientos sociales, etc. De ahí que las Redes Sociales sean un tema recurrente en las conferencias más prestigiosas sobre Ciencia Computacional e Inteligencia Artificial, y que proliferen los talleres centrados en distintos aspectos de las mismas.

Merece la pena destacar el *First International Workshop on Mining Social Media* [2], un taller internacional a celebrar junto con la Decimotercera Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA), en Sevilla el 9 de noviembre de 2009. Este taller se centra en la aplicación de técnicas de Minería de Datos en relación con las Redes Sociales, y se pueden enviar propuestas de artículos antes del 31 de julio.

Además de tratarse de un taller internacional muy próximo geográficamente, y de contar con un excelente comité de programa, cuenta con la presencia de William Cohen, un investigador estadounidense con una larga carrera en el ámbito del análisis del lenguaje natural, y que actualmente es un referente en la investigación sobre Redes Sociales.

[1] Minería de Datos: [http://es.wikipedia.org/wiki/Data\\_Mining](http://es.wikipedia.org/wiki/Data_Mining).  
[2] First International Workshop on Mining Social Media: <http://www.socialgamingplatform.com/msm09/>.

**Tema:** noticia

La nueva generación de competidores de Google en el negocio de la búsqueda Web tiene uno de sus representantes más interesantes en Bing, el nuevo motor de búsqueda de Microsoft. La aparición de esta herramienta ha desencadenado acontecimientos como que Sergei Brin, uno de los propietarios y creadores de Google, haya puesto en marcha un grupo de sus mejores ingenieros para que desarrollen mejoras en su tecnología de búsqueda, según algunos medios [1].

Lo que desde luego es un hecho es que si alguien desea competir con Google, debe ofrecer algo no solo bueno, sino nuevo. Resultan curiosos en ese sentido algunos comentarios de admiradores del nuevo motor de búsqueda: "Google es tan, tan, ... del siglo pasado."

Y una de las cosas nuevas que ofrece Bing, es la protección contra contenidos indeseados. Además de una opción de búsqueda segura similar a SafeSearch de Google, Bing está planteando la posibilidad de devolver los resultados de determinadas búsquedas en URLs del dominio "explicit.bing.net" [2]. De este modo, aunque un usuario

pretenda acceder a imágenes o vídeos pornográficos y desactive la búsqueda segura, se puede configurar de manera sencilla una regla de filtrado que bloquee las URLs que contengan este dominio. De este modo, resulta más sencillo evitar el desperdicio de recursos que supone la navegación por estos contenidos en el puesto de trabajo, y se contribuye a la protección del menor en Internet.

[1] Fear Grips Google. New York Post, 14/06/2009: [http://www.nypost.com/seven/06142009/business/fear\\_grips\\_google\\_174235.htm](http://www.nypost.com/seven/06142009/business/fear_grips_google_174235.htm).

[2] Bing Community: Safe Search Update: <http://www.bing.com/community/blogs/search/archive/2009/06/12/safe-search-update.aspx>.

### Sección Técnica "Auditoría SITIC" (Marina Touriño Troitiño, Manuel Palao García-Suelto)

**Tema:** Auditoría vs Aseguramiento de TI

En distintas publicaciones, y especialmente en normas y publicaciones de la *Information Systems Audit and Control Association* (ISACA), desde hace pocos años vienen apareciendo como sinónimos las palabras "Auditoría de TI" y "Aseguramiento de TI".

Es más, una gran parte de los nombres de las normas y directrices de auditoría emitidas y publicadas por la ISACA, han sido modificados para incluir la palabra "aseguramiento". De esta forma, como ejemplo, las normas o estándares (de obligado cumplimiento para los profesionales CISAs) son enunciadas como "Normas para la Auditoría y Aseguramiento de TI".

Aún cuando estos términos se usan como equivalentes (etimológicamente no significan lo mismo, ni por lo tanto su alcance es el mismo), en ocasiones, se identifican como actividades complementarias, y en determinados documentos, con un objetivo distinto.

Según el DRAE:

■ Aseguramiento: 1. m. Acción y efecto de asegurar<sup>2</sup>. Asegurar: Afirmar la certeza de lo que se refiere.

■ Auditoría (por supuesto no incluye la definición de auditoría de TI, pero por aproximación): Revisión de la contabilidad de una empresa, de una sociedad, etc., realizada por un auditor. Examinar la gestión económica de una entidad a fin de comprobar si se ajusta a lo establecido por ley o costumbre.

Si se busca la traducción de "assurance" que es de donde viene el término aseguramiento, nos encontramos que, en los distintos diccionarios consultados, la asimila a "confianza" o "garantía" en español.

La propia ISACA en su página al hablar de "assurance", indica que "la confiabilidad (assurance) de la información es la base (los cimientos) sobre la cual se construye la toma de decisiones de una entidad". En el Manual para la preparación del examen CISA de 2009, se menciona esta palabra con diverso espectro y aplicación, pero uno de los más curiosos es la indicación, para su norma 3, de que "*los auditores de SI que tengan funciones de 'aseguramiento' deben garantizar un cuidado razonable en la realización de su trabajo...*". En todo el libro se repite constantemente "los auditores de SI en labores de aseguramiento", pero en ningún momento, ni siquiera en el glosario se indica si estos dos términos son sinónimos o tienen matices distintos, pero apreciables y perceptibles en el campo de trabajo.

Ante tanta indefinición de las asociaciones profesionales (no solo ISACA asimila estas palabras sin definir sin son equivalentes, o

complementarias, o de ámbito diferente de aplicación, sino también otras asociaciones de auditores) surgen las siguientes dudas:

■ ¿Es tal vez una tarea de "aseguramiento", el comprobar la aplicación adecuada de una "norma" o buena práctica (COBIT, ISOs, etc.)?

■ ¿Acaso esta confusión no es simplemente el fruto de una cierta liviandad y falta de rigor en el uso del lenguaje, sino que realmente está planificada para ir induciendo a formar la "visión" de que un auditor de SI sea considerado un "comprobador" de la aplicación de determinada norma o "buena práctica" en TI?

■ ¿Estamos abandonando los criterios de la auditoría de TI en la que el auditor, de forma independiente y objetiva, identifica riesgos para el negocio en el ámbito de las TI?

Como conclusión, es obvio que "nuestro" mundo de TI cambia a mucha velocidad y que las actividades de las organizaciones también. Por ende, estos cambios también suelen reflejarse en los términos de referencia. No obstante, es necesario avanzar cuidadosamente desde los entornos profesionales para no crear confusiones en la labor de las distintas profesiones relacionadas con TI, especialmente en el real significado de los términos utilizados.

Y debería ser una premisa en las asociaciones profesionales: permanecer al margen de grupos de presión, y brindar un servicio "neutral" y profesional. En el caso de la auditoría de TI, es ésta una labor acuñada ya hace mucho tiempo, y por lo tanto los clientes y la sociedad son ya conscientes de su objetivo y alcance, así como de la responsabilidad que tiene un auditor de TI y de lo que se puede esperar de este profesional.

<sup>1</sup> Este tema se ha mencionado en artículos anteriores en esta sección en relación a COBIT, y a las guías para su "auditoría" o "autoevaluación".

<sup>2</sup> El auditor no "asegura", da una opinión objetiva en base a una tarea sistemática de evaluar pruebas y evidencias.

### Sección Técnica "Derecho y Tecnologías" (Elena Davara Fernández de Marcos)

**Tema:** *consulta ciudadana sobre nuevo rango de numeración de móviles*

Hoy en día, el teléfono es uno de los aparatos que más usamos, tanto a nivel profesional como personal ya que nos permite estar en contacto permanente con todos, con independencia del lugar en el que estén o de la hora que sea. Clara muestra de la importancia del móvil para la población española es que, actualmente, en nuestro país, la penetración de la telefonía móvil en España supera ampliamente el 110% de la población. No obstante, según pasa el tiempo, han ido adquiriendo relevancia otras funcionalidades y aplicaciones distintas de la telefonía móvil tradicional, entre los que cabe citar: los accesos móviles de banda ancha a Internet o los nuevos sistemas de comunicación máquina a máquina (M2) sobre redes móviles sin intervención humana.

En este sentido, y precisamente con el fin de habilitar nuevos números para que puedan ser empleados por los servicios de comunicaciones móviles interpersonales y servicios de acceso de banda ancha a Internet, el Ministerio de Industria, Turismo y Comercio ha abierto una consulta pública, que se puede descargar desde su propia web hasta el 3 de julio de 2009, para conocer la opinión de los usuarios sobre el establecimiento de este nuevo rango de numeración para la telefonía móvil.

En todo caso, hay que tener presente que el objetivo de esta medida

no es sino evitar que en un futuro tengamos que hacer frente a la escasez de numeración para nuevos servicios móviles. Asimismo, se plantea la utilización de este rango para prestar servicios de comunicaciones máquina-máquina, con la posibilidad de habilitar un rango diferenciado de números de 13 dígitos. Hay que tener en cuenta que esta medida ha sido adoptada tras el incremento experimentado por las solicitudes de números del rango 6 para la prestación del servicio telefónico móvil, desde la entrada en vigor del plan nacional de numeración telefónica. En todo caso, no deja de ser cierto que este aumento ha tenido lugar de forma paralela y proporcional al número de abonados a este servicio durante los últimos años.

Por último, traer a colación palabras del propio Ministro de Industria que justifica la necesidad de esta iniciativa, anticipándose a los potenciales problemas que puedan aparecer en el futuro, planteando "medidas para la ampliación del segmento de numeración para las comunicaciones móviles, facilitando de este modo el desarrollo de los servicios existentes, la aparición de nuevas aplicaciones y la competencia en el mercado" <<http://www.consumer.es/web/es/tecnologia/2009/06/06/185810.php>>.

**Tema:** *escaso interés de las empresas españolas por los nombres de dominio bajo.es*

Pese a que hoy en día nadie duda del valor del nombre de dominio como activo de la empresa, el III Congreso de Dominios de Internet, que se ha celebrado recientemente en Madrid ha puesto de manifiesto una realidad completamente diferente. Y es que han concluido que las empresas españolas no están a la altura de las europeas en lo que se refiere a la adquisición de nombres de dominio, de hecho, muestran un gran desinterés en adquirir dominios de Internet genéricos y solo les preocupa el de su propia marca.

De hecho, Sedo.com, la empresa que llevó a cabo la mayor subasta de dominios en español hace apenas un mes, reveló el escaso interés de las empresas españolas ajenas al sector de las Nuevas Tecnologías por adquirir dominios de su sector. No obstante, también se destacó la posible influencia de la crisis económica en la que estamos inmersos como factor para esta realidad, en el sentido, de que, a día de hoy, las empresas no pueden ofrecer el dinero que solicitaban los propietarios de los dominios en cuestión.

Asimismo, otra de las conclusiones que alcanzaron los expertos que se dieron cita en el Congreso fue que la nueva liberalización de TLDs (o extensiones formadas por el final del dominio como son .com, .es, .ue...) por parte del organismo internacional ICANN, está haciendo que muchos interesados en una determinada palabra para crear un dominio estén esperando la nueva regulación. Así, se explicó que la nueva tendencia es adoptar TLDs de ciudades (city TLDs).

Por último, decir que en el Congreso se aludió a la posible influencia que iba a tener en los nombres de dominio bajo el .es la decisión de las autoridades mexicanas de cambiar su actual terminación de dominio ".com.mx" por la de ".ex". Y es que, esto supone la posibilidad de volver a registrar palabras en castellano con el ".ex", aunque ya estén ocupadas en otras extensiones como .com o .es. <<http://www.noticiasdot.com/wp2/2009/05/29/las-empresas-espanolas-no-están-interesadas-en-los-dominios-de-internet/>>.

**Tema:** *Facebook, la red social más visitada por los internautas*

Las redes sociales no han dejado de crecer y estar presentes en todos los ámbitos de la vida cotidiana, personal y profesional. No obstante, la red social que mayor éxito ha experimentado entre los usuarios, y de un modo especial entre los internautas estadounidenses, es Facebook. Así, lo muestran datos de Nielsen Media recogidos por Europa Press. Y es que, el tiempo total que los usuarios de Internet

invierten en visitar Facebook se ha incrementado aproximadamente en un 700%, creciendo desde los 1.700 millones de minutos de abril de 2008 a los 13.900 millones del mismo mes de 2009.

No obstante, el citado estudio hace un análisis de la realidad actual de las redes sociales en relación con las preferencias de los internautas y concluye que, tras la popular Facebook, la segunda que goza de mayor aceptación es la web de MySpace, en la que los usuarios invirtieron un total de 4.973 millones de minutos en el último año.

En esta misma línea, también aparece Twitter que, pese al considerable aumento que experimentó el pasado año, aún queda lejos de las cifras de Facebook.

