



4^{as} Jornadas Internacionales

gvSIG

Avanzando juntos

www.gvsig.gva.es
www.gvsig.org

PATROCINADORES HONOR



PATROCINADORES ORO



PATROCINADORES PLATA



PATROCINADORES BRONCE



COLABORADORES



MEDIOS ASOCIADOS



CONSELLERIAS COLABORADORAS

CONSELLERIA DE E'DUCACIÓ CONSELLERIA D'INDÚSTRIA, COMERC I INNOVACIÓ
CONSELLERIA DE MEDI AMBIENT, AIGUA, URBANISME I HABITATGE CONSELLERIA DE SANITAT



Open Planet²



Sumario

Cuartas Jornadas gvSIG. Cuatro años de avance, cuatro años de ilusión	2
Gestión de la colaboración	3
Supervisión de fábrica	7
'Arquitecturando' gvSIG	10
Internacionalización	17
Testeo colaborativo	20
Documentación	23
Tour 2008	24
Entrevistas Comunidad gvSIG	26
Comunidad gvSIG: la Universidad	28
<i>Lluís Vicents.</i> Miembro del equipo de proyectos SIG del Servicio de Sistemas de Información Geográfica y Teledetección (SIGTE) de la Universitat de Girona, y miembro del equipo docente del Máster UNIGIS	
<i>Antoni Pérez.</i> Estudis d'Informàtica, Multimèdia i Telecomunicació. Universitat Oberta de Catalunya	
Comunidad gvSIG: Europa	33
<i>Alessandro Sgambati.</i> Spec. tec. Alessandro SGAMBATI. Responsabile dell' Ufficio cartografico Ispettorato ripartimentale foreste di Trieste e Gorizia. Regione Autonoma Friuli Venezia Giulia	
<i>Chris Puttick.</i> CIO Oxford Archaeology: Exploring the Human Journey	
Comunidad gvSIG: Latinoamérica	34
<i>Doctor Carlos Figueira.</i> Presidente del Centro Nacional de Tecnologías de Información. Ministerio del Poder Popular para las Telecomunicaciones y la Informática. Gobierno Bolivariano de Venezuela	
<i>Ingeniero Juan Ernesto Rickert.</i> Especialista en Infraestructuras de Datos Espaciales. Jefe de Tecnología del Instituto Geográfico Militar de la República Argentina. Codirector del proyecto PROSIGA	
Comunidad gvSIG: España	36
<i>Juan Antonio Bermejo Domínguez.</i> Técnico GIS (Sistemas de Información Geográfica). Dirección Insular de Nuevas Tecnologías y Energías Renovables. Área de Planificación y Servicios Públicos. Exmo. Cabildo Insular de La Palma	
gvSIG en la Web	37
gvSIG en la Blogosfera	39

Open Planet n°2

Producción

Edit Lin editorial

Avda. de Portugal, 85-local
28011.- Madrid

Coordinación

Álvaro Anguix

Redactores

Gabriel Carrión, Jorge Gaspar, Nacho Brodín, Joaquín del Cerro, Cesar Ordiñana, José Ignacio Yarza, Mario Carrera, Manuel Madrid, Victoria Agazzi y Víctor Acevedo

Licencia

Creative Commons. Reconocimiento-No comercial-Compartir bajo la misma licencia 3.0 Unported

Fotografías bajo licencia Creative Commons cortesía de: Michel Filian (pág. 3); Peter Huys (págs. 7-8); William Murphy (pág. 13); Eddy Van 3000 (pág. 17); Comisión Europea (pág. 18); cyancey (págs. 20-21), Eric James Sarmiento (pág. 23); Fiona (pág. 35).

Cuatro años de avance, cuatro años de ilusión

Gabriel Carrión Rico
Director del Proyecto gvSIG

Nos encontramos con las cuartas jornadas de gvSIG. Un proyecto de Software Libre que año tras año sigue creciendo fiel a sus valores de colaboración y de solidaridad, de independencia tecnológica y de desarrollo sostenible. Y en estas cuartas jornadas, más amplias, más internacionales, con mayor convocatoria que nunca, y también hay que decirlo, con mayores dificultades que nunca pues como ya saben, estamos en el año de "la crisis", creemos conveniente hacer un repaso de la evolución de gvSIG a través de sus jornadas y más concretamente a través de cuales han sido sus diferentes lemas.



"gvSIG: Comparte el conocimiento"

En 2005, cuando llevábamos aproximadamente un año y medio en el desarrollo del proyecto gvSIG decidimos organizar nuestras primeras jornadas. "gvSIG: Comparte el conocimiento" fue el lema de las mismas. Nuestro propósito fue realizar una convocatoria pública y abierta donde pudiéramos enseñar qué es lo que habíamos hecho en ese año y medio y cuáles eran nuestros proyectos de futuro. "Comparte el conocimiento" era el lema, pues ése creemos que es el principal motor del modelo de desarrollo del Software Libre, el conocimiento compartido frente a la especulación con el conocimiento adquirido propio de los modelos privativos.

"gvSIG: Construyendo realidades"

En aquellos momentos gvSIG ya era una herramienta utilizable en ámbitos profesionales y así se manifestó en las segundas jornadas que tuvieron lugar en el 2006, cuando nos bajamos del estrado y nos sentamos entre el público para que fuera la gente, para que fuera la Comunidad la que tuviera ocasión de darnos sus proyectos. Los proyectos que se habían construido con gvSIG. "gvSIG: Construyendo realidades" fue el lema de las segundas jornadas donde se presentaban ya resultados visibles y se comenzaba a atisbar la proyección de gvSIG.

"gvSIG: Consolidar y Avanzar"

A partir de ese momento fue espectacular la progresión; el crecimiento de empresas, universidades e instituciones interesadas en el proyecto, el crecimiento de las listas de correo; mensajes con propuestas que éramos incapaces de abarcar con tranquilidad, al menos con la tranquilidad que nos fuera posible y de afianzar los resultados obtenidos.

"gvSIG: Consolidar y Avanzar" fue el lema de las terceras jornadas. Teníamos que Consolidar los resultados obtenidos y prepararnos para seguir avanzando, pero de forma ordenada y sostenible.

"gvSIG: Avanzando Juntos"

En este último año hemos estado trabajando en estas tareas de consolidación del proyecto, en crear una estructura que asegure la calidad del mismo y que aproveche todo el potencial de la Comunidad.

Que se aproveche toda esa energía existente que quiere, que nos pide, que nos propone contribuir en el desarrollo y difusión de gvSIG. Teníamos que trabajar para crear una infraestructura que permitiera poner a gvSIG en manos de la Comunidad. Una infraestructura que facilitara la colaboración, porque no debemos olvidar que facilitar la colaboración es favorecer el desarrollo.

Hemos trabajado en la arquitectura del proyecto, en la organización de su desarrollo, en la internacionalización, en diseñar un modelo de testeo colaborativo, en potenciar la visión de proyecto colectivo y solidario. En crear un ecosistema que se entienda, que se coordine, que sea capaz de organizarse a sí mismo, donde tengan cabida todas las opiniones y que la toma de decisiones sea transparente. Eso sí, siempre desde los principios éticos de colaboración y solidaridad, de independencia tecnológica y eficiencia en la gestión que se promueven desde gvSIG.