Por último, en el estudio se indica que la frecuencia media de visitas a las redes sociales de Internet aumentó en un 83% respecto al año anterior. No obstante, se hace hincapié en la dificultad de llevar a cabo una comparativa de las distintas redes sociales por cuanto tanto la filosofía como el uso de aplicaciones y herramientas de los portales es totalmente diferente y, por tanto, prácticamente no es posible tomar parámetros de comparación. <<http://www.europapress.es/tecnologia/internet-00446/noticia-tiempo-invertido-facebook-aumenta-700-ciento-ano-20090604134343.html>>.

### Sección Técnica "Entorno Digital Personal" (Diego Gachet Páez, Andrés Marín López)

**Tema:** *Redes de Sensores Inalámbricas*

Las Redes de Sensores, también llamadas Redes Sensoriales, o en inglés *Wireless Sensor Networks* (WSN), están formadas por una serie de pequeños dispositivos electrónicos que interactúan con el mundo exterior por medio de sensores. El nombre que se le da a este tipo de dispositivos es el de "mote", que proviene de la traducción inglesa de la palabra "mota de polvo", con la finalidad de indicar en una sola palabra dos características principales: su pequeño tamaño y la idea de que pueden estar situados en cualquier lugar. Este concepto dio lugar a la creación de las redes de polvo (*Dust Networks*) y al sobrenombre de "polvo inteligente" (*Smart Dust*).

Los motes tienen la capacidad de comunicarse entre sí mediante la creación de redes malladas (redes *mesh*) usando el protocolo ZigBee y retransmitir la información adquirida a través de la red hasta un punto central que registre los valores observados e incluso tome decisiones. Son los propios motes los que se organizan automáticamente cada cierto tiempo para determinar cuáles son las rutas de comunicación disponibles. Esto permite que los motes puedan cambiar de lugar para monitorizar un área distinta en un determinado momento.

La capacidad de auto-organización de la red permite que la monitorización de mayores espacios pueda hacerse fácilmente añadiendo nuevos motes. Al hacerlo, éstos son reconocidos automáticamente por el resto, los cuales actualizarán sus tablas de rutas en consecuencia. Esta característica es muy importante ya que permite que puedan instalarse en gran variedad de escenarios.

La misma característica de auto-organización permite que el área de monitorización también pueda reducirse. En caso de que un mote deje de funcionar, el resto de valores pueden seguir siendo transmitidos por caminos alternativos gracias a un algoritmo de enrutamiento dinámico que se encarga de actualizar las rutas. De esta manera, se garantiza siempre el funcionamiento del resto de la red, al contrario de lo que sucedería con un modelo de comunicación centralizado.

El hecho de que la comunicación sea inalámbrica, eliminando así la

necesidad de trasladar cables de red, y que la alimentación se realice por medio de baterías, hace que la instalación sea fácil y posible en cualquier entorno, abriendo así un gran abanico de aplicaciones que pueden beneficiarse de esta tecnología.

Una de las empresas más innovadoras en el campo de las Redes Inalámbricas de Sensores es española, Libelium, y está desarrollando la plataforma Waspote <<http://www.libelium.com/>>.

### Sección Técnica "Estándares Web" (Encarna Quesada Ruiz, José Carlos del Arco Prieto)

**Tema:** llamadas a la implementación del W3C

Son varias las llamadas a la implementación que actualmente tiene abiertas el W3C. El objetivo principal de estas llamadas es el de incentivar la participación y obtener opiniones e información de la implementación de sus documentos técnicos. Entre las llamadas a la implementación abiertas actualmente podemos mencionar:

#### ■ Implementación de OWL 2

El Grupo de Trabajo de OWL invita a la implementación de su Lenguaje de Ontología Web OWL 2. OWL 2 es una extensión compatible de OWL 1, que proporciona algunas funcionalidades adicionales. Este documento contiene 9 especificaciones técnicas y 4 documentos instructivos <<http://www.w3.org/News/2009#item101>>.

#### ■ Implementación de XQuery Update Facility 1.0

El Grupo de Trabajo de XML Query ha publicado una actualización de la candidata a recomendación XQuery Update Facility 1.0. El documento define una actualización que extiende el lenguaje XML Query, proporcionando expresiones que permiten realizar cambios persistentes a instancias de XQuery 1.0 y XPath 2.0 modelo de datos <<http://www.w3.org/News/2009#item99>>.

#### ■ Implementación de SOAP sobre Java Message Service 1.0

El Grupo de Trabajo de SOAP-JMS Binding hace una llamada a la implementación de la candidata a recomendación SOAP sobre Java Message Service 1.0. El propósito de estos documentos es formar un conjunto de estándares para el transporte de los mensajes SOAP sobre JMS (*Java Message Service*). La idea principal es garantizar la interoperabilidad entre diferentes proveedores de servicios. Asimismo, debería permitir a los clientes implementar sus propios servicios web para parte de su infraestructura, y mantener la interoperabilidad con los proveedores de servicios web <<http://www.w3.org/News/2009#item95>>.

### Sección Técnica "Informática y Filosofía" (José Angel Olivas Varela, Karim Gherab Martín)

**Tema:** *libro*

**Andy Clark.** *Supersizing the Mind: Embodiment, Action, and Cognitive Extension.* Oxford: Oxford University Press, 2008. ISBN-10: 0195333217. En su libro, Andy Clark defiende la idea de que ciertos procesos cognitivos se producen literalmente fuera del cerebro, incluso fuera del cuerpo del individuo que los genera y asimila. En otras palabras, la mente se extiende más allá de las fronteras corporales del organismo biológico. Así pues, al leer a Clark cabe preguntarse: ¿dónde termina la mente y comienza el resto del mundo?

Sabemos que al menos desde Aristóteles el hombre se pregunta qué es esa cosa llamada *mente*. Pocos fueron los progresos hasta que Descartes trazó la primera línea de demarcación: mente es aquella sustancia que se manifiesta a través del pensamiento, en tanto que materia es aquella sustancia que se manifiesta a través del atributo de la extensión. En otras palabras, Descartes afirmaba que el mundo

se divide en un dualismo de reinos independientes: el pensamiento (las mentes) y la extensión (la materia). De estar en lo cierto Descartes, se deduce que ni la mente es extensa ni la materia puede ser pensante. No obstante, añadía el filósofo francés, la mente puede afectar a la materia y la materia a la mente. Es decir, habría una relación causal bidireccional entre mente y materia, entre pensamientos y extensión.

Muchos filósofos desde entonces se han preguntado qué es la mente y qué es el pensamiento. Diversas teorías como el Dualismo de Sustancias, el Dualismo de Propiedades, el Conductismo, la Teoría de la Identidad Mente-Cerebro, el Materialismo Eliminativo, diversos tipos de Reduccionismo, el Emergentismo, el Funcionalismo, etc., han intentado dar respuesta a esta interrogante. Y todas ellas consideraron que el pensamiento, sea lo que sea, es algo que acaece dentro de la frontera corporal que delimita la existencia extensa de un individuo. Para unos, por ejemplo los defensores de la teoría de la identidad psico-neural, la mente se encuentra en el cerebro, y se identifica con los circuitos neuronales. Para otros, por ejemplo los conductistas, la mente no es más que el cúmulo de respuestas (gestos) corporales y lingüísticas que son visibles y audibles en el individuo del que provienen.

El primero en derribar el mito de que los eventos mentales se producen (en su totalidad) en el sujeto que los piensa fue Hilary Putnam, un filósofo de la Universidad de Harvard, quien con un sutil argumento en su artículo "The Meaning of Meaning" mostró que los significados son parcialmente externos al individuo, es decir, se encuentran en parte en el objeto mismo (un objeto que es extenso) en lugar de en el sujeto epistémico.

Andy Clark lleva la idea de Putnam más lejos aún. La tesis de Clark es que los procesos cognitivos pueden llevarse a cabo fuera de los límites craneales, y fuera del cuerpo mismo. Las anotaciones en papel, las calculadoras, las agendas electrónicas o los recursos computacionales modernos son entidades físicas que pueden ser designadas como cognitivas por derecho propio si cumplen una serie de condiciones. Clark habla pues de cognición extendida: el *input* inicial comienza en el interior del individuo, y el cálculo posterior se realiza externamente hasta generar un *output* que vuelve al interior del individuo que toma conciencia del dato resultante.

En cierto modo, Clark da un pequeño rodeo a la hora de responder a la pregunta "¿Qué es la mente?" puesto que sostiene que el *dónde* puede ayudarnos a responder *qué* es la cognición. Clark hace un repaso de multitud de ejemplos de cognición extendida, en los que los instrumentos, las herramientas y los dispositivos materiales son centrales para entender qué es la mente. Podemos pues, según Clark, ir más lejos y hablar no sólo de "cognición extendida" sino de "mente extendida". La tesis de la mente extendida fue originalmente propuesta por Andy Clark y el brillante filósofo de la mente australiano David Chalmers en un artículo publicado en 1998, "The Extended Mind", en la revista *Analysis*.

Este libro de Clark, *Supersizing the Mind*, comienza con un prólogo de Chalmers, y contiene como apéndice el artículo sobre la mente extendida antes mencionado. El libro consta de diez capítulos y consiste en una puesta al día (extendida) de ideas que Clark publicó en anteriores libros: "*Being There: Putting Brain, Body, and World Together Again*" y "*Natural-Born Cyborgs: Minds, Technologies, and the Future of Human Intelligence*".

Pero, ¿cual es la motivación académica de Clark? ¿Qué postura defiende? ¿A quien está respondiendo? Para entender en profundidad el libro de Clark, hay que volver a Putnam. Putnam fue el fundador del funcionalismo, una teoría de la mente que se convirtió en hegemónica en la década de los 60 y lo sigue siendo actualmente. El funcionalismo aboga por un modelo computacional de la mente y es neutral en lo que respecta al sustrato material con que aquella se

manifiesta. Un estado mental es un estado funcional, cuya función viene dada por su relación causal (causas y efectos) con otros estados mentales/funcionales que, en última instancia, empiezan por estímulos sensoriales (*inputs*) y terminan en acciones motoras (*outputs*). Así, según el funcionalismo, los procesos cognitivos tienen que ver con las neuronas tanto como las piezas de ajedrez tienen que ver con la madera. Es decir, que la mente se manifieste en circuitos neuronales es un evento contingente; se podría manifestar igualmente en tuberías hidráulicas que tuvieran la suficiente complejidad y, naturalmente, la misma configuración funcional.

El funcionalismo asumía que los procesos cognitivos humanos se producían (contingentemente) en el cerebro. Era una teoría internalista. Sin embargo, fue el propio Putnam quien puso en aprietos su propia teoría funcionalista con el artículo antes mencionado. ¿Cómo podía el funcionalismo dar cuenta del problema de que los significados que damos a una cosa dependen de esa cosa, o sea, de una cosa que es externa? Putnam renunció al funcionalismo, y aquí es donde aparece la figura de Clark. Clark acepta el argumento externalista de Putnam y, sin renunciar al funcionalismo (o más correctamente, haciendo uso de él), se inventa la cognición extendida. Así pues, el objetivo de Clark es buscar artefactos externos que cumplan las directrices funcionalistas y se integren adecuadamente con nuestra cognición interna, es decir, que sean un continuo de ida y vuelta con nuestros cerebros. Con esta estrategia, funcionalismo y externalismo resultarían ser compatibles.

Con independencia de los gustos de cada lector, ha de quedar claro que quien quiera aprender o saber más sobre cognición extendida debe leer a Andy Clark pues este autor es hoy en día probablemente la referencia mundial y la fuente principal respecto de este tema.

### Sección Técnica "Informática Gráfica" (Roberto Vivó Hernando, Miguel Chover Sellés)

Tema: *libro*

**E. Langetepe, G. Zachmann.** *Geometric Data Structures for Computer Graphics*. A.K.Peters, 2006. ISBN-10: 1568812353. En este número nos ocupamos de un texto ciertamente interesante sobre estructuras de datos generales que han tenido una aplicación directa en el campo de la geometría computacional y la informática gráfica. Hasta donde conocemos no existe en la literatura un texto específico sobre la materia, además del que comentamos, más allá de una descripción al vuelo sobre estructuras necesarias para la comprensión de ciertos algoritmos.

En este texto se da una visión condensada de las estructuras de datos y su algorítmica asociada más relevantes de la materia de informática gráfica proporcionando ejemplos de su aplicación al final de cada capítulo, lo que facilita su comprensión. Hacía falta, en nuestra opinión, disponer de un texto semejante donde poder consultar específicamente cuestiones relativas a la construcción de estructuras como árboles y grafos que son medulares en el tratamiento y síntesis de imágenes.

Nadie se espere un libro aplicable a una titulación de grado. El texto es profundo, dejando al lector libertad de interpretación algorítmica y mucho trabajo por delante si lo que se quiere es incorporar este conocimiento al campo de la docencia. El libro va de menos a más, comenzando por estructuras relativamente simples como los árboles cuadradales hasta grafos de proximidad pasando por partición binaria del espacio, árboles KD, estructuras jerárquicas de contenedores, campos de distancia y diagramas de Voronoi, etc. En los últimos capítulos trata sobre el complejo problema de la actualización de las estructuras de datos frente a cambios geométricos, su robustez y degeneración.

En general, el libro es excelente para profesionales e investigadores en el campo, siendo demasiado ambicioso para fines docentes. En todo caso debería encontrarse en las estanterías de todos aquellos programadores dedicados a resolver problemas relativos a la informática gráfica.

### Sección Técnica "Ingeniería del Software" (Javier Dolado Cosin, Luis Fernández Sanz)

**Tema:** libros

**Thomas Erl.** *SOA Design Patterns*. Prentice-Hall, 2009. ISBN-10: 0136135161. Este libro es un voluminoso tomo de 814 páginas que contiene 24 capítulos y 6 apéndices. Como el propio título indica, este texto describe 85 patrones de diseño para las aplicaciones software de Arquitecturas Orientadas a Servicios (SOA). Es un libro eminentemente práctico que describe los patrones para el diseño e implementación de conjuntos de servicios. También incluye patrones de diseño de seguridad, escalabilidad, composición de servicios, diseño de mensajería, etc. Como ejemplos de títulos de capítulos de este libro podemos mencionar: "Foundational Service Patterns", "Service Implementation Patterns" y muchos otros. Es un libro recomendable.

**Michael Ogrinz.** *Mashup Patterns: Design and Examples for the Modern Enterprise*. Addison-Wesley, 2009. ISBN-10: 032157947X. Es un libro orientado al profesional, donde se describen los patrones de diseño para combinar distintas fuentes de información dentro de la empresa desde la perspectiva del término "web 2.0". Es de fácil lectura y permite situarse rápidamente dentro del problema. Algunos títulos de los capítulos son: "Understanding mashup patterns", "Mashup patterns terminology", "Harvest patterns", "Enhance patterns", "Assemble patterns", "Anti-patterns", etc. En el apéndice se describen distintos casos de estudio, de manera muy genérica.

### Sección Técnica: "Lenguajes de Programación" (Oscar Belmonte Fernández, Inmaculada Coma Tatay)

**Tema:** Web asíncrona

Llevamos ya un tiempo trabajando con Ajax (*Asynchronous JavaScript AndXML*), tecnología que permite mantener una comunicación asíncrona con el servidor, introduciendo una capa de proceso adicional entre éste y la página Web. Esta capa intermedia permite que la comunicación con el servidor se produzca en cualquier momento de forma asíncrona, es decir, se encarga por un lado de interceptar las peticiones del usuario y por otro de gestionar las comunicaciones del servidor en un segundo plano de forma asíncrona. De esta forma el navegador no queda bloqueado esperando todos los datos del servidor ante una acción del usuario. Una de las ventajas frente a las tradicionales comunicaciones síncronas de la web es que se disminuye el tiempo de recarga de una página (ya no es necesario que se vuelva a descargar la página completa cuando existe información previa que no se modifica) y por tanto mejora la interacción, volviéndose más fluida. Además, mejora la experiencia de usuario ofreciendo una respuesta más rápida a las acciones de éste. Sin embargo, tal y como señala *Steve Maryka* en su artículo sobre la revolución que va a suponer la web asíncrona (*What is the Asynchronous Web, and How is it Revolutionary?*), es necesario dar un paso más y conseguir una aplicación que no necesite de interacciones del usuario, permitiendo mostrar cambios dinámicos en el estado del sistema sin la necesidad de acciones de éste sobre la interfaz. Para ello será necesario modificar y avanzar en los modelos de programación que estamos utilizando. <<http://www.theserverside.com/tt/articles/article.tss?l=WhatistheAsynchronousWeb>>.

**Tema:** libro

**P. Ballard, M. Moncur.** *Ajax, JavaScript y PHP*. ANAYA

Multimedia, 2009. ISBN: 9780672329654. Este libro explica de forma detallada cómo programar sitios Web fluidos y dinámicos utilizando técnicas y herramientas de programación tanto del lado del cliente como del servidor, combinando ambas técnicas con Ajax. El libro está dividido en seis partes, más otra dedicada a apéndices. La primera parte está dedicada a explicar los fundamentos de la Web, los conceptos básicos de HTML y CSS y por último a introducir los conceptos Ajax. La segunda parte habla sobre cómo crear *scripts* con el lenguaje Javascript, explicando detalladamente el DOM, la sintaxis y los objetos. La tercera parte se trata de una introducción a la programación con Ajax cuyas tecnologías más complejas se explican en las partes quinta y sexta. Por último, la cuarta parte muestra cómo se crean *scripts* con PHP en el lado del servidor. El libro consta de 393 páginas e incluye un CD con el código fuente de los ejemplos y un conjunto de aplicaciones que nos permitirán empezar a programar Ajax.

### Sección Técnica "Lingüística computacional" (Xavier Gómez Guinovart, Manuel Palomar)

**Tema:** Procesamiento sintáctico de lenguaje natural

**Sandra Kübler, Ryan McDonald, Joakim Nivre.** *Dependency Parsing*. Synthesis Lectures on Human Language Technologies, Morgan and Claypool Publishers, 2009, 115 páginas. ISBN: 978-1-5982-9596-2. Este es un libro dedicado a las técnicas de análisis sintáctico automático (o *parsing*) basado en dependencias, es decir, a los métodos de procesamiento sintáctico del lenguaje natural que se fundamentan en la corriente de la lingüística teórica conocida como gramática de dependencias, cuya difusión en Europa se inicia con la obra póstuma del lingüista francés Lucien Tesnière (1893-1954). Una de las razones del éxito de este modelo lingüístico en el ámbito del procesamiento del lenguaje natural es su adecuación para la descripción lingüística de las lenguas de orden libre de palabras, lo cual explica su implantación, por ejemplo, en las tecnologías de la lengua del euskera. Igualmente, las representaciones formales resultado del análisis sintáctico dependencial, en las que se identifican claramente las relaciones sintáctico-semánticas entre predicados y argumentos, y entre núcleos y modificadores o complementos, constituyen un excelente punto de partida para el procesamiento necesario en aplicaciones de traducción automática o de extracción de la información. En este volumen monográfico de Kübler (Universidad de Indiana), McDonald (Google Research, Nueva York) y Nivre (Universidad de Uppsala), los autores ofrecen una introducción detallada a los fundamentos teóricos y a la aplicación práctica de los diversos algoritmos existentes en el ámbito del procesamiento sintáctico de dependencias, agrupados en tres categorías según se basen en transiciones, en grafos o en gramáticas. Por su carácter recopilatorio y su vocación didáctica, el libro puede ser muy útil como lectura en cursos especializados de procesamiento de lenguaje natural y de lingüística computacional. <<http://www.morganclaypool.com/doi/abs/10.2200/S00169ED1V01Y200901HLT002>>.

### Sección Técnica "Profesión informática" (Rafael Fernández Calvo, Miquel Sarries Griñó)

**Tema:** Consejo de Colegios de Ingeniería Informática

El pasado 9 de junio, el Congreso de los Diputados aprobó por unanimidad las proposiciones de ley de creación del Consejo General de Colegios de Ingenierías Informáticas, que fueron presentadas por el Grupo Socialista y el Grupo Popular. La proposición del PSOE lleva por título "Proposición de Ley de creación del Consejo General de Colegios Oficiales de Ingeniería en Informática" y su texto está disponible en <[http://www.congreso.es/public\\_oficiales/L9/CONG/BOCG/B/B\\_182-01.PDF](http://www.congreso.es/public_oficiales/L9/CONG/BOCG/B/B_182-01.PDF)>.

La proposición del Partido Popular tiene un título mucho más largo y de significado e implicaciones mucho más amplias: "Proposición de Ley de creación del Consejo General de Colegios Profesionales de Ingeniería Técnica en Informática, a fin de asumir la representación de los intereses profesionales en el ámbito nacional e internacional, así como la defensa y desarrollo de la sociedad de la información". Su texto está disponible en <[http://www.congreso.es/public\\_oficiales/L9/CONG/BOCG/B/B\\_166-01.PDF](http://www.congreso.es/public_oficiales/L9/CONG/BOCG/B/B_166-01.PDF)>.

La transcripción de la discusión de ambas proposiciones en el hemiciclo está en las páginas 14 a 20 del Diario de Sesiones del Congreso de Diputados del 9 de junio de 2009, en <[http://www.congreso.es/public\\_oficiales/L9/CONG/DS/PL/PL\\_088.PDF](http://www.congreso.es/public_oficiales/L9/CONG/DS/PL/PL_088.PDF)>. A raíz de la aprobación de dichas proposiciones de ley ATI emitió el 19 de junio un comunicado de prensa que se publica en este número de *Novática* y que está disponible además en <<http://www.ati.es/spip.php?article1262>>. Para conocer con más detalle la posición de ATI respecto a este asunto se recomienda además la lectura del comunicado de prensa de 30 marzo de este año, bajo el título "Ingeniería Informática y Profesión Informática", publicada en el número 197 de *Novática*, disponible también en <<http://www.ati.es/spip.php?article1168>>.

### Sección técnica "Seguridad"

(Javier Areitio Bertolín, Javier López Muñoz)

**Tema:** libros

**B.G. Raggad.** *Information Security Management Concepts and Practice*. CRC. ISBN 1420078542. 2009.

**S.P. Vadhan.** *A Study of Statistical Zero-Knowledge Proofs*. Springer. ISBN 3540713735. 2009.

**M. Whitman.** *Roadmap to Information Security*. Delmar Learning. ISBN 1435480309. 2009.

**P. Stephenson.** *Information Assurance Essentials*. Auerbach. ISBN 1439800308. 2009.

**J. Lopez, S. Furnell, S. Katsikas.** *Securing Information and Communications Systems: Principles, Technologies and Applications*. Artech House Publishers. ISBN 1596932287. 2009.

**J. Areitio.** *Seguridad de la Información: Redes, Informática y Sistemas de Información*. Cengage Learning and Paraninfo. ISBN 8497325028. 2009.

**H. Niederreiter, C. Xing.** *Algebraic Geometry in Coding Theory and Cryptography*. Princeton University Press. ISBN 0691102880. 2009.

**P.B. Garrett.** *Making, Breaking Codes: An Introduction to Cryptology*. Prentice-Hall. ISBN 0131861468. 2009.

**Tema:** Congresos-Conferencias-Simposium

**CANS ' 2009.** (8<sup>th</sup> International Conference on Cryptology and Network Security). Del 12 al 14 de diciembre 2009. Kanazawa. Ishikawa. Japon.

**Eurocrypt ' 2010.** Del 30 de mayo al 3 de junio 2010. Niza. Francia.

**NDSS.** (17<sup>th</sup> Annual Network and Distributed System Security Symposium). Del 28 de febrero al 3 de marzo 2010. San Diego. California. USA.

**ICCCIS ' 2010.** (International Conference on Cryptography, Coding and Information Security). Del 13 al 15 de febrero 2010. Rio de Janeiro. Brasil.

**ACNS ' 2010.** (8<sup>th</sup> International Conference on Applied Cryptography and Network Security). Del 22 al 25 de junio 2010. Beijing. China.

### Sección Técnica: "Tecnología de Objetos"

(Jesús García Molina, Gustavo Rossi)

**Tema:** libro

**Richard C. Gronback.** *Eclipse Modeling Project. A Domain Specific Language (DSL) Toolkit*. Addison-Wesley, 2009. ISBN-10: 0321534077. La monografía de *Novática* de marzo/abril de 2008, número 192, dedicada al "Desarrollo de Software Dirigido por Modelos" (DSDM), incluía un artículo de Richard C. Gronback y Ed Merks, titulado "Arquitectura dirigida por modelos en Eclipse", en el que se analizaba el papel que había jugado el *Eclipse Modeling Project* (ECP) en el área del DSDM. Richard, uno de los líderes del ECP e investigador responsable de Borland Software en productos relacionados con el modelado, acaba de publicar el libro aquí referenciado en el que presenta ECP como un *toolkit* para crear lenguajes específicos del dominio (DSL).

El texto está organizado en cuatro partes. La primera consta de dos capítulos. El primero introduce los conceptos básicos del DSDM (metamodelo, modelo, transformación y DSL), y presenta el ECP, y el segundo presenta ECP como un *toolkit* de creación de DSLs a través de EMF (Eclipse Modeling Framework), GMF (Graphical Modeling Framework) y los proyectos relacionados con transformaciones modelo a texto (M2T) y modelo a modelo (M2M).