Queremos aprovechar para hacer el llamamiento a todos aquellos que estén interesados en colaborar, en aportar al proyecto que se pongan en contacto con nosotros. Cualquier cosa que penséis que podéis aportar, como participar en el testeo de la herramienta, en la internacionalización traduciendo o supervisando documentación, en generar materiales de

formación o en ser desarrolladores que quieran participar en el mantenimiento y sostenibilidad de gvSIG, bien a título individual bien a través de la organización a la que pertenezcáis. Estaremos encantados de recibir vuestras propuestas y ver cómo podemos atenderlas. Nuestro objetivo es ser ese elemento de coordinación que proporcione un punto de encuentro a la Comunidad gvSIG.

En esta revista encontrarás varios artículos sobre los trabajos que estamos realizando para conseguir que el lema de las cuartas jornadas se convierta en una realidad. "gvSIG: Avanzando Juntos".

Estamos creando esta infraestructura que os queremos presentar y que nos debe permitir Avanzar Juntos. Pretendemos que estas jornadas supongan un punto de encuentro, un lugar de debate, de intercambio de ideas y propuestas que nos permita avanzar, que nos permita que todos juntos podamos seguir construyendo gvSIG.



> Gabriel Carrión Rico.

Gestión de colaboración

Este artículo trata de exponer al lector las diversas áreas en las que el equipo gvSIG debe gestionar tanto los recursos propios del proyecto, como los que la comunidad pone a su disposición. Además, el proyecto tiene un compromiso adquirido con la comunidad para tratar de ofrecer la mejor documentación, buscando en todo momento cumplir los criterios de calidad que la comunidad se merece. Mejorar estos medios de comunicación y documentación es una tarea diaria, que evoluciona con el proyecto y, por lo tanto, un reto constante.

A lo largo del artículo se presentarán las diferentes herramientas con las que la comunidad cuenta para interactuar con el proyecto, para buscar documentación, para acceder al código fuente, cómo gvSIG interactúa con otros proyectos y posibles nuevas vías de colaboración.

Listas de correo

La base de la comunicación entre los participantes en el proyecto, bien como usuarios, bien como desarrolladores está en las listas de correo. Éstas permiten recibir dudas, inquietudes y aportaciones del resto de usuarios, así como exponerlas. De esta forma, se abren temas de interés que reciben respuesta de otros usuarios. También son un medio para que los desarrolladores de gvSIG reciban 'feedback' del trabajo publicado y una vía de comunicación entre la gente encargada de gestionar el proyecto y los usuarios.

Hay tres listas de correo a las que se puede acceder a través de la página de gvSIG. Desde allí se ofrece la posibilidad de consulta de todos los mensajes a través de una página web, o bien la posibilidad de suscribirnos. Ésta es la forma en la que podemos participar con nuestros comentarios recibiendo en nuestra cuenta de correo toda la información que llegue referente al proyecto.

De las tres listas oficiales hay dos en castellano, una en la que se tratan temas referentes al uso de gvSIG y otra de desarrollo donde se habla de aspectos más técnicos, relativos al trabajo a nivel de código o distribución. Otra lista, llamada internacional, viene a cubrir el grupo de usuarios de habla no hispana y que generalmente se comunican en inglés. En esta última aparecen temas de todo tipo.

Existe otra lista de distribución de gvSIG en italiano alojada en el dominio de gvSIG, pero administrada por usuarios

de la aplicación de este país. En ella no se da soporte directamente por parte del proyecto, pero cuenta con una masa de usuarios que puede ayudar a solucionar muchas de las dudas y problemas surgidos con la aplicación.

Las listas de correo empezaron su actividad en septiembre del año 2004. Desde entonces se han ido incorporando más y más subscriptores, y el número de mensajes ha ido creciendo. Actualmente la lista de usuarios es la que más participación tiene, cuenta con un total de 1.088 inscritos y cerca de 6.000 correos en los tres últimos años.

La lista de desarrollo en castellano tiene 567 adscritos. Es sorprendente que, en volumen de inscritos, sea la mitad que la de la lista de usuarios. Esto indica que mucha gente desarrolla o al menos ha comenzado algún desarrollo con gvSIG.

La evolución de la cantidad de información en las listas es muy significativa, ya que es un indicativo de la variación del volumen de usuarios y uso que se hace de la aplicación. Por ejemplo, en la lista de usuarios la constante siempre ha sido incrementar el nivel de tráfico, en parte debido al hecho de que ha habido un cambio de actitud de los usuarios frente a la lista. En un principio la lista únicamente era el soporte técnico donde la gente podía apoyarse y el grupo de trabajo del proyecto aportaba, en la medida de sus posibilidades, respuesta a esa demanda. Poco a poco los propios usuarios han sido capaces de ir dando soluciones y

OBJETIVOS GENERALES DEL PORTAL

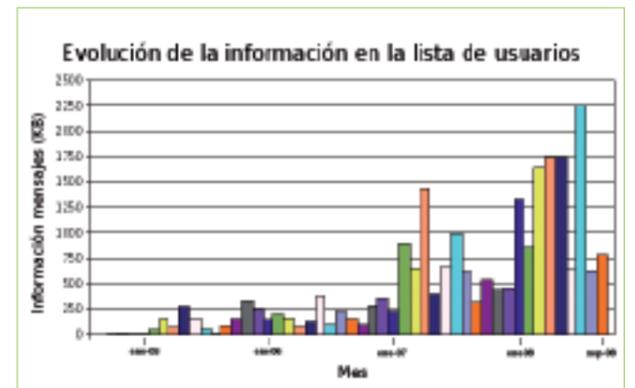
- Gestión de la documentación de usuario.
- Gestión de la documentación de desarrollo.
- Mantenimiento de la información de los diferentes subproyectos.
- Gestión de la documentación organizativa.
- Publicación de noticias y eventos.
- Publicación del catálogo de extensiones.
- Mantenimiento de traducciones de la documentación.

respuesta a inquietudes que por cuenta propia han desarrollado.

También la temática de los mensajes ha variado un poco. Está claro que los correos de soporte siguen abundando y es imprescindible que lo hagan, pero otro tipo de hilos donde se discute problemas de SIG más generales, se aportan datos para mejorar, o se ponen en conocimiento iniciativas personales en relación a gvSIG están dando paso a hilos de mensajes donde la comunicación es más fluida y se aporta igual o más que se recibe.

El portal colaborativo

El portal colaborativo de gvSIG [1] es una web implementada sobre Plone [2].



> Estadísticas de mensajes.

Este gestor de contenidos permite disponer de gran cantidad de funcionalidades, tanto en su instalación base como a través de extensiones y personalizaciones que se han ido realizando en función de las necesidades específicas que un proyecto como gvSIG requiere.

Este portal, junto con el repositorio de código fuente, forma el verdadero centro neurálgico de trabajo para la mayoría del equipo del proyecto, y cada vez más se espera que también lo sea para la comunidad a través de las distintas iniciativas que se están potenciando.

Documentación de usuario: del PDF a la web

La documentación de usuario de gvSIG tradicionalmente se ofrecía a los usuarios a través del portal oficial como documentos PDF. Este formato, útil para imprimir, resulta incómodo para realizar búsquedas y en general resulta demasiado "estático" para un proyecto tan dinámico como gvSIG.

En cualquier caso se seguirá generando el manual en PDF, pero a partir de la documentación 'web' como un producto derivado, una verdadera 'fotografía' de la misma en un instante concreto.

Así, utilizar el portal colaborativo como lugar donde publicar la documentación de usuario es algo lógico y de indudable interés para la comunidad. El portal permite a los documentadores publicar con rapidez cualquier corrección o mejora en el manual, sin tener que esperar a una nueva versión de gvSIG o a una revisión completa del mismo.

Otra ventaja evidente es que al utilizar una publicación 'web' la documentación se vuelve más interactiva al permitir incluir hipervínculos entre otras partes de la documentación o incluso externos al portal. El aprovechamiento de esta nueva "característica" se podrá ver de forma gradual conforme se vaya actualizando el manual de usuario.

Por último, y se apunta aquí solamente porque es tratado en otro artículo, el portal colaborativo permite la traducción de la documentación de usuario (en realidad cualquier documentación) a otros idiomas. Esto abre una puerta tremendamente interesante para el proyecto, además de un reto organizativo importante: la participación de la comunidad en la traducción de textos. En un ámbito tan específico como el de un SIG de escritorio como es gvSIG, es difícil que en el proyecto se encuen-

ten especialistas en todos los idiomas. Es mucho más factible "reclutar" voluntarios, especialmente nativos, que traduzcan la documentación con rigor.

Documentación de desarrollo

Al contrario que otros proyectos (y esto es una característica tal vez innovadora del proyecto, dentro del ámbito del Software Libre), gvSIG se ha caracterizado por una mayor atención al usuario final que a los desarrolladores de la aplicación. Con esto ha conseguido una masa crítica de usuarios que ha potenciado la llegada de desarrolladores de todo el mundo que "demandan" más documentación técnica que permita desarrollar sus propias personalizaciones y adaptaciones de gvSIG.

Así, el portal colaborativo se configura como el punto de entrada para cualquier desarrollador que quiera utilizar gvSIG como plataforma tecnológica, a través de la documentación técnica del proyecto. En este sentido, cabe destacar la necesidad del proyecto de localizar colaboradores que traduzcan la documentación (actualmente toda en español) al inglés para dar cabida a un mayor número de desarrolladores, fuera del ámbito hispano-hablante.

En cualquier caso, hay que tener presente que en un proyecto del tamaño y actividad de gvSIG la documentación técnica difícilmente va a estar nunca "completa". Muy al contrario, con el tiempo surgirán nuevas necesidades. Así, una vez cubierta una documentación general de la aplicación surgirá la necesidad de publicar una lista de preguntas frecuentes (FAQ's) y pequeños tutoriales cortos para realizar tareas concretas (HOW-TO's).

Catálogo de extensiones

gvSIG es proyecto que pretende que la comunidad se beneficie no sólo del

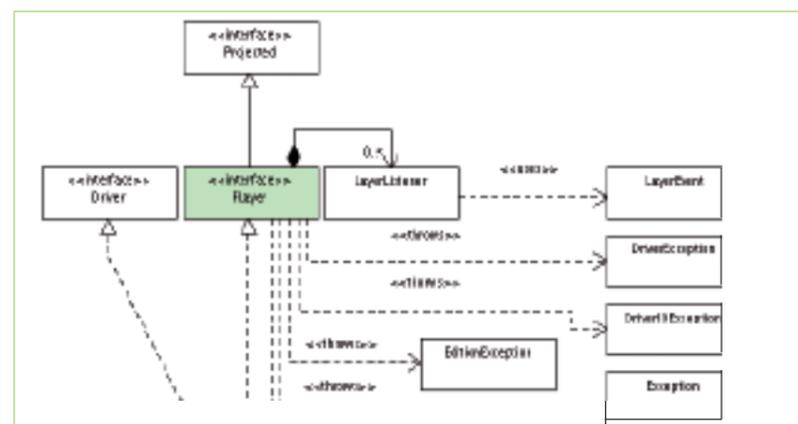
software que se produce en el seno del proyecto, sino de cualquier otra contribución que la propia comunidad genere. Es decir, muchos actores (programadores independientes, empresas, administraciones y universidades) están desarrollando sus propias extensiones y personalizaciones de gvSIG y creen que pueden ser de utilidad para toda la comunidad de usuarios y desarrolladores.

Dada esta necesidad, el proyecto ha configurado un área en su portal colaborativo para dar publicidad a estas nuevas contribuciones. Este área es, por tanto, un catálogo de proyectos [3] (grandes y pequeños) que aportan nuevas funcionalidades (a veces muy experimentales o simples prototipos), nuevos recursos o nueva documentación.

Cabe destacar que el catálogo es eso, un catálogo. En el portal no se 'almacenan' los proyectos ni se proporciona infraestructura colaborativa para las aportaciones, sino que se configura como un verdadero **escaparate** para aquellas personas o entidades que quieran difundir su trabajo. Es decir, este catálogo ofrece a los desarrolladores e instituciones un lugar donde exponer un poco más su trabajo.

Por otro lado hay que remarcar que estas contribuciones con carácter general no son testadas por el proyecto y, por tanto, no son "oficialmente soportadas". Es decir, gvSIG como proyecto no se puede responsabilizar del buen funcionamiento de las mismas, ni dar soporte a través de las listas.

Si su organización o empresa ha desarrollado algún nuevo plugin, ha mejorado algún recurso o ha desarrollado documentación sobre gvSIG que cree que es de interés para el resto de la comunidad, visite el manual del catálogo [4] para obtener más información sobre el proceso.



> Ejemplo de diagrama UML en la documentación técnica.

El repositorio de código fuente

Un repositorio de código es un espacio en un servidor donde se almacena el código fuente de uno o varios proyectos. Todos los desarrolladores que trabajan con él están conectados a ese repositorio y disponen de una copia del proyecto o parte de éste actualizada. Cuando terminan un trabajo o piensan que lo que tienen está en un estado óptimo para ser compartido, se sincronizan con el repositorio de fuentes y "suben" su aportación de trabajo para que el resto de personas puedan sincronizarse y tener las nuevas modificaciones para seguir evolucionando. Ni que decir tiene que esta forma de trabajo requiere, por parte de los desarrolladores, un especial cuidado a la hora de trabajar para impedir situaciones de conflicto en las que puede que el código no funcione, haya errores a la hora de compilar, problemas de dependencias o cosas por el estilo.

Este tipo de herramientas dispone de otras capacidades que aportan flexibilidad al trabajo en grupo. Entre ellas cabe destacar la gestión de un histórico de cambios en el que podemos consultar qué modificaciones se han hecho y qué usuario las hizo. También dispone de herramientas para la resolución de conflictos entre cambios realizados por distintos usuarios sobre los mismos ficheros. Esta resolución pasa por realizar una mezcla de los cambios aportados dejando los más interesantes.

Una de las funciones principales podría ser la gestión de distintas líneas de desarrollo dentro del mismo repositorio llamadas "branches". Como su propio nombre indica, son ramas de desarrollo en las que se bifurca un proyecto. Estas líneas son evoluciones distintas de la aplicación, que pueden corresponder con distintas versiones que llevarán una vida propia o trabajos puntuales de I+D de una determinada funcionalidad que al acabar confluirá de nuevo en la línea principal del proyecto.

El repositorio de gvSIG

Entre las tecnologías disponibles se eligió el sistema de control de versiones llamado "subversión" que suponía una evolución del anterior utilizado "CVS". Desde septiembre de 2008 este repositorio de código [5] está disponible en modo de sólo lectura para cualquier usuario que desee descargar el código fuente más actual. Los desarrolladores salvan allí

diariamente modificaciones que amplían y mejoran las posibilidades de gvSIG. El repositorio tiene una estructura de carpetas donde cada versión y cada proyecto tiene su espacio.

gvSIG se encuentra inmerso en un verdadero ecosistema de proyectos de muy diversa índole, tanto del mundo geoespacial como de otros ámbitos

La rama principal de desarrollo donde se encuentran las funcionalidades más avanzadas, pero al mismo tiempo menos estables, están en la carpeta 'trunk'. Así mismo, otras ramas de desarrollo están en la carpeta 'branches' donde se puede encontrar una por cada versión de gvSIG o parte de éste que ha ido apareciendo a lo largo del tiempo. Cada empaquetado que se hace de una versión del proyecto recibe el nombre de "build" y tiene asignado un número. De esta forma, para una versión concreta de gvSIG se tienen múltiples 'builds' que se generan a lo largo de la vida de esa versión. Esos 'builds' se almacenan en la carpeta "tags" del repositorio de fuentes.