La segunda parte está dedicada al desarrollo de DSL y consta de seis capítulos que abordan los distintos elementos que componen un DSL. El capítulo 3 explica el concepto de sintaxis abstracta y cómo crear un metamodelo en EMF; se introduce el modelo de dominio *MindMap* que se usa a lo largo del libro como ejemplo. El capítulo 4 se ocupa de la sintaxis concreta y cómo crear notaciones gráficas con GMF, siendo una de las principales referencias para aprender a usar GMF. El siguiente capítulo es muy corto y se limita a indicar cómo crear una sintaxis concreta textual con xText y TCS, las dos herramientas más conocidas para crear DSLs textuales. El capítulo 6 aborda las transformaciones modelo a modelo, discutiendo aspectos como el *refactoring* o la mezcla de modelos y presenta el lenguaje QVT Operational que es ilustrado con dos ejemplos. Las transformaciones modelo a texto son tratadas en el siguiente capítulo, utilizando xPand para implementar los ejemplos. Finalmente, el capítulo 7 describe cómo empaquetar un DSL y desplazarlo.

La tercera parte es un manual de referencia de todos las herramientas usadas en la parte anterior: Graphical Editing Framework (GEF), GMF Runtime and Tooling, QVT Operational y xPand. La última parte incluye dos apéndices, uno sobre MDA en Eclipse que repasa los estándares MDA implementados en Eclipse y otro que reproduce una tabla del manual de GMF con combinaciones de teclas útiles.

Sin duda se trata de la principal referencia para aprender a definir DSLs con GMF, y resuelve el problema de la escasa documentación sobre cómo usar GMF que existía hasta ahora, al tiempo que también es un excelente libro para comprender los fundamentos de la definición de DSLs, en particular DSL gráficos. La elección de QVT Operational y xPand ha sido acertada. Su publicación es una buena noticia para todos aquellos que han empezado o tienen la intención de acercarse al mundo de los DSLs utilizando GMF.

### Sección Técnica: "Tendencias tecnológicas"

(Alonso Alvarez García, Gabriel Martí Fuentes)

**Tema:** alertas

■ El cloud computing se consolida como modelo empresarial para las TIC <[http://www.tendencias21.net/El-cloud-computing-se-consolida-como-modelo-empresarial-para-las-TICs\\_a3078.html?preaction=nl&id=4954309&idnl=47388&](http://www.tendencias21.net/El-cloud-computing-se-consolida-como-modelo-empresarial-para-las-TICs_a3078.html?preaction=nl&id=4954309&idnl=47388&)>.

■ Esta es algo antigua pero interesante, sobre la importancia del microblogging: Las plantas podrán pedir agua a través de un

dispositivo y Twitter <<http://www.elmundo.es/elmundo/2009/03/27/navegante/1238142791.html>>.

■ La empresa catalana Symmetric ha creado el programa gratuito Approbo, que detecta si un trabajo o artículo es original o coincide con otros de la red <<http://approbo.citilab.eu/>>.

■ Cloud is an operations model, not technology <[http://news.cnet.com/8301-19413\\_3-10249486-240.html?tag=mncol](http://news.cnet.com/8301-19413_3-10249486-240.html?tag=mncol)>.

■ SISMIC. El Laboratorio de Estudios Geofísicos Eduard Fontserè (LEGEF), proporciona información sismográfica (casi) en tiempo real de los diferentes observatorios <<http://sismic.am.ub.es/earthquakes/>>.

■ Después de la notoriedad adquirida por siniestros recientes como el del Airbus de Air France y las dificultades para su localización, webs como estas (basadas en la colaboración de los mismos usuarios, Web 2.0) demuestran su importancia y casi necesidad en momentos en que es importante localizar a tiempo a las víctimas de un siniestro (aunque en este último caso poco se podía hacer). <<http://www.localizato.com/>>.

■ The World Economic Forum (WEF) has announced 34 visionary companies selected as Technology Pioneers 2009 for their ability as innovators and whose technologies will have a deep influence on business and society. <<http://www.weforum.org/en/Communities/Technology%20Pioneers/index.htm>>.

■ Los vídeos de las presentaciones del Telecom I+D están disponibles públicamente. <[http://isabel.dit.upm.es/component/option,com\\_docman/task,cat\\_view/gid,123/](http://isabel.dit.upm.es/component/option,com_docman/task,cat_view/gid,123/)>.

■ "Biggest Battle Yet For Social Networks: You, Your Identity And Your Data On The Open Web" <<http://www.techcrunch.com/2008/11/30/facebook-google-myspace-data/>>.

■ "El Gobierno quiere promover el software libre en la Administración" <<http://www.expansion.com/2009/01/07/funcion-publica/1231318890.html>>.

■ "YouTube llega a la televisión... a través de la consola" <<http://baquia.com/actualidad/noticias/14508/youtube-llega-a-la-television...-a-traves-de-la-consola>>.

■ Cuando parecía que el PLC dejaba de ser una opción, ahora son las bombillas las que amenazan a la radio: <<http://www.cellular-news.com/story/33993.php>>.

■ Interesantes productos y prototipos de interfaces hápticos <<http://www.cscout.com/blog/2008/10/24/digital-vision-for-the-blind.html>>.

■ "Quantum encryption and the weakness of European science". I+D europea: buena transmisión universidad-empresa, pobres resultados académicos <<http://arstechnica.com/business/news/2008/10/quantum-encryption-and-the-weakness-of-european-science.ars>>.

■ "Opening the Cloud". Herramientas Open Source para Cloud Computing <[http://www.technologyreview.com/web/21642/?bcsi\\_scan\\_87A666907766D0F0=5cYK7uC0mnOArziJxD57azMAAADsOY8B](http://www.technologyreview.com/web/21642/?bcsi_scan_87A666907766D0F0=5cYK7uC0mnOArziJxD57azMAAADsOY8B)>.

■ BBC se ha hecho eco de un estudio de mercado en el que afirman que la complejidad de los móviles actuales genera frustración en muchas personas. <<http://news.bbc.co.uk/2/hi/technology/7833944.stm>>.

## Sección Técnica: "TIC y Turismo"

(Andrés Aguayo Maldonado, Antonio Guevara Plaza)

**Tema:** Centro Tecnológico Robotiker: proyecto eTourGUNE

ROBOTIKER-Tecnalia es un Centro Tecnológico especializado en Tecnologías de la Información y Comunicaciones (TIC), y en Desa-

rollo de Producto, que tiene como objetivo contribuir activamente al desarrollo sostenible de la Sociedad a través de la Investigación y la Transferencia Tecnológica <<http://www.robotiker.es>>.

Se constituye como una Fundación privada, sin ánimo de lucro, en la vanguardia de la Investigación, el Desarrollo Tecnológico y la Innovación (I+D+i), manteniendo un compromiso máximo con la calidad, la eficiencia en el servicio a los clientes y el constante desarrollo y formación de sus profesionales.

El centro tecnológico está formado por 24 empresas y 4 instituciones. Las instituciones son las siguientes:

■ DIPUTACIÓN FORAL DE BIZKAIA, Departamento de Innovación y Promoción Económica

■ GOBIERNO VASCO, Departamento de Educación, Universidades e Investigación

■ GOBIERNO VASCO, Departamento de Industria, Comercio y Turismo

■ UNIVERSIDAD DEL PAÍS VASCO, Escuela Técnica Superior de Ingeniería de Bilbao

Entre los diferentes proyectos en marcha, se encuentra eTourGUNE, enmarcado dentro del Programa de Investigación Estratégica ETORTEK del Departamento de Industria, Comercio y Turismo del Gobierno Vasco, que tiene como principal objetivo profundizar y avanzar en el desarrollo de la investigación estratégica en Ciencias del Turismo a través de la convergencia de áreas científicas y tecnológicas tradicionales con el fin de favorecer la competitividad y la innovación en el sector turístico de Euskadi y, por consiguiente, en su economía y sociedad. El Turismo es un fenómeno complejo que agrupa una actividad empresarial e industrial complementada por una producción y oferta de servicios otorgados por organizaciones de carácter público-privadas. Las compañías aéreas, hosteleras, restauradoras, las empresas dedicadas al entretenimiento y al ocio, las agencias y las organizaciones de destino son interdependientes a la hora de ofrecer un servicio-producto. Esta interdependencia de los sub-sectores a la hora de producir y alcanzar unos objetivos comunes hace que la actividad turística deba ser concebida como un sistema funcional, entendiendo como tal, un conjunto de elementos relacionados entre sí para el logro de unos objetivos comunes.

Los principales objetivos específicos de este proyecto son los siguientes:

■ Apoyar la I+D+i científico-tecnológica en el área de eTurismo, desarrollando proyectos avanzados mediante la aplicación de las Tecnologías de la Información y Comunicaciones, incluyendo aspectos relacionados con la adquisición, almacenamiento y gestión de los contenidos turísticos de una forma sencilla y sostenible para todos los actores en la cadena de valor.

■ Potenciar el conocimiento y el desarrollo tecnológicos para la prestación de servicios turísticos avanzados e integrados que mejoren la participación, movilidad e interoperabilidad.

■ Promover el conocimiento para facilitar las innovaciones bajo el concepto de nuevos productos y servicios turísticos de valor añadido que posicionen a la industria turística de la Comunidad Autónoma de Euskadi en una posición preferente.

■ Desarrollar un conjunto de demostradores, en campos de aplicación seleccionados, que permitan evaluar y prever la situación presente y evolución futura del sector turístico en general y de sus empresas en particular.

■ Desarrollar un conjunto de instrumentos metodológicos y tecnológicos de soporte para la evaluación de la competitividad y la innovación tecnológica del sector turístico en general y de sus empresas en particular, haciendo especial hincapié en las empresas de base tecnológica.

Víctor Amadeo Bañuls  
Silvera, José Luis Salmerón  
Silvera

Universidad Pablo de Olavide, Sevilla

<{vabansil,salmeron}@upo.es>

# Un metaanálisis de la actividad prospectiva internacional en el campo de los Sistemas y Tecnologías de la Información

## 1. Introducción

El análisis de los avances en el campo de los Sistemas y Tecnologías de la Información (SI/TI) es una prioridad científica de primer orden. Por una parte, por la relevancia del sector de los SI/TI en el desarrollo de las naciones. Por otra parte, debido al dinamismo de este sector. Es por ello que los SI/TI han sido frecuentemente tratados en los procesos de Prospectiva Tecnológica a nivel nacional, como parte del análisis de las tendencias de las Tecnologías de la Información (TI) en la economía y la sociedad.

La primera experiencia de Prospectiva Tecnológica que incluyó las TI como área clave fue el programa japonés. Actualmente, casi todos los programas nacionales incluyen este tópico como área clave. Este es el caso de Alemania, Francia, Corea del Sur, Reino Unido, Suecia, Holanda, Australia y EE.UU., por citar algunos de los estudios más relevantes entre otros a nivel nacional (se pueden ver más ejemplos en [1][2][3][4]).

En España, al igual que en el resto del mundo, ha sido relevante la prospección de los desarrollos en el campo de los SI/TI en ejercicios de Prospectiva Tecnológica. Ejemplo de dicha actividad son los informes que el Instituto Catalán de Tecnología (ICT), integrante del Observatorio de Prospectiva Tecnológica Industrial (OPTI), ha presentado acerca de la Sociedad de la Información en España.

En este trabajo se realiza un metaanálisis de los resultados en materia prospectiva en SI/TI a nivel internacional. El enfoque aquí presentado supone una novedad en el ámbito académico, tanto en España como a nivel internacional. A continuación se detallan las secciones en las que se va a estructurar este trabajo. En la sección segunda se conceptualiza la Prospectiva Tecnológica y su difusión a nivel internacional. En la sección tercera se aborda el rol de las Tecnologías de la Información en la actividad prospectiva internacional. A partir de la sección cuarta se realiza un análisis de las áreas y tecnologías clave en el sector de los SI/TI en el horizonte temporal. Este artículo finaliza con las conclusiones.

## 2. La Prospectiva Tecnológica

La Prospectiva Tecnológica es una disciplina de conocimiento orientada a visualizar el largo plazo de la ciencia y la tecnología, la economía y la sociedad con el ánimo de

**Resumen:** *los sistemas y tecnologías de la información han sido frecuentemente tratados en los procesos de Prospectiva Tecnológica como parte del análisis de las tendencias de las Tecnologías de la Información en la economía y la sociedad. En este trabajo se realiza un metaanálisis, mediante la técnica denominada análisis de contenido, de los resultados en materia de Sistemas y Tecnologías de la Información de la actividad prospectiva internacional con el objetivo de obtener un listado de tecnologías críticas en este campo. El enfoque presentado en este trabajo supone una novedad en el ámbito académico, tanto en España como a nivel internacional.*

**Palabras clave:** *análisis de contenido, metaanálisis, Prospectiva Tecnológica, Sistemas de Información, Tecnologías de la Información.*

identificar las áreas estratégicas de investigación así como la aparición de tecnologías genéricas y emergentes que den paso a los mayores beneficios económicos y sociales [5].