El acceso al código fuente de gvSIG, de esta forma tan directa, da muchas posibilidades a los desarrolladores que deseen modificar el proyecto y/o aportar soluciones a problemas a la comunidad. Gracias a la característica de extensibilidad que tiene gvSIG, cualquiera puede descargar el código fuente de un gvSIG básico y programar una extensión que cubra una necesidad particular. Cualquier desarrollador externo puede aportar cambios o soluciones a los proyectos que constituyen gvSIG en forma de lo que se denominan "parches". Un parche no es más que uno o varios ficheros que describen las modificaciones que hay que hacer en el código para resolver el problema o incorporar la funcionalidad.

En principio, la generación de parches es muy sencilla ya que el entorno de desarrollo Eclipse [6], que es el recomendado para gvSIG, los genera automáticamente. Esto lo hace comparando el código que había inicialmente en el repositorio y el que nosotros hemos creado. Cuando un desarrollador genera un parche que soluciona algún tipo de 'bug' o aporta alguna mejora, tiene la posibilidad de incorporarlo al proyecto.

Actualmente el canal de comunicación para realizar aportaciones de este tipo

pasa por comunicarlo en las listas de correo de gvSIG. Hay previsto un protocolo mucho más adecuado para este tipo de procedimientos, que en breve se va a poner en marcha, en el que el usuario, a través de un formulario en la web, podrá comunicar 'bugs' detectados o aportar los cambios que los solucionan.

Este formulario dará de alta un 'ticket' que será distribuido al equipo de mantenimiento de la parte que está afectada y que aplicará la solución para que aparezca corregido en la siguiente versión, y con carácter inmediato sea corregido en el repositorio de fuentes.

Colaboración con otros proyectos e iniciativas

OSGeo

→ Incubación en OSGeo

gvSIG no es un proyecto aislado, muy al contrario se encuentra inmerso en un verdadero ecosistema de proyectos de muy diversa índole, tanto del mundo geoespacial como de otros ámbitos. Dentro del ámbito geoespacial, desde 2006 existe una fundación de carácter internacional que trata de ser el punto común de reunión para los proyectos libres del ámbito de la geomática. La "Open Source for Geospatial Foundation" (OSGeo [7]) podría asimilarse, en cierta medida, al equivalente a la fundación Apache [8] en el área geoespacial. Esta fundación reúne a, probablemente, los mejores proyectos geoespaciales, tanto en el área de las bibliotecas de componentes, como los servidores de mapas y metadatos o los SIG de escritorio.

Todos los proyectos pasan por un proceso conocido como "incubación", en el que la Fundación pide a los proyectos que hagan una auditoría de sí mismos. En este proceso los proyectos se evalúan tanto técnica como organizativamente.

El tiempo de incubación puede ser más o menos largo, normalmente en función de la complejidad del proyecto. A cada proyecto incubado se le asigna un mentor, una persona más o menos relevante y de prestigio en la comunidad de OSGeo que velará porque la revisión sea honesta y veraz.

Una vez el proyecto ha superado todos los requisitos que la fundación exige, éste pasa a ser un proyecto "graduado", nombrándose al representante del mismo vicepresidente de la Fundación.

En la actualidad los proyectos que se encuentran en OSGeo son los que vemos en la **Tabla 1**.

Web mapping	SIG de escritorio	Bibliotecas	Metadatos	Otros
Mapbender MapBuilder OpenLayers MagGuide OS UMN MapServer deegree	GRASS GIS Quantum GIS <i>gvSIG</i> <i>OSSIM</i>	FDO GDAL/OGR GeoTools <i>GEOS</i> MetaCRS	Geonetwork OS	Geodatos públicos Educación

TABLA 1. Proyectos en OSGEO.

*en cursiva los que están en incubación.

gvSIG en OSGeo

gvSIG es proyecto en incubación en OSGeo desde el otoño de 2007. El mentor de gvSIG es Jeroen Ticheler, líder de GeoNetwork OpenSource [9], el servidor de metadatos de referencia (aunque no es el único) de la comunidad geoespacial. En este tiempo se han realizado algunos avances en el proyecto que se alinean con los objetivos de la incubación (lo cual no es lo mismo que decir que se deben a la misma). Como hechos más destacados se pueden marcar la publicación de la documentación de desarrollo, la participación en eventos relacionados con OSGeo y especialmente la apertura del repositorio de código fuente. Este último es uno de los aspectos técnicos más demandados por la comunidad de desarrolladores que finalmente se ha producido.

El siguiente reto que deberá acometerse durante el próximo año es documentar con exactitud los procesos internos de gobierno del proyecto, en especial del área técnica (la que afecta directamente a los proyectos de software que gvSIG, como organización, acomete).

Otros proyectos

La participación de gvSIG en otros proyectos es una necesidad. gvSIG se nutre de un gran número de bibliotecas y utiliza igualmente otros proyectos en su infraestructura colaborativa. Colaborar con todos estos redundará en mejorar gvSIG. Así, cuando se encuentra un error en una biblioteca o componente, es responsabilidad del equipo de desarrollo comunicarlo a los desarrolladores de esa biblioteca. Chris Schmidt [10]



> Jeroen Ticheler.

car estos errores para que en próximas versiones se subsanen y así se mejore la estabilidad de gvSIG.

Así, desarrolladores de gvSIG participan en muchos proyectos (no sólo del ámbito geoespacial) como UMN Mapserver, GeoTools, GDAL, Maven, etc.

Participación en eventos internacionales

La participación en eventos relacionados con la geomática o con el Software Libre juega un papel importante en la gestión de la colaboración. En función del contexto de estos eventos se establecen relaciones a diferentes niveles. En eventos de alto contenido técnico las relaciones entre desarrolladores, pueden favorecer la integración de diferentes tecnologías o la resolución conjunta de problemas similares. En eventos relacionados con las IDE o las Administraciones Públicas, se pueden establecer relaciones de colaboración de más alto nivel donde instituciones (empresas y administración pública) pueden favorecer líneas de actuación que impulsen la promoción del Software Libre, el fomento del empleo de estándares abiertos o la difusión de geodatos libres. De igual forma, las empresas que participan en el desarrollo de gvSIG, también obtienen beneficios comerciales de la participación en este tipo de eventos.

Otras formas de colaboración

La gestión de la colaboración es un área de la organización que se encuentra con retos continuos. El equipo debe adaptarse a las necesidades que van surgiendo y

muchas veces actuar con agilidad para dar respuesta a una situación cambiante como es el ecosistema de usuarios, empresas, administraciones públicas y centros de investigación que trabajan con gvSIG.

Así, actualmente existen nuevas actividades que si bien no se han abordado con profusión, están dentro del 'radar' de actividades que se pueden empezar. Una lista no exhaustiva podría ser:

- > Redacción de casos de éxito y su publicación tanto en el portal del proyecto como en otras revistas y portales del sector.
- > Posibilidad de implantar un 'planeta' que aglutine los bálagos y sitios de noticias relacionadas con gvSIG.
- > Organizar eventos técnicos orientados exclusivamente a desarrolladores, tanto de dentro como de fuera del proyecto, también conocidos como "hacker meetings".

Conclusiones

La gestión de la colaboración es un área horizontal en el proyecto, involucrando a todos los niveles de participación. Desde los propios usuarios que gestionan los grupos de testeo o traducción, hasta los actores de más alto nivel del proyecto. Todos tenemos un compromiso de transparencia con la comunidad, intentando en la medida de lo posible ofrecer lo mejor del proyecto a ella.

Este objetivo es duro y ambicioso, no siempre se cumple por completo, pero no hay duda que el proyecto gvSIG es un éxito gracias al esfuerzo de todos.

REFERENCIAS

- [1] <https://gvSIG.org>
- [2] <http://www.plone.org>
- [3] <https://gvSIG.org/plugins/downloads>
- [4] https://gvSIG.org/web/community/participate/repository/manual_repo
- [5] <http://subversion.gvSIG.org/gvSIG>
- [6] <http://www.eclipse.org>
- [7] <http://www.osgeo.org>
- [8] <http://www.apache.org>
- [9] <http://geonetwork-opensource.org/>
- [10] <http://www.flickr.com/photos/crschmidt/2905716041/>



Supervisión de fábrica

Joaquín del Cerro
Responsable de fábrica gvSIG
IVER

Gestión y coordinación de fábrica

Sinceramente resulta complicado delimitar las tareas de gestión y coordinación de lo que podríamos denominar "fábrica gvSIG", principalmente porque la complejidad creciente del proyecto, compuesto a su vez por diversos proyectos (mobile, 3D, raster, redes, etc.), no permite definir estas tareas de modo fijo. Aun así, básicamente hay tres grandes grupos de tareas:

- > Estar al tanto del estado de los distintos proyectos que componen gvSIG. Qué planificaciones tienen, qué equipo está trabajando en cada proyecto o cuál es el avance del mismo.
- > Estar al tanto, a grandes rasgos, de la arquitectura general de cada proyecto. No con el objetivo de conocer con detalle todos los proyectos, sino para detectar posibles confluencias entre proyectos e intentar que éstas coincidan en el tiempo.
- > Detectar las tareas que hay que ir llevando a cabo alrededor del desarrollo de gvSIG y generar la documentación e infraestructura necesarias para que dichas tareas se puedan realizar, unas veces generándolas desde dentro del seno del grupo de organización del desarrollo y otras intentando que se formen los grupos adecuados para su tratamiento.

Para hacer estas cosas, y con el tamaño de proyecto que actualmente

tiene gvSIG, la cantidad de trabajo requiere que no sólo lo lleve a cabo una persona. Dentro de lo que es la organización del desarrollo ha sido necesario crear un grupo que se encarga de revisar arquitectura, acometer parte de las tareas de seguimiento de los proyectos y sincronizar la generación de toda la documentación que esto conlleva.

Es crucial para mí, como responsable de fábrica, la labor que se realiza desde estos grupos de trabajo, siendo los pilares en los que me apoyo para detectar deficiencias de infraestructura o sinergias entre proyectos.

La necesidad de organizar y planificar el desarrollo

Muchas veces los técnicos no ven la necesidad de llevar a cabo estas tareas de supervisión general de los distintos proyectos. Están acostumbrados a un tamaño de proyecto en el que un sólo equipo de trabajo hace su análisis, su diseño e implementa la solución final.

El problema con el que nos encontramos cuando comenzó el crecimiento de gvSIG es que era muy difícil que cada grupo de trabajo encargado de un proyecto dentro de gvSIG, supiera en lo que estaban trabajando los otros. La gente intentaba seguir trabajando como cuando era un proyecto pequeño y nos encontrábamos con un escenario en que equipos de desarrollo aislados apenas interactuaban entre sí, y si lo hacían dependía más del nivel de relación que había entre los miembros de los distintos proyectos que de cualquier otra razón.

Actualmente estamos adaptándonos a un nuevo modelo de desarrollo en el que, sin duda, es necesario organizar y planificar el desarrollo a nivel global. Así pues, es necesario plantear la construcción de gvSIG desde una perspectiva diferente, de mayor envergadura. En el proceso de adoptar esta perspectiva encontramos que aparecen varias formas de hacer lo mismo y no sólo a nivel de arquitectura o librerías. Por ejemplo, se han dado casos en que dos equipos han desarrollado las mismas funcionalidades de usuario porque no sabían que lo estaban haciendo los otros.

Estos casos, que pueden ser los más llamativos, son sólo la parte que se ve cuando miras un poco. Cuando intentas meterte en el modo en que cada proyecto implementa una determinada funcionalidad, te encuentras con porciones de código de aquí y allá que hacen básicamente lo mismo. Esto, si no se controlara, haría muy difícil mantener la estabilidad de la aplicación y los futuros desarrollos.

Por todos estos motivos se ha formado el equipo que lleva la organización del desarrollo, con el objetivo principal de intentar paliar estas situaciones, en la medida de lo posible.

Los problemas del día a día

Hay varias categorías de problemas con los que tenemos que lidiar desde nuestra área de trabajo:

→ **Deficiencias en la infraestructura del proyecto.** Se detecta una deficiencia en la infraestructura técnica en la

que se sustenta algo, sea en la web, el repositorio de fuentes, o en la gestión de traducciones de la aplicación, por decir algunos ejemplos, y hay que encauzar que alguien se encargue de elaborar una propuesta de cómo tratar ese problema, supervisando las soluciones técnicas que aporta desde la visión general del proyecto.

→ **Asignación de tareas en las fronteras entre proyectos.** Un problema importante que aparece a menudo es determinar quién puede abordar la realización de determinadas tareas. Los recursos humanos de que dispone cada proyecto o grupo de trabajo son limitados y generalmente van ligados al desarrollo de nuevas funcionalidades de la aplicación.

Ha sido necesario crear un grupo que se encarga de revisar la arquitectura, a comer parte de las tareas de seguimiento de los proyectos y sincronizar la generación de toda la documentación que esto conlleva

Para muchas tareas es difícil discernir de qué grupo de trabajo dentro de gvSIG es responsabilidad hacerlas. Así es normal que si estas tareas están en la zona que se aproxima a los límites entre proyectos, cada uno de ellos tienda a dejarlas de lado o dedicar poco esfuerzo por tratarse de algo que generalmente es auxiliar dentro de las metas de su proyecto. Suele ser muy difícil determinar cuáles son las fronteras entre proyectos y qué responsabilidad deben asumir cada uno de los proyectos que confluyen en esa frontera.

→ **Resistencia a ser observados y a aceptar decisiones de arquitectura y diseño por parte de los grupos de trabajo de los proyectos.** Éste ha sido uno de los problemas más conflictivos con los que nos hemos encontrado. Al crecer y aumentar en tan poco espacio de tiempo la magnitud del proyecto, hemos pasado de la actuación "yo me lo guiso, yo me lo como" de los desarrolladores, a una actuación en la que esos mismos desarrolladores tienen que dar cuentas de lo que hacen y muchas veces cambiar la forma en cómo diseñarían o implementarían, con el fin de intentar alcanzar objetivos que están relacionados con otros desarrollos con

los que, de forma natural, no se sienten identificados. Este cambio de actitud en los desarrolladores muchas veces es difícil de aceptar y suele ser fuente de conflictos. Ya no trabajas para implementar como mejor puedas la solución a tu proyecto, has pasado a trabajar para dar soporte a los desarrollos de otros, y no siempre podrás hacer las cosas como quieras. Suele ser muy difícil determinar cuáles son las fronteras entre proyectos y qué responsabilidad deben asumir cada uno de los proyectos que confluyen en esa frontera.