Los orígenes de la Prospectiva Tecnológica se sitúan en la década de los 60 en Estados Unidos, bajo la denominación de actividades de predicción y evaluación tecnológica. En los años 70, en el seno del Instituto Japonés de Ciencia y Tecnología (NISTEP), se inició una de las actividades de Prospectiva Tecnológica más representativas a nivel internacional, la cual continúa en nuestros días con una periodicidad de cinco años.

Dicha actividad consiste en la realización de un estudio prospectivo acerca del futuro de la ciencia y tecnología de Japón. No obstante, no es hasta la década de los 90s cuando se produce un incremento notable de las actividades de Prospectiva Tecnológica, tanto a nivel nacional como supranacional. Desde entonces, casi la totalidad de los países y regiones tienen programas oficiales de Prospectiva Tecnológica, orientados a dar soporte a su estrategia de I+D. Estos programas reúnen a expertos destacados en diversos campos de la ciencia y la tecnología con el objetivo de encontrar tendencias tecnológicas y tecnologías críticas para la competitividad de la región objeto de estudio.

Para cumplir este objetivo, existen diversas metodologías prospectivas, las cuales tienen como objeto la sistematización de la reflexión colectiva, que es la base del proceso, y la consecución de consensos entre expertos. Dichas herramientas suelen estar inspiradas en herramientas de Predicción Tecnológica. En particular, los paneles de expertos, el método Delphi y los métodos de generación de escenarios se han empleado para sistematizar

expectativas casi siempre de forma combinada.

En España, al igual que en el resto del mundo, ha proliferado la actividad prospectiva en los últimos años. Podemos citar a modo de ejemplo de dicha actividad los informes que realiza periódicamente desde 1998 el OPTI<sup>1</sup> (Observatorio de Prospectiva Tecnológica Industrial) en 14 sectores industriales, creado por el entonces Ministerio de Industria y Energía, actualmente Ministerio de Industria, Turismo y Comercio o los emitidos por la Agencia Nacional de Evaluación y Prospectiva (ANEP). Otros trabajos, no ligados directamente con las autoridades públicas, que han abordado tendencias tecnológicas y su impacto sobre la sociedad española con un enfoque prospectivo, son los del Grupo de Estudios sobre Tendencias Sociales así como los informes anuales de la Fundación Cotec y de la Fundación Telefónica. Otras entidades con actividad periódica en materia prospectiva en España son Prospektiker y la Federación Española de Empresas de Tecnología Sanitaria.

## 3. Las Tecnologías de la Información en la actividad prospectiva

Tradicionalmente, los estudios de Prospectiva Tecnológica han abordado las tendencias de las TI como medio de representación de una visión futura de la sociedad en general. El exponente más destacado de este enfoque es la actividad prospectiva del NISTEP sobre el futuro tecnológico de Japón [6].

Otro enfoque es el estudio de las tecnologías críticas para la competitividad nacional. Ejemplo de ello son Francia y EE.UU. Los criterios para la consideración de tecnología crítica en cada país varían, siendo por ejemplo en caso de EE.UU. la prosperidad económica y seguridad nacional, y en el caso francés la posición

País	Programa
Sudáfrica	National Research and Technology Foresight Project
EE.UU.	New Forces at Work: Industries Views Critical Technologies
India	Technology Visio 2020
Japón	Technology Foresight Survey
Corea del Sur	Technology Forecast
República Checa	Analyses of international key technologies lists
España	OPTI, Informes de Prospectiva Tecnológica Industrial
Finlandia	On the way to technology vision
Francia	French 100 Key Technologies
Irlanda	Technology Foresight Ireland
Israel	Science and Technology Foresight for Israel
Suecia	Teknisk Framsyn
Holanda	Technology Radar
Reino Unido	The Foresight Program
Australia	Matching science and technology to future needs 2010

**Tabla 1.** Muestra de programas de Prospectiva Tecnológica.

relativa del país, atractivo de la industria y factores críticos de éxito.

En algunos casos, los estudios de acerca de tendencias y/o tecnologías críticas se complementan con informes acerca del sector de las TI en el país de estudio. Estos estudios pueden centrarse en el estado actual del sector, por ejemplo las contribuciones del ITEC Group Technologies al programa de prospectiva del Reino Unido o los informes acerca de la industria de contenidos digitales, las tendencias de evolución en las TI o el futuro de los medios de comunicación social que se han realizado en el seno de la OPTI; o en el estado futuro en forma de escenarios, como en el caso australiano o tailandés.

Respecto a la actividad prospectiva en SI/TI, no está extendida más que como un subconjunto de la actividad prospectiva en TI. Es decir, las tendencias emergentes en SI/TI son tratadas como un apartado de las tendencias generales de las TI. En la sección siguiente se analiza en detalle esta actividad prospectiva en SI/TI, en base a una muestra de programas internacionales.

#### 4. Muestra de la actividad prospectiva internacional

En la **tabla 1** se relacionan los programas internacionales que componen la muestra en base a la cual se realiza el metaanálisis de la actividad prospectiva internacional en el campo de los SI/TI.

La muestra se ha obtenido a partir de la revisión de la actividad de prospectiva tecno-

lógica internacional contemplada en bases de datos oficiales (Futur, IPTS, APEC) y publicaciones académicas. Los criterios para la inclusión en la muestra son:

- Contemplar las TI<sup>2</sup> como área clave.
- Resultados disponibles en español, inglés o francés.
- Resultados con vigencia actual.

En el caso de existir varios programas candidatos en un mismo país, se ha seleccionado el último y/o más representativo. Más información sobre los estudios contenidos en la muestra puede ser encontrada en [7].

#### 5. Metodología de análisis

El resultado de la muestra de procesos prospectivos ha sido procesado con la técnica

denominada análisis de contenido. Según Krippendorff ([8] p.28) el análisis de contenido es "una técnica de investigación destinada a formular, a partir de ciertos datos, inferencias reproducibles y válidas que puedan aplicarse a su contexto". Esta metodología se ha seleccionado, debido a su adecuación para procesar datos científicos. A continuación se detallan algunos aspectos concernientes a esta metodología como categorización, unidades de muestreo y registro y la codificación.

La categorización de los resultados se realiza en función a las áreas clave propuestas por Straub y Wetherbe [9]: interfaz humano-máquina, comunicaciones y soporte a sistemas. Las motivaciones para seleccionar esta categorización son: (1) es uno de los trabajos más referido en los antecedentes de la Prospectiva en SI/TI (2) dicha clasificación fue validada en investigaciones posteriores [10].

Una vez definidas las categorías, se determinan las unidades de muestreo y registro. Las unidades de muestreo, definidas como las porciones de la realidad observada que se consideraran independientes unas de otras [8], son los resultados en materia de SI/TI de los procesos incluidos en la muestra. Las unidades de registro, definidas como fracciones de las unidades anteriores que determinan la codificación de las diferentes categorías [8], son cada una de las tecnologías señaladas como clave en cada uno de los procesos de prospectiva recogidos en la muestra. En total se detectaron 297 unidades de registro entre las 15 unidades de muestreo<sup>3</sup>. La labor de codificación, esto es la asignación de las unidades de registro a las distintas categorías, ha sido realizada por los autores. Estos codificadores cumplen los requisitos esperados de nivel cultural, formación y conocimiento del marco teórico.

El coeficiente de fiabilidad (CF) -expresión

País	Interfaz Humano-Máquina	Comunicaciones	Soporte a Sistemas	Unidades de Registro
Sudáfrica	29%	14%	57%	7
EE.UU.	10%	12%	78%	73
India	0%	69%	31%	16
Japón	8%	77%	15%	13
Corea del Sur	0%	52%	48%	29
Rep. Checa	0%	17%	83%	6
España	0%	50%	50%	8
Finlandia	14%	29%	57%	7
Francia	3%	17%	80%	30
Irlanda	22%	39%	39%	23
Israel	29%	43%	29%	7
Suecia	14%	57%	29%	7
Holanda	33%	0%	67%	3
Reino Unido	20%	26%	54%	65
Australia	0%	67%	33%	3

**Tabla 2.** Mapa global de actividad prospectiva.

Categoría	Tecnología	Unidades de Registro
Interfaz Humano-Máquina	Interfaces de usuarios.	4
	Trabajo multimedia.	4
	Realidad virtual.	5
	Reconocimiento de imagen y voz.	7
	Dispositivos avanzados de interacción.	7
	Otros.	7
Comunicaciones	Ancho de banda.	17
	Tecnología móvil.	21
	Tecnología inalámbrica.	8
	Datos y encriptación.	19
	Tecnología óptica.	9
	Otros.	19
Soporte a Sistemas	Computación ubicua y portabilidad.	15
	Componentes y dispositivos.	46
	Software.	29
	Tecnologías de proceso.	25
	Datos y almacenamiento.	29
	Otros.	26

Tabla 3. Tecnologías de la información clave.

(1)- es cercano al 100% (94,27%) lo cual indica un elevado nivel de fiabilidad en la codificación.

$$CF = 2m / (N_1 + N_2) = 94,27\%; m=280; N_1 = N_2 = 297. \quad (1)$$

donde:

- "m" (m=280) es el número de decisiones de codificación en las que coinciden el codificador 1 y el codificador 2.
- "N<sub>1</sub>" (N<sub>1</sub>=297) es el número de decisiones de codificación realizadas por el codificador 1.
- "N<sub>2</sub>" (N<sub>2</sub>=297) es el número de decisiones de codificación realizadas por el codificador 2.

Asimismo, cabe destacar que cuando se clasificaron el 5% unidades de registro, el 100% de las categorías ya habían aparecido, pudiéndose clasificar al finalizar el proceso el total de las unidades de registro en las categorías propuestas.

### 6. Resultados y conclusiones

En la tabla 2 se muestran los resultados por unidad de muestreo, es decir, por programa de Prospectiva Tecnológica incluido en la muestra. A través de este análisis, se puede apreciar como cada país tiene un enfoque hacia una categoría de tecnologías en concreto. En primer lugar están los países focalizados en *tecnologías de soporte sistemas* (Holanda, Francia y EE.UU.). Otro sector de países está más orientado a las *comunicaciones* (India y Japón).

El resto de países mantiene un equilibrio entre estas dos categorías y las tecnologías de *interfaz humano-máquina*. Dichas diferencias pueden estar motivadas principalmente por las necesidades tecnológicas, económicas, sociales o industriales de cada país.

En un segundo nivel, se analiza en cada categoría cuales son las tecnologías concretas más relevantes a nivel global (tabla 3). Dicho análisis se ha realizado una vez categorizadas las unidades de registro, mediante la observación de ocurrencias en cada categoría. Como resultado del mismo se muestran las cinco tecnologías en cada área con mayor nivel de ocurrencias. Los resultados de la tabla 3 toman como referencia el conjunto de tecnologías en total. Debido a las diferencias de unidades de registro existentes entre países (de los 73 de EE.UU. a los 3 de Australia) el peso del global de tecnologías está influenciado en gran medida por las prioridades tecnológicas de los países con mayor número de unidades de registro, en este caso EE.UU. y Reino Unido.

Del análisis de resultados se desprende que las tecnologías más relevantes en los próximos años van a estar relacionadas con las tecnologías móviles, el desarrollo de componentes así como de dispositivos avanzados de interacción. Además, cabe destacar del análisis de los informes comprendidos en la muestra que, en opinión de los expertos implicados en estos procesos de Prospectiva Tecnológica, estos desarrollos llegarán a su madurez en el horizonte temporal 2015 [11]. Una vez alcanzada esta madurez, las prioridades científicas en el ámbito de los SI/TI se centrarán en su confluencia con áreas cercanas, como las ciencias vivas, desarrollo de materiales, nanotecnología y biotecnología.

### Agradecimientos

Esta investigación ha sido financiada por el Ministerio de Educación y Ciencia como parte del proyecto SEJ2006-1474 cuyo investigador principal es el Dr. José Luis Salmerón Silvera.

### Referencias

[1] V.A. Bañuls, J.L. Salmeron. Benchmarking the Information Society in the Long Range. *Futures*, 39(1), pp. 83-95 (2007).

[2] V.A. Bañuls, J.L. Salmeron. A Scenario-based Assessment Model. *SBAM, Technological Forecasting and Social Change*, 74(6), pp. 750-762 (2007).

[3] V.A. Bañuls, J.L. Salmeron. Foresighting key areas in the information technology industry. *Technovation*, 28(3), pp. 103-111 (2008).

[4] V.A. Bañuls, J.L. Salmeron. *Áreas Clave para Desarrollo Económico y Social: Una Visión desde la Actividad Prospectiva Internacional, Problemas de desarrollo*. Pendiente de publicación (2008).

[5] B. Martín. Foresight in Science and Technology. *Technology Analysis and Strategic Management*, 7(2), pp. 139-168 (1995).

[6] T. Kuwahara. Technology Forecasting Activities in Japan. *Technological Forecasting and Social Change*, 60, pp. 5-14. (1999).

[7] V.A. Bañuls, J.L. Salmeron. *Fundamentos de la Prospectiva de Sistemas de Información*. Madrid: RAMA, pendiente de publicación (2009).

[8] K. Krippendorf. *Metodología de Análisis de Contenido: Teoría y Práctica*. Barcelona: Ediciones Paidós, 1999.

[9] D.W. Straub, J.C. Wetherbe. Information Technologies for the 1990s: an Organizational Impact Perspective. *Communications of the ACM*, 32(11), pp. 1328-1339 (1989).

[10] V. Grover, M. Goslar. Information Technologies for the 1990s: the Executives' View. *Communications of the ACM*, 36(3), pp. 17-19.

[11] NISTEP. The Seventh Technology Forecast Survey - Future Technology in Japan. *NISTEP (Informe 71, 2001)*. Tokio, Japón.

### Notas

<sup>1</sup> El listado de sus últimas publicaciones en materia de prospectiva se encuentra en <<http://www.opti.org/prospectiva.asp>>.

<sup>2</sup> Como se ha referido anteriormente, en la actividad de Prospectiva Tecnológica se recogen los SI/TI como una categoría de análisis dentro de las tendencias de TI.

<sup>3</sup> Tan solo se consideraron como unidades de registro los SI/TI, es decir, las TI comprendidas dentro del ámbito definido para la Prospectiva en SI/TI. Las TI excluidas hacían referencias a desarrollos de las TI fuera del ámbito de los Sistemas de Información. Concretamente fueron excluidas por este motivo 66 unidades de registro del análisis. Además se excluyeron aplicaciones concretas de los SI/TI.

## Una paradoja divina

Antonio Vaquero Sánchez  
Catedrático Emérito de la Universidad  
Complutense de Madrid; socio de ATI

<arvsmcmg@telefonica.net>

Ha salido a la calle en autobús. Dios tiene publicidad pagada. ¿Necesita Dios publicidad? El creyente responderá que no. Y el no creyente se gasta el dinero en hacer publicidad de Dios. Y el creyente acude al trapo y responde con más publicidad. Parece un contrasentido. En realidad es cómico; pero también descorazonador, porque hay que ver en lo que gastamos el tiempo y el dinero. Como si no hubiera cosas más interesantes que hacer. ¿Cómo es posible que haya dinero para esas chorradas cuando está la casa sin barrer? Cuando hay que andar por el suelo, miramos al cielo. Así nos va.

Parece que el espectáculo público ha remitido. ¿En qué nueva forma volverá? Pero no hay mal que por bien no venga. Corre tinta abundantemente, creacionista y evolucionista, porque coincide el bicentenario del nacimiento de Darwin. Ciencia y religión. Evolucionismo, creacionismo, diseño inteligente... y por ahí. Pues aquí va un poco de tinta científica.

La existencia de Dios no es una cuestión del dominio de la Ciencia, por supuesto. Entonces, ¿qué tiene que decir un científico sobre Dios?

En primer lugar la Ciencia estudia el lenguaje; es decir, quiere responder a la pregunta ¿qué quiere decir lo que se dice?, o sea las relaciones entre lenguaje y pensamiento. En particular la Informática trata con lenguajes. Formales, sí; pero la Inteligencia Artificial trata también de responder involucrando el lenguaje natural. El informático está acostumbrado "a pensar con lógica"; en particular a descubrir el sentido de lo que se dice. ¿Tiene sentido lo que se dice? Conviene analizar las frases chocantes para sacar su sentido, si es que lo tienen, o para probar que no lo tienen.

Hay muchas afirmaciones, aparentemente con sentido, pero que, después de ser formuladas, se ha visto que no lo tienen. Por ejemplo en Física cuántica, Werner Heisenberg desmanteló una buena cantidad de teorías que violaban su "principio de incertidumbre" (1927); es decir, los que las expresaban no entendían lo que decían. Él sí fue consciente de que no las entendía y esa desazón fue un acicate para llegar a alcanzar su ley. Moraleja: Lo que hoy no se entiende, puede que algún día se llegue a explicar.

Con respecto a Dios hay cosas que se dicen y chocan. Por ejemplo, en el Credo católico se

dice: "*Jesucristo... hijo único de Dios, engendrado, no creado, de la misma naturaleza que el Padre...*". La idea monoteísta de un Dios creador, incluida la Trinidad cristiana, pasa por la clave de que **Dios no se ha creado a sí mismo**.

Independientemente de que esa creencia es parte de la fe religiosa, se debe analizar la frase desde la perspectiva del lenguaje y la lógica. La verdad es que suena como un sinsentido. Rondan por la cabeza las paradojas sabidas... Algo tendrán que ver en esto, piensa uno.

Recordemos que la famosa paradoja de Russell (1901)<sup>1</sup> es un esquema: "El conjunto C de todos los conjuntos que no se contienen a sí mismos, ¿se contiene a sí mismo?" La respuesta es: Ni Sí ni No.

En efecto:

Si se contiene a sí mismo, entonces C tendría que estar en el conjunto de **todos** los conjuntos que no se contienen a sí mismos. Pero SI ESTÁ en ese conjunto, entonces no se contiene a sí mismo, puesto que todos los conjuntos que están no se contienen a sí mismos. Luego es falso, puesto que la hipótesis es que se contiene a sí mismo.

Y si NO, entonces C tendría que estar en el conjunto de **todos** los conjuntos que no se contienen a sí mismos; pero si está, se contiene a sí mismo. Luego también es falso, puesto que la hipótesis es que se no contiene a sí mismo.

Así pues, es rigurosamente falsa tanto la proposición afirmativa como la negativa. Las consecuencias de esta paradoja en la Lógica y en la Matemática han sido trascendentes [1]. También en Informática; así el concepto de "tipo de datos" proviene directamente de eludir esa paradoja.

Hay muchos ejemplos de este esquema, como la famosa Paradoja del Barbero: El barbero que afeita a todos los hombres que no se afeitan a sí mismos. Si se sustituye "barbero" por "Dios", "afeitar" por "crear" y "hombres" por "cosas", se sacan las mismas consecuencias lógicas.

En efecto una particularización de la paradoja de Russell aplicable al caso es la Paradoja de Dios: Dios ha creado todas las cosas que

no se han creado a sí mismas. Es claro que esa afirmación concuerda con la fe del creyente. Pero de ella se sigue que las afirmaciones "Dios se ha creado a sí mismo" y "Dios no se ha creado a sí mismo" son rigurosamente falsas. Las dos.

Si Dios no se ha creado a sí mismo es algo que no se ha creado a sí mismo y, por tanto, como Dios ha creado todas las cosas que no se han creado a sí mismas, entonces se ha creado a sí mismo, en contra de la hipótesis. Luego la afirmación "Dios no se ha creado a sí mismo" es rigurosamente falsa. Pero también es falso "Dios se ha creado a sí mismo", ya que entonces entra como algo creado por él mismo entre las cosas creadas; pero al entrar entre las cosas creadas, éstas no se han autocreado, en contra de la hipótesis.

No es fácil proponer ejemplos de paradojas de Russell. Las que se conocen son bastante traídas por los pelos. ¿Alguien conoce a un hombre que no se afeite nunca? O la de la lista de listas que no hacen referencia a sí mismas, etc. En cambio "Dios no se ha creado a sí mismo" hace siglos que se viene diciendo y, mucho antes, "Dios ha creado todo lo que existe".

Es muy curioso que la paradoja de Dios no se haya formulado antes. Releyendo la famosa discusión, recuperada ahora en la revista electrónica Tendencias21 de la Universidad de Comillas<sup>2</sup>, entre Russell y el padre Copleston S.J., sobre la existencia de Dios (1948), uno se asombra de que no se le ocurriera al propio Russell. Ver sus argumentos en [2]. Un análisis actual de dicha discusión puede verse en la misma revista<sup>3</sup>.

Pero ¿quiere esto decir que, a pesar de que no tiene sentido la frase "Dios no se ha creado a sí mismo", éste no existe? No. Sólo quiere decir que expresamos muchas cosas que no tienen sentido, que no entendemos, que chocan con la razón. Para el no creyente Dios es un producto de la mente humana, como parte del lenguaje, un producto social. Para el creyente la existencia de Dios es su fe.

Ahora bien el intento creacionista de "explicar" desde esa fe lo que no entendemos del mundo, incluyéndonos a nosotros, aunque fuese verdad no puede ser válido para un científico, creyente o no creyente, que ha de ver en esa "explicación" un atajo inadmisibles. Lo

que aún no se entiende hay que intentar entenderlo alcanzando nuevo conocimiento, ¿habrá que decir "científico"? Valga la redundancia. Sin atajos, sin violar la razón. Desde el "Epur si mueve" galileano hasta ahora han pasado más de cuatro siglos en los que la ciencia sigue su camino, siempre preguntándose antes de contestar, siempre volviendo a preguntarse si la contestación no concuerda con lo que se observa del mundo.

"Los creacionistas, en cambio, parece que lo saben todo de antemano", Bernard Wood *dixit*. Intentan justificar la fe dando respuesta con ella a preguntas que no deben ser respondidas desde la fe, sino desde la razón. El creacionismo no sigue el hilo de los argumentos de la razón:

1. Dios no se ha creado a sí mismo es falso, por la paradoja de Dios.
2. La referencia del tiempo y el espacio, que son percepciones humanas, no son el momento y el punto en que Dios creó el mundo, como yo le escuché decir en sus clases de Relatividad a D. Julio Palacios (curso 1960-61). El referente absoluto no tiene cabida en la ciencia; es decir en la cabeza humana. Las teorías que se han esforzado en mantenerlo no tienen validez, como la teoría de la relatividad de Palacios [3], que no forma grupo. Tampoco se mantiene en pie el llamado "momento 0". ¿Quién entiende la frase "**antes** de ese instante no había nada"?

Por tanto la evolución se debe intentar explicar por la ciencia.

El creacionismo va contra la evolución y hace propaganda de los premios Nobel creacionistas. Entre ellos el químico Anfinsen, por poner un ejemplo. Citemos una noticia aparecida en el periódico New York Times del día 19 de agosto de 1986: *72 premios Nobel reclaman a la Corte que elimine las leyes creacionistas, en referencia a las pretensiones de Louisiana de enseñar creacionismo en las escuelas*. En ese manifiesto los firmantes afirman claramente: *El creacionismo no tiene ningún fundamento dentro de la ciencia legítima*. Ésta fue la primera vez que un número tan amplio de laureados con el Nobel firmaban un manifiesto de cualquier tipo. Uno de los firmantes fue Christian B. Anfinsen. Por tanto, si Anfinsen es considerado creacionista por webs afines a este movimiento religioso, éste es un error mayúsculo. Una referencia fiable de eminentes científicos creyentes, evolucionistas o creacionistas, es el libro [4].

No hay que justificar la fe. Simplemente hay que respetarla, pero siempre que no se intente explicar con ella lo que aún no se entiende. Si no se puede entender ni explicar la fe, no es de recibo que se pretenda entender y explicar desde ella este mundo, o sea el mundo. Se puede ser científico de prestigio, creyente y evolucionista, como Anfinsen.

Para quien tenga interés en estos temas filosóficos, fronterizos entre ciencia y religión, además de consultar enciclopedias de Filosofía, como la de Oxford, editada en español por la editorial Tecnos, puede navegar por sitios como '*tendencias21*', '*conoze.com*', '*nav/cryf/cosmologiaactual*', etc. Desde luego, a partir de ahora, se habrá de tomar en cuenta la paradoja de Dios. En efecto, los conceptos "causa" y "efecto" en la segunda vía tomista para probar la existencia de Dios, así como los de "ente contingente" y "ente necesario" en la tercera, se tambalean al tomarla en consideración. Para una idea somera de las cinco vías tomistas y sus refutaciones se puede consultar <<http://driverop.com.ar/viastomistas.php>>.

No hay nada de soberbia en intentar entender el mundo. Es una exigencia humana. Ya se preguntó Platón, y no sería el primero: ¿Qué es esto? ¿Qué hago yo aquí? Muy al contrario, hay que tener mucha humildad y mucha paciencia. En fin mi propósito no es ganar acólitos para el ateísmo, no. Mi propósito es despertar la razón.

### Referencias

- [1] Marcus Giaquinto. *The search for certainty. A Philosophical account of Foundations of Mathematics*. Oxford University Press, 2002.
- [2] Bertran Russell. *¿Por qué no soy cristiano?*. Edhasa, 1981.
- [3] Julio Palacios. *Relatividad: Una nueva teoría*. Ed. Espasa Calpe, 1960.
- [4] Antonio Fernández-Rañada. *Los Científicos y Dios*. Ed. Nobel, 2000.