→ **La financiación del proyecto.** Aunque a primera vista esto no parece que tenga que ver con la parte de organización del desarrollo, repercute directamente sobre ésta.

Un proyecto como gvSIG no sólo requiere el desarrollo de unas funcionalidades contratadas por tal o cual administración, sino que se hace necesario abordar una serie de tareas que repercuten directamente en la necesidad de horas de desarrollo. El grueso de los desarrollos de gvSIG ha nacido en el contexto de migración de la CIT a Software Libre. La CIT ha sacado contratos para desarrollar las herramientas que existen en gvSIG. Como es normal, desde un punto de vista tradicional de desarrollo de software, los adjudicatarios de estos contratos intentaban verlos como contratos al uso: me das unas especificaciones funcionales y te entrego una aplicación. Los desarrollos de gvSIG, por su propia naturaleza, son algo más que un conjunto de especificaciones funcionales y es el propio proyecto gvSIG el que debe orientar en la construcción de la solución. Afortunadamente esta concepción clásica tiende a remitir y las empresas que desarrollan gvSIG entienden, cada vez más, que es un desarrollo donde el 'cliente', sin dejar de serlo, es un colaborador.

Proceso de generación de nuevas versiones

Uno de los procesos más delicados, dentro del desarrollo de gvSIG, es la generación de una nueva versión.

Hay que ver qué desarrollos coinciden en el tiempo, para entregar un juego de funcionalidades que estos consideren estable.

Una vez se ha observado que uno o varios equipos de desarrollo pueden entregar un juego de funcionalidades

estable, se genera un instalable con él, y se pasa al equipo de testeo para que le de un vistazo y realice un informe subjetivo que evalúe el estado de madurez del entregable. Si se considera que es aceptable, se montan dos equipos de estabilización, uno encargado del testeo y otro del desarrollo, y se fija un tiempo máximo para obtener un entregable.

El equipo de desarrollo implicado es un equipo especializado en estabilizar. A diferencia de los equipos normales de desarrollo, tiene como prioridad la estabilidad de la aplicación frente a la incorporación de nuevas funcionalidades o una "codificación elegante". Normalmente este equipo consta de dos o tres personas que derriban y supervisan los arreglos que se realizan durante el periodo de estabilización. En este equipo cobran una especial importancia los papeles de responsable de la estabilización y del empaquetado.

El equipo de testeo se dará un periodo de dos o tres semanas para revisar el instalable entregado y obtener una lista de errores a corregir. Con esta lista de errores, el equipo de estabilización realizará una estimación y planificación de cuánto tiempo pueden llevar las correcciones y se negociará con testeo cuáles deben corregirse y con qué errores se puede asumir publicar. Teniendo en cuenta esto, si el tiempo de estabilización se excede del tiempo marcado para realizar esta tarea, se desestimarán la salida a la calle de la nueva versión y se devolverá el control a los equipos de desarrollo para que mejoren la estabilidad del producto. En caso de que se decida continuar con el proceso de estabilización, se prepara un calendario de corrección de errores entre los responsables de testeo y desarrollo.

Cada día sincronizarán sus tareas y se coordinarán en la corrección de errores, teniendo como meta conseguir que cada dos o tres días se entregue un instalable a testeo con un número reducido de errores corregidos sobre el que se puede controlar su impacto.

El periodo de estabilización no debería exceder de mes o mes y medio, y al final de éste se obtiene una Release Candidate (RC), que se publica. Unos días antes de que esto suceda, se empiezan a ensamblar las piezas de la documentación de usuario a entregar con la versión, y normalmente se intenta

tener una versión preliminar de esta documentación en un plazo de una semana, a partir de la publicación de la primera RC.

Nos encontramos inmersos en un proceso constante de apertura y transparencia de las acciones que realizamos en cuestión de desarrollo y hasta que éste no se complete satisfactoriamente, es difícil encontrar el modelo adecuado para la colaboración

En estos momentos hay que intentar coordinar a varios equipos que trabajan alrededor de gvSIG intentando conseguir:

- > Disponer del instalable de la aplicación, responsabilidad del equipo de estabilización (tanto el de desarrollo como el de testeo).
- > Debe confluir la generación de la documentación de usuario para la primera RC.
- > El equipo encargado de preparar la traducción de la documentación de usuario al inglés entra en juego en cuanto se dispone de la versión en castellano.
- > Se preparan las traducciones del interfaz de la aplicación a los distintos idiomas que están soportados.
- > Se preparan los avisos de la aparición de las distintas RC y sus traducciones a varios idiomas para su divulgación por la Oficina de Comunicación. Y al final de todo esto aparecen en la calle las primeras RC de una versión de gvSIG.

Proceso de la documentación de desarrollo

Lo que le pedíamos a la documentación de desarrollo ha ido evolucionando a lo largo de la vida del proyecto. En el momento en que empezó a considerarse importante disponer de una documentación de desarrollo para ofrecer a la comunidad, se intentó disponer de la documentación de análisis y diseño que se había utilizado durante el desarrollo de cada uno de los proyectos. Esto ocasionó sus problemas, ya que cada equipo de desarrollo seguía sus propias metodologías de desarrollo, generando la documentación de análisis y diseño de formas distintas y en formatos distintos.

Así que lo primero que se persiguió fue intentar que todos adoptasen el mismo soporte y formato para esta documentación, seguido de una normalización sobre qué se debía disponer documentado y unas reglas mínimas que diesen homogeneidad a esta documentación.

Una vez puesto esto en marcha, nos permitía entregar a la comunidad la documentación de desarrollo usada dentro del proyecto o, dicho de otro modo, transmitir el conocimiento del equipo al exterior.

Una vez dispusimos de una documentación base, nos planteamos documentar 'mínimamente' los desarrollos anteriores, por lo menos en lo que al núcleo de gvSIG se refiere, y se dedicaron un par de personas a realizar estas tareas.

Actualmente estamos intentando que, además de la documentación de análisis y diseño, para las librerías de la aplicación se disponga de una documentación mínima que nos diga cómo usarlas, junto con la documentación a modo de javadocs de las APIs de éstas.

El refactoring de gvSIG

Hace casi un año se empezaron a realizar tareas de reconstrucción de algunas partes de la aplicación, principalmente la parte de la aplicación relacionada con el acceso a datos. Muchas veces nos referimos a ello como el "refactoring del acceso a datos", aunque más adecuado que refactoring sería el término "reconstrucción".

Se detectaron una serie de errores que eran inherentes a la arquitectura que tenía gvSIG en lo referente al acceso a datos, y se decidió abordar la corrección de estos realizando un nuevo diseño e implementación de toda la capa de acceso a datos de gvSIG.

La meta principal de toda esta labor era dotar de más estabilidad a la aplicación, de una mayor predecibilidad de su comportamiento, y disponer de una API bien acotada que permitiese a los desarrolladores de nuevas funcionalidades de gvSIG, trabajar de forma rápida y cómoda. ¿Ventajas?

- > Una aplicación más estable.
- > Una mayor capacidad de evaluación de impacto al realizar cambios en la aplicación.

> Una curva de aprendizaje más suave para los desarrolladores al disponer de una API clara y documentada, permitiendo que nuevos desarrolladores se sumen al proyecto.