### Notas

- <sup>1</sup> Wikipedia. <[http://en.wikipedia.org/wiki/Russell's\\_paradox](http://en.wikipedia.org/wiki/Russell's_paradox)>.
- <sup>2</sup> <[http://www.tendencias21.net/TENDENCIAS-DE-LAS-RELIGIONES\\_r18.html](http://www.tendencias21.net/TENDENCIAS-DE-LAS-RELIGIONES_r18.html)>.
- <sup>3</sup> <[http://www.tendencias21.net/Recuperada-la-discusion-Russell-Copleston-sobre-la-existencia-de-Dios\\_a2925.html](http://www.tendencias21.net/Recuperada-la-discusion-Russell-Copleston-sobre-la-existencia-de-Dios_a2925.html)>.

### Notas del Editor

Este artículo ha sido seleccionado para su publicación en esta sección de colaboraciones personales a **Novática** en virtud de lo que nos ha parecido una ingeniosa aportación del autor en cuanto a la particularización de la paradoja de Russell (que es considerada hoy en día como parte de la Matemática Fundamental) al caso de Dios.

Por lo demás, siendo obvio que el evolucionismo no ha permitido tampoco vislumbrar una respuesta al famoso "dilema del huevo o la gallina", consideramos el resto del artículo como una visión subjetiva del autor que se enmarca entre las muchas que ha generado este debate filosófico creacionismo-evolucionismo. Debate en el cual nunca va a entrar **Novática** porque obviamente es un *off-topic* para nuestra revista.

# Reconstrucción

Enrique Martín Martín, Cristóbal Pareja Flores

Universidad Complutense de Madrid

<emartinm@fdi.ucm.es>,  
<cpareja@sip.ucm.es>

El enunciado de este problema apareció en el número 198 de **Novática** (marzo-abril 2009, p. 76). Es el problema G de los planteados en el V Concurso Universitario de la Comunidad Autónoma de Madrid (CUPCAM 2007) del que ATI fue entidad colaboradora.

En el enunciado comenzábamos con una frase:

Lo mejor: un ejemplo.

y la reorganizábamos por líneas, palabra por palabra, rellenando con espacios a la derecha de cada palabra hasta formar una matriz:

```
L o
m e j o r :
u n
e j e m p l o .
```

Leyendo ahora esta matriz por columnas, se forma la cadena siguiente, donde hemos cambiado cada espacio por un guión '-' para que se vea mejor (antes y después de la "o" final irían tres guiones):

Lmueoenj-j-e-o-m-r-p-:-l-o—.

El objetivo del problema era, a partir de esta cadena transformada, descubrir qué frase fue la originaria, si es que había alguna.

El procedimiento para resolver este problema es tantear los diferentes tamaños de matriz que pueden haber dado lugar a la cadena, y comprobar que en dicha matriz cada fila es una palabra, esto es, una cadena de caracteres desde el principio, sin blancos intermedios aunque posiblemente terminada con un signo de puntuación. Para la cadena del ejemplo, cuyo tamaño es 32, las únicas matrices que encajan son las de tamaños (ancho×alto) siguientes: 1×32, 2×16, 4×8, 8×4, 16×2 y 32×1. La matriz de 1×32 no es válida ya que en ese caso se trata de la cadena entera, y no forma una palabra válida porque tiene espacios intermedios en blanco. Para 2×16 la matriz sería la siguiente,

```
L u o n
M e e j j e o m r p : l o .
```

que tiene el mismo problema que antes más un signo de puntuación también antes del final. Al probar con 4×8 encontraríamos una solución, y no necesitaríamos probar con ninguna otra matriz ya que en caso de haber varias soluciones el problema nos pide la de menor número de palabras (es decir, filas).

En resumen, por cada par de dimensiones de matriz válidas, se ha de comprobar que cada fila es también válida y que la última acaba en punto. Aunque podríamos generar cada matriz para estas comprobaciones, es más económico extraer las filas directamente de la cadena inicial calculando la correspondencia entre las posiciones de la matriz y las de la secuencia dada. Miremos por ejemplo la matriz de tamaño 4×8. Las letras que irían en cada casilla serían las que ocupan las siguientes posiciones:

0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

Por tanto la primera fila (la número 0) estaría formada por las letras 0,4,8,12,16,20,24 y 28, es decir, las letras en posiciones  $0 + 4*k$ , con  $0 \leq k < 8$ . En general, si la matriz tiene tamaño  $\text{numF} \times \text{anchoF}$ , la fila número n estará formada por las letras en posiciones siguientes:

$$n + \text{numF} * k \quad \text{con } 0 \leq k < \text{anchoF}$$

El código C++ para una función que devuelve la fila número n de la cadena que proviene de una matriz de  $\text{numF} \times \text{anchoF}$  es:

```
string filaNum(string frase, int numF,
int anchoF, int n) {
string linea = «»;
for (int j=0; j<anchoF; j++)
linea = linea + frase[j*numF + n];
return linea;
}
```

Una vez que podemos extraer las filas, hemos de comprobar que cada fila es válida, es decir, que está formada por letras, seguidas posiblemente por un símbolo '.', ',', ':', o ';', y a continuación espacios en blanco:

```
bool filaValida(string linea) {
int longitud = linea.length();
int pos = 0;

//el primer carácter debe ser una letra
bool ok = isalpha(linea[pos]);
//Saltamos las letras:
do {
pos++;
} while ( (pos < longitud) &&
isalpha(linea[pos]));
//Ahora puede haber un símbolo
if ( pos < longitud &&
esSimbolo(linea[pos]) )
pos++;

//Sólo pueden quedar blancos:
while (ok && (pos < longitud)) {
ok = (linea[pos] == ' ');
pos++;
}
return ok;
}
```

```
bool esSimbolo(char c) {
return (c == '.' || c == ',' ||
c == ';' || c == ':');
}
```

Para solucionar el problema comenzamos leyendo el número de frases a reconstruir. Al hacerlo no "consumimos" el salto de línea, por lo que hemos de leer el resto de la línea con getline para no dejar basura. Tras esto trataremos cada caso. Lo primero es leer la cadena y eliminar el

símbolo '#' final, dando lugar a una cadena de tamaño N. Luego tanteamos las dimensiones de la matriz en orden (de menos a más palabras) hasta que encontremos una en la que todas las filas sean válidas. Son válidas como dimensiones los pares de enteros (numF, anchoF) cuyo producto es N; así pues, basta con tantear como numF los divisores de N. Al hacerlo en orden creciente, la primera solución que encontremos será la de menos filas, esto es, la de menos palabras. Si el tanteo de todas las dimensiones válidas es infructuoso (por no contener ninguna una colección de palabras válidas), imprimiremos por pantalla una línea conteniendo "— NO SOLUTION —", en otro caso mostraremos la solución encontrada.

El código de la función principal es:

```
int main() {
    int numCasos, posPuntoFinal;
    cin >> numCasos;
    string basura;
    //Elimina el salto de línea de la entrada
    getline(cin, basura);

    for (int k=1; k<=numCasos; k++) {
        string frase;
        getline(cin, frase);
        const int N = frase.length()-1;
        //quitamos el carácter final «#»
        frase = frase.substr(0, N);

        posPuntoFinal
            = frase.rfind(«.», N-1) + 1; // [1]
        int numF, anchoF;
        numF = 0;
        bool solved = false;
        do {
            numF++;
            //Tanteo de una matriz numF * anchoF = N
            if (N % numF == 0) {
                anchoF = N / numF;

                if ((posPuntoFinal % numF) == 0) { // [1]
                    //El último punto de la frase está
                    //en la última línea
                    int n = 0;
                    bool algunaFilaCompleta = false;
                    bool estaFilaEsValida;
                    do {
                        string fila =
                            filaNum(frase, numF, anchoF, n);
                        if (fila[anchoF-1] != ' ') // [2]
                            algunaFilaCompleta = true;

                        estaFilaEsValida =
                            filaValida(fila);
                        n++;
                    } while(estaFilaEsValida && n<numF);
                    solved = estaFilaEsValida &&
                        algunaFilaCompleta;
                }
            }
        } while ((!solved) && (anchoF > 2)); // [3]
        if (solved)
            escribirFrase(frase, numF, anchoF);
        else
            cout << «— NO SOLUTION —» << endl;
    } //for
}
```

Una vez conocemos las dimensiones numF×anchoF de la matriz válida, mostrar por pantalla la frase original (en el código la función escribirFrase) es sencillo. Para ello utilizamos las mismas ideas que para extraer filas, mostrando en orden los caracteres y eliminando los espacios repetidos; teniendo en cuenta que todas las palabras han de tener un solo espacio al final para separarlas entre ellas, salvo la última.

Para concluir, algunos detalles del código:

[1] Aparte de que todas las filas formen palabras válidas, se ha de comprobar que la última fila acaba en punto. Para ello, basta con que el último punto de la cadena caiga en la última fila. La comprobación de que ese punto está bien colocado ahí (tras las letras) recaerá en filaValida. La función rfind busca hacia atrás la primera aparición del carácter '.' desde la posición final (N-1) de frase.

[2] También hemos de asegurar que al menos una fila ocupa el ancho de la matriz. De no ser así la matriz no sería válida pues el procedimiento inicial habría colocado la frase en una matriz más pequeña.

[3] No comprobamos el caso de una matriz de filas de longitud uno. Nunca podría ser válido, ya que el punto y final vendría en una fila él solo y, por tanto, ésa no sería una fila válida.

## XI Jornadas de Innovación y Calidad del Software

La XI edición de las Jornadas de Innovación y Calidad del Software (II edición de la Conferencia Iberoamericana de Calidad del Software) se va a celebrar los próximos días 3 y 4 de septiembre de 2009 en la Universidad de Alcalá de Henares <<http://www.uah.es/>> coincidiendo con la prestigiosa conferencia EuroSPI <<http://2009.eurospi.net>> dedicada a la mejora de procesos de software, por primera vez organizada en España.

En esta ocasión, las ponencias seleccionadas (sólo un 41% de las remitidas han tenido cabida en las jornadas) de entre una gran variedad de propuestas conforman un programa completo donde se incluyen ponencias invitadas que tratan temas de gran interés como los servicios del Laboratorio Nacional de Calidad del Software, el desarrollo del nuevo estándar de pruebas de software ISO 29119 o el desarrollo en entornos SAP utilizando modelos UML.

Estas jornadas están apoyadas por la Universidad de Alcalá a través de su vicerrectorado de Investigación e Innovación y cuentan también con el apoyo de DEISER <<http://www.deiser.com/>>.

Para facilitar la asistencia en estos tiempos de crisis, la organización de las jornadas ha optado por un esquema *low-cost* para ofertar el mejor precio: sólo 145 euros para los socios de ATI y de las organizaciones con convenio.

Toda la información está disponible accediendo directamente a la página de las jornadas <<http://www.ati.es/spip.php?rubrique62>> o dirigiéndose a la página del grupo de Calidad del Software de ATI <<http://www.ati.es/spip.php?rubrique233>> donde también puede encontrarse información gratuita de todas las actividades realizadas desde 1997.

## Programación de Novática

Por acuerdo de los Consejos Editoriales de **Novática** y **UPGRADE**, los temas y editores invitados de las restantes monografías del año 2009 y la primera de 2010 serán, salvo causas de fuerza mayor o imprevistos, los siguientes:

Nº 200 (julio-agosto): "Calidad del Software". Editores invitados: **Luis Fernández Sanz** (Universidad de Alcalá) y **Darren Dalcher** (National Centre for Project Management, Reino Unido).

Nº 201 (septiembre-octubre): "En el 20º Aniversario de CEPIS: Presente y futuro de la informática en Europa". Editores invitados: **Fiona Fanning** (Responsable de Comunicación de CEPIS), **Nello Scarabottolo** (Universidad de Milan, Tesorero Honorario de CEPIS) y **Bob McLaughlin** (Queen's University of Belfast, British Computer Society).

Nº 202 (noviembre-diciembre): "Gestión de identidades y privacidad". Editores invitados: **Javier López Muñoz** (Universidad de Málaga) y **Miguel Soriano Ibáñez** (Universitat Politècnica de Catalunya).

Nº 203 (enero-febrero): "2010, tendencias tecnológicas". Editor invitado: **Alonso Alvarez García** (Observatorio Tecnológico de Telefónica I+D).