Colaboración de la comunidad en el área de desarrollo

Ahora mismo estamos intentando conseguir una fórmula para que sea más fácil aceptar colaboraciones en torno a gvSIG. Hay ya muchos colaboradores en otras áreas, como internacionalización o testing, gracias al buen trabajo que se ha realizado para facilitar dicha colaboración. El siguiente paso es conseguir que sea sencillo colaborar en el desarrollo.

Se trata de una tarea compleja ya que el número de escenarios que hay es muy elevado y casi todos ellos requieren unos recursos por parte del proyecto que actualmente no disponemos. Existen varios grupos de trabajo en el seno de gvSIG cuya meta principal es la de preparar los mecanismos para la articulación de colaboraciones en torno al proyecto. Nos encontramos en un escenario en el que queremos que colaboren, pero no sabemos cómo articularlo manteniendo la coherencia del código de gvSIG.

Existen varios grupos de trabajo en el seno de gvSIG cuya meta principal es la de preparar los mecanismos para la articulación de colaboraciones en torno al proyecto. Nos encontramos en un escenario en el que queremos que colaboren, pero no sabemos cómo articularlo manteniendo la coherencia del código de gvSIG

Nos encontramos inmersos en un proceso constante de apertura y transparencia de las acciones que realizamos en cuestión de desarrollo y hasta que éste no se complete satisfactoriamente, es difícil encontrar el modelo adecuado para la colaboración. No se puede colaborar en algo que no se conoce. Así mismo, es necesario disponer de un calendario de publicación de versiones estable para que la gente sepa en qué fechas saldrá lo que se está haciendo.

Poco a poco iremos avanzando en intentar disponer de la infraestructura básica para gestionar las colaboraciones.

'Arquiturando' gvSIG

César Ordiñana
Arquitecto Java
DISID

José Ignacio Yarza
Arquitecto Java
Open Sistemas

¿Qué es la Arquitectura?

Veámoslo a través de un par de citas:
En el primer escrito sobre arquitectura que se conoce, Vitruvio decía que "La arquitectura descansa en tres principios: la Belleza (Venustas), la Firmeza (Firmitas) y la Utilidad (Utilitas) que es la base de la utilización y/o función de la arquitectura.

La arquitectura se puede definir, entonces, como un equilibrio entre estos tres elementos, sin sobrepasar ninguno a los otros. No tendría sentido tratar de entender un trabajo de la arquitectura sin aceptar estos tres aspectos".

De Architectura, Marco Vitruvio, año 15 a.C.

En cualquier caso, todos los arquitectos que han escrito sobre ello a lo largo de la historia, tienen su propia definición personal de lo que es la arquitectura. Todas ellas tienen algo en común, a la vez que representan diferentes puntos de vista.

El rol del Arquitecto

... de software?

El término "Arquitecto" se introdujo en el terreno de la ingeniería de software hace un par de décadas para definir un rol necesario en proyectos u organizaciones de cierta complejidad. Se puede decir que se trata de un rol más, dentro de la variedad de roles que puede desempeñar un ingeniero de software.

Se puede considerar un proyecto de software complejo aquel que tiene más de dos grupos de trabajo. Además hay otras variables que pueden aumentar la complejidad, como la dispersión geográfica o los niveles de interdependencia.

Antecedentes

En una organización, normalmente cada equipo de proyecto constituía un grupo de trabajo más o menos cerrado. En una disciplina en la que se produce gran cantidad de conocimiento durante la vida de un proyecto, la falta de comunicación entre los grupos era un grave inconveniente que limitaba la evolución de la organización. Debido principalmente a este problema

de comunicación entre grupos de trabajo, aparecían también problemas de heterogeneidad en conceptos de diseño de software. Es decir, cada equipo aplicaría su criterio técnico durante el diseño de su componente sin apoyarse en otros proyectos ni contribuir a los mismos, "reinventando la rueda" en muchos casos.

Como consecuencia, el repositorio de componentes de la organización a menudo consistía en un variadísimo catálogo de todo tipo de ruedas: redondas, cuadradas, grandes, pequeñas, verdes o azules. Para perpetuar la situación, cuando hacía falta un componente para un nuevo proyecto, el desarrollador optaba por construirlo desde cero, porque era menos costoso que ponerse a buscar y averiguar cuál de los componentes del repositorio podía servir.

El resultado de esta forma de trabajo eran proyectos más costosos y una mala gestión del conocimiento dentro de la organización. Se hizo manifiesta la necesidad de un rol que tuviera una visión de conjunto del software y los proyectos, y que promoviera la buena gestión del conocimiento adquirido en el día a día de los proyectos, así como la homogeneidad de los productos.

Funciones del Arquitecto

En un entorno de organización, la complejidad se deriva del número de proyectos que hay en marcha en un momento dado, en cuyo caso el rol del arquitecto consiste en supervisar ciertas propiedades que

En otra definición, veinte siglos después, se dice que la arquitectura es "un instrumento cuya función central es intervenir en favor del hombre".

James M. Fitch, 1972.

afectan a todos los proyectos, promover diseños que favorezcan la reusabilidad del código, encontrar posibles sinergias entre proyectos o velar por que se alcancen unos parámetros de calidad homogéneos.

En un entorno de proyecto complejo como gvSIG, compuesto por varios subproyectos, donde los grupos de trabajo están geográficamente dispersos, la situación es similar. En este caso, el arquitecto tiene una visión de conjunto del diseño de la aplicación y de todos los proyectos y grupos de trabajo que participan. El arquitecto hace de aglutinador, adquiere y promueve una cultura de proyecto, y define las piezas centrales que constituyen el software y que le permiten crecer de forma ordenada y controlada.

Es quien dice cuál es el tipo de ladrillo estándar para construir la casa y supervisa que no se utilizan ladrillos de diferentes medidas, materiales y formas, ya que, aunque por separado todos valen, no es recomendable mezclarlos porque eso favorecería el anidamiento de insectos en los huecos, además de las goteras.

Arquitectura y conocimiento

Cuando la arquitectura es suficientemente articulada y coherente, la complejidad crece a un ritmo más lento que el software. Sin embargo, si la arquitectura es ambigua o está insuficientemente definida, la complejidad crece a un ritmo mayor que el

propio software, porque éste se comporta de forma orgánica, creciendo cada parte libremente en función de las necesidades de cada subproyecto, con cierta independencia del resto. Finalmente, un software así se hace difícil de comprender, de mantener, de estabilizar y de enriquecer.

Una consecuencia positiva de la buena gestión del conocimiento (ideal en el hipotético caso de que se diera una especie de catarsis colectiva en el seno de un proyecto) es que la organización madura al tiempo que lo hace cada uno de sus integrantes, así como se garantiza un nivel de calidad del software producido.

Hace poco más de un año se creó en gvSIG un grupo de arquitectura y coordinación para velar por estos aspectos del proyecto de forma colegiada.

Grupo de Arquitectura y Coordinación

Algunas de las principales responsabilidades del grupo, tanto las que ya se han puesto en marcha actualmente, como las que se plantean de cara a un futuro, son las que vemos en el **Esquema 1**.

Coordinación/seguimiento de proyectos

Actualmente existe una gran cantidad de proyectos que se realizan para añadir o mejorar funcionalidades a gvSIG, realizados por distintos grupos de trabajo que se encuentran distribuidos entre distintas organizaciones y ubicaciones.

El grupo se encarga de coordinar la ejecución de los distintos proyectos, sobre todo en lo que respecta a las interdependencias entre los distintos proyectos y a las entregas de cada uno de ellos.