## 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 asociativa, 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](mailto:info@ati.es)> o a cualquiera de las sedes de ATI. En la actualidad son socios institucionales de ATI las siguientes empresas y entidades:

AGENCIA DE INFOR. Y COMUN. COMUNIDAD DE MADRID  
 AGROSEGURO, S.A.  
 AIGÜES TER LLOBREGAT  
 ALMIRALL PRODEFARMA, S.A.  
 BARCELÓ CORPORACIÓN EMPRESARIAL, S.A.  
 BBR INGENIERÍA DE SERVICIOS, S.L.  
 BURKE FORMACION, S.A.  
 CÁLCULO, S.A.  
 CENTRO DE ESTUDIOS VELAZQUEZ S.A. (C.E. Adams)  
 CHOICE, S.A.  
 CONSULTORES SAYMA, S.A.  
 CONSULTORÍA DE DATOS ESPAÑA  
 DEPARTAMENT D'ENSENYAMENT DE LA GENERALITAT  
 DIGITAL PARKS DATA CENTER SERVICES, S.L  
 ELOGOS, S. L.  
 EPISER, S.L.  
 ESPECIALIDADES ELÉCTRICAS, S.A. (ESPELSA)  
 ESTEVE QUÍMICA, S.A.  
 FUNDACIÓ BARCELONA MEDIA - UNIVERSITAT POMPEU FABRA  
 FUNDACIÓ CATALANA DE L'ESPLAI  
 FUNDACIÓN SAN VALERO  
 GIN-KO  
 GRUPO CORPORATIVO GFI INFORMÁTICA, S.A.  
 IN2  
 INFORMÁTICA Y COMUNICACIONES AVANZADAS, S.L.  
 INSTITUT D'ESTUDIS CATALANS  
 INSTITUT MUNICIPAL D'INFORMÀTICA  
 INVERAMA  
 ITUM TECHNOLOGY S.A.  
 KRITER SOFTWARE, S.L.  
 NTR - NET TRANSMIT & RECEIVE, S.L.  
 OCCIDENTAL HOTELES MANAGEMENT, S.A.  
 ONDATA INTERNATIONAL, S.L.  
 PRACTIA CONSULTING, S.L.  
 QRP MANAGEMENT METHODS INTERNATIONAL  
 SOFTWARE, S.L.  
 SADIEL, S.A.  
 SCATI LABS, S.A.  
 SISTEMAS TÉCNICOS LOTERIAS ESTADO (STL)  
 SOCIEDAD DE REDES ELECTRÓNICAS Y SERVICIOS, S.A.  
 SOGETI ESPAÑA, S.L.  
 SOPORTES, SISTEMAS, SOFTWARE, S.L.  
 SQS, S.A.  
 TECNOLOGIA Y CALIDAD DE SOFTWARE, S.A.  
 TISA ORDENADORES, S.A.  
 TRAINING & ENTERPRISE RESOURCES  
 T-SYSTEMS ITC Services España S.A.  
 UNIVERSIDAD ANTONIO DE NEBRIFA  
 UNIVERSIDAD DE EXTREMADURA - E. POLITÉCNICA DE CÁCERES  
 UNIVERSITAT DE GIRONA  
 UNIVERSITAT OBERTA DE CATALUNYA

Rellene esta hoja y envíela a:

**Novática (Suscripciones)**

Vía Laietana 46, ppal, 1ª

08003 Barcelona, España

Tlfno.: 93 412 52 35 Fax: 93 412 77 13

Correo elec.: <novatica@ati.es>

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 .....

**Tarifas 2009.** 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: 62,00 € (+4% IVA)
- \_ suscripciones: 58,00 € cada una (+4% IVA)

**\* Otros países de la Unión Europea y Marruecos**

- 1 suscripción: 75,00 €
- \_ suscripciones: 70,00 € cada una

**\* Resto del mundo**

- 1 suscripción: 82,00 €
- \_ suscripciones: 76,00 €

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

De conformidad con la LO 15/99 de Protección de Datos de carácter personal le informamos que los datos que usted nos facilite serán incorporados a un fichero propiedad de Asociación de Técnicos de Informática para poder gestionar su suscripción a nuestra revista y hacerle llegar los ejemplares correspondientes. Si usted desea acceder, rectificar, cancelar u oponerse al tratamiento de sus datos puede dirigirse por escrito a secregen@ati.es.

**DATOS BANCARIOS PARA DOMICILIACION**

Banco/Caja.....

CÓDIGO CUENTA CLIENTE			
ENTIDAD	OFICINA	D.C.	NÚMERO DE CUENTA

NV 199



**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

Todos los datos son obligatorios a menos que se indique otra cosa

Una vez cumplimentada esta hoja, se ruega enviarla por correo electrónico a [secregen@ati.es](mailto:secregen@ati.es), o por fax al 93 4127713, o por correo postal a ATI, Vía Laietana 46, ppal. 1ª, 08003 Barcelona

► Solicito inscribirme como: **Socio de número**  (76€)\* / **Socio junior**  (22€)\* / **Socio jubilado**  (25€)\* / **Socio adherido**  (53€)\*  
(Para inscribirse como **socio estudiante** se ruega utilizar la hoja de inscripción específica disponible en <http://www.ati.es/estudiantes>)

\* **Cuota anual para el año 2009** – Ver en la siguiente página información detallada sobre ATI y los diferentes tipos de socios

**- Datos personales del solicitante**

Apellidos		
Nombre		
Domicilio	Nº	Piso
Localidad	Código Postal	
Provincia	Teléfono	
Dirección de correo electrónico <sup>1</sup>		
Fecha de nacimiento	DNI	

**- Datos de la empresa o entidad donde trabaja (si es autónomo indíquelo en el campo "Empresa o entidad")**

Empresa o entidad	Sector
Puesto actual	Depto.
Dirección	Nº
Localidad	Código Postal
Provincia	Teléfono

**- Domiciliación de la cuota anual (ATI se encarga de su envío al banco o caja)**

Nombre de la entidad bancaria: \_\_\_\_\_

Código de entidad	Oficina	D.C.	Cuenta

**- Datos complementarios (si necesita más espacio para estos datos continúe en otra hoja)**

Títulos superiores o medios que posee y centros otorgantes: .....

.....

Resumen de experiencias profesionales: .....

.....

Número de años de experiencia profesional informática: .....

**- Presentado por los Socios de número (\*\*)**

(\*\*) Esta información no es necesaria para solicitar inscribirse como socio junior, estudiante o adherido; para inscribirse como socio de número o jubilado, si el solicitante no conoce a ningún socio de número que pueda presentarle, la Secretaría General de ATI le contactará para determinar otra forma fehaciente de acreditar su profesionalidad.

1) Apellidos y Nombre ..... Nº de socio ..... Fecha .././..... Firma

2) Apellidos y Nombre ..... Nº de socio ..... Fecha .././..... Firma

Firma del solicitante

Fecha \_\_\_\_\_

Mediante su firma el solicitante declara que todos los datos incluidos en esta solicitud son ciertos.

**Nota sobre protección de datos de carácter personal:** De conformidad con la LO 15/99 de Protección de Datos de Carácter Personal, le informamos de que los datos que usted nos facilite serán incorporados a un fichero propiedad de Asociación de Técnicos de Informática (ATI) para poder disfrutar de los servicios que su condición de socio le confiere, así como para enviarle información acerca de nuevos servicios, ofertas y cursos que ATI ofrezca y puedan resultar de su interés. Sus datos podrán ser comunicados a aquellas instituciones, sociedades u organismos, con los que ATI mantenga acuerdos de colaboración, relacionados con el sector de los seguros, la banca y la formación para el envío de información comercial. Si usted desea acceder, rectificar, cancelar u oponerse al tratamiento de sus datos puede dirigirse por escrito a [secregen@ati.es](mailto:secregen@ati.es).

No deseo recibir información comercial de ATI ni de terceras entidades colaboradoras de ATI.

No deseo recibir información comercial de terceras entidades colaboradoras de ATI.

No autorizo la comunicación de mis datos a terceras entidades colaboradoras de ATI.

<sup>1</sup> Una vez validados por la Secretaría de ATI la hoja de inscripción y los documentos requeridos, Vd. recibirá en esta dirección de correo la información sobre el procedimiento para poder utilizar todos los servicios de la red ATINET (ver reverso).



## Una asociación abierta a todos los informáticos

## Una asociación útil a sus socios, útil a la Sociedad

Creada en 1967, **ATI (Asociación de Técnicos de Informática)** es la asociación profesional más numerosa, activa y antigua de las existentes en el Sector Informático español, con sedes en Barcelona (sede general), Madrid, Valencia y Zaragoza. Cuenta con 4.000 socios, que ejercen sus funciones como profesionales informáticos en empresas o Administraciones Públicas, o como autónomos.

ATI, que está abierta a todos profesionales informáticos independientemente de su titulación, representa oficialmente a los informáticos de nuestro país en Europa (a través de CEPIS, entidad que coordina a asociaciones que representan a más de 400.000 profesionales informáticos de 32 países europeos) y en todo el mundo (a través de IFIP, entidad promovida por la UNESCO para coordinar trabajos de Universidades y Centros de Investigación), y pertenece a la CLEI (Centro Latinoamericano de Estudios en Informática). ATI tiene también un acuerdo de colaboración con ACM (*Association for Computing Machinery*).

En el plano interno tiene establecidos acuerdos de colaboración o vinculación con Ada Spain, ASTIC (Asociación Profesional del Cuerpo Superior de Sistemas y Tecnologías de la Información de la Administración del Estado), Hispalinux, AI2 (Asociación de Ingenieros en Informática), Colegios de Ingenierías Informáticas de Cataluña y con RITSI (Reunión de Estudiantes de Ingenierías Técnicas y Superiores de Informática).

### Tipos de socio

- ✓ **Socios de número:** deben acreditar un mínimo de tres años de experiencia profesional informática (o dos años si se posee un título de grado superior o medio), o *bien* poseer un título de grado superior o medio relacionado con las Tecnologías de Información, o *bien* haber desarrollado estudios, trabajos, o investigaciones relevantes sobre dichas tecnologías
- ✓ **Socios estudiantes:** deben acreditar estar matriculados en un centro docente cuya titulación dé acceso a la condición de Socio de Número (hoja específica de inscripción para socios estudiantes disponible en <http://www.ati.es/estudiantes>)
- ✓ **Socios junior:** la condición de socio junior de un profesional informático será válida hasta el día final del año en que cumpla los 30 años de edad
- ✓ **Socios jubilados (Aula de Experiencia):** socios de ATI que, al jubilarse y cesar su actividad laboral, deciden continuar perteneciendo a ATI colaborando con su experiencia con la asociación
- ✓ **Socios adheridos:** profesionales informáticos que no cumplan las condiciones para ser Socios de Número o también personas que, no siendo profesionales informáticos, quieran participar en las actividades de ATI
- ✓ **Socios institucionales:** personas jurídicas, de carácter público o privado, que quieran participar en las actividades de ATI (para más información sobre esta modalidad se ruega ponerse en contacto con la sede general de ATI)

### ¿Qué servicios ofrece ATI a sus socios?

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 de información**
  - Revistas bimestrales **Novática** (decana de la prensa informática española), **REICIS** (Revista Española de Innovación, Calidad e Ingeniería del Software) y **UPGRADE**, publicación digital editada en lengua inglesa por **Novática** en nombre de CEPIS
  - Red asociativa **ATInet** (IntrATInet, acceso básico gratuito a Internet, correo electrónico con dirección permanente, listas de distribución generales y especializadas, foros, blogs, página personal, ...)
  - Servidor web <http://www.ati.es>, pionero de los webs asociativos españoles
- ✓ **Servicios profesionales**
  - Asesoramiento profesional y legal
  - Peritajes, diagnósticos y certificaciones
  - Bolsa de Trabajo
  - Emisión en España del certificado profesional europeo EUCIP (*European Certification of Informatics Professionals*)
  - Emisión en España del certificado ECDL (*European Computer Driving License*) para usuarios
- ✓ **Servicios personales**
  - Los que ofrece la Mutua de los Ingenieros (Seguros, Fondo de pensiones, Servicios Médicos)
  - Los que ofrece la Caja de Ingenieros (gozar de las ventajas de ser socio de esta caja cooperativa)
  - Promociones y ofertas comerciales

### ¿Dónde está ATI?

- ✓ **Sede General y Capítulo de Catalunya** - Vía Laietana 46 ppal. 1ª, 08003 Barcelona - Tlf. 93 4125235; fax 93 4127713 / <secregen@ati.es>
- ✓ **Capítulo de Andalucía** - <secreand@ati.es>
- ✓ **Capítulo de Aragón** - Lagasca 9, 3-B, 50006 Zaragoza - Tlf./fax 976 235181 / <secreara@ati.es>
- ✓ **Capítulo de Galicia** - <secregal@ati.es>
- ✓ **Capítulo de Madrid** - Padilla 66, 3º dcha., 28006 Madrid - Tlf. 91 4029391; fax. 91 3093685 / <secremdr@ati.es>
- ✓ **Capítulo de Valencia y Murcia** - Reino de Valencia 14, 46005 Valencia - Tlf./fax 96 3918531 / <secreval@ati.es>
- ✓ **Revistas Novática, REICIS y UPGRADE** - Padilla 66, 3º, dcha., 28006 Madrid - Tlf. 91 4029391; fax. 91 3093685 / <novatica@ati.es>