Además se prepara y tutoriza a los distintos grupos en el uso de las herramientas colaborativas que tiene el proyecto gvSIG, como por ejemplo:

- > La web del proyecto en www.gvsig.org (Plone).
- > El repositorio de fuentes (Subversion).
- > La gestión de tareas y bugs (Trac).

Supervisión del diseño y arquitectura de proyectos

A nivel de cada proyecto, y de gvSIG como aplicación completa, se supervisa la arquitectura y el diseño del desarrollo que realiza cada uno.

El objetivo es mantener una arquitectura común en la funcionalidad que aportan los distintos proyectos de desarrollo, bien se trate de nuevas librerías o extensiones para gvSIG.

Definición de buenas prácticas de desarrollo

Se definen buenas prácticas a aplicar en el desarrollo de gvSIG. Estas prácticas persiguen obtener:

- > **Mejor calidad de código:** mediante la aplicación de buenas prácticas, fruto de la experiencia de los miembros del equipo en anteriores proyectos, así como de recomendaciones de distintos libros y artículos técnicos, se consigue una calidad de código mayor. Algunos ejemplos serían: uso de interfaces, separación entre API e implementación, aplicación de patrones de diseño, tests unitarios, etc.
- > **Mejor documentación:** para facilitar la lectura y comprensión del código de gvSIG, haciendo hincapié en el Javadoc, que es el mecanismo estándar de Java para documentar el código.
- > **Normalización de nomenclaturas:** para que todo el código de gvSIG siga unas reglas de nombrado comunes, que facilite también el análisis y comprensión del código, así como su uso desde otras partes de gvSIG.

Gestión técnica del núcleo de gvSIG

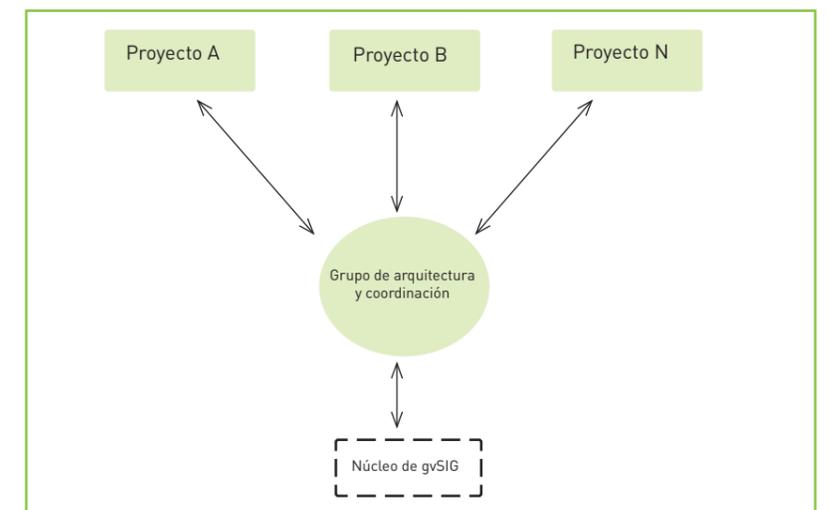
gvSIG es una aplicación extensible a través de extensiones, que aportan gran parte de la funcionalidad disponible en cada una de sus versiones publicadas.

Además de dichas extensiones, gvSIG está compuesto por un núcleo, que define la funcionalidad básica y proporciona los mecanismos necesarios para permitir la extensibilidad de la aplicación.

Los distintos proyectos oficiales que se abordan desde gvSIG suelen tener como objetivo la incorporación de nueva funcionalidad o el enriquecimiento de la misma, a través de extensiones que se instalan junto a dicho núcleo. Tal y como se indica en el primer punto ("Coordinación/seguimiento de proyectos"), el grupo se encarga de coordinar dichos proyectos. Sin embargo, el núcleo de gvSIG no pertenece a ninguno de estos, por lo que el grupo de arquitectura y coordinación se constituye en el gestor del mismo, como si se tratara de un proyecto más.

El núcleo de gvSIG está compuesto principalmente, siguiendo la arquitectura de capas de la aplicación, por lo siguiente:

- > Una serie de librerías de soporte básico, que permiten incorporar funcionalidades a gvSIG que no están relacionadas directamente con la Geomática, como soporte multi-idioma, lectura y escritura de archivos XML, utilidades Java diversas, etc.
- > La lógica geo básica: carga de datos geográficos, visualización y edición de geometrías, etc.
- > Interfaz de usuario: soporte de vistas, documentos y proyectos, así como el resto de apartados en los que pueden añadir sus propias pantallas y opciones las distintas extensiones, como los menús, botoneras, ventanas, paneles de configuración, etc.



> ESQUEMA 1. Grupo de arquitectura y coordinación.

**gvSIG en la actualidad:
gvSIG en números**

gvSIG lleva una progresión continua en cuanto al número de funcionalidades que se van publicando, en forma de versiones que se van liberando periódicamente en la web del proyecto. Hasta la actualidad, el histórico de versiones publicadas, por orden cronológico, es el siguiente:

- > 1.0
- > 1.0.1
- > 1.0.2
- > 1.1
- > 1.1.1
- > 1.1.2

Actualmente, la versión en desarrollo está previsto que se publique como versión 2.0, debido al gran número de nuevas funcionalidades y mejoras en las existentes, así como a los cambios en la arquitectura y organización interna de la aplicación. Además, las versiones anteriores de gvSIG podían funcionar sobre una versión 1.4.2 o superior de Java, mientras que la versión 2.0 está previsto que tenga como requisito la versión 1.5 o superior.

Dicha progresión en funcionalidad, se ve reflejada también a nivel técnico en el aumento de código necesario. En la **página siguiente** se muestran datos (como el número de líneas de código [1]) y algunas gráficas donde se puede observar dicha progresión.

Sobre la versión en desarrollo existen actualmente en marcha más de 12 proyectos oficiales (algunos formarán parte de la versión 2.0), para cada uno de los cuáles existe un responsable y un equipo de desarrollo distintos.

Con esta progresión se puede hacer una idea de la complejidad que está alcanzando el proyecto, no sólo por el tipo de aplicación y su funcionalidad, sino también por el tamaño y la complejidad que tiene actualmente como aplicación. Además hay que unir los siguientes factores:

> **Desarrollo colaborativo:** el desarrollo de las distintas funcionalidades y extensiones de gvSIG se realiza a través de distintos proyectos, en los que colaboran un número importante de instituciones y empresas. Existe, además, la perspectiva de aumentar dicha colaboración de forma significativa a nivel internacional. Para soportar este tipo de colabora-

ción en el desarrollo, y el crecimiento en el número de participantes, gvSIG debe realizar un esfuerzo importante, tanto de cara a la organización de la colaboración, como de cara a facilitar al máximo el desarrollo de nuevas funcionalidades.

> **Compatibilidad con distintos dispositivos y sistemas operativos:** se está realizando un esfuerzo importante de cara a mantener el máximo de compatibilidad hacia JavaME CDC, que es la versión de Java para dispositivos con recursos limitados, objetivo del proyecto gvSIG Mobile. De esta forma, al menos las funciones internas que gestionan los datos geográficos, podrán ser compartidas entre la versión para ordenadores personales y la versión mobile. A todo esto hay que unir que, dentro de la versión para ordenadores personales, gvSIG se distribuye para sistemas operativos Linux, Windows y MacOSX.

Todo este aumento en la complejidad del proyecto ha propiciado la necesidad de realizar un esfuerzo específico en lo que respecta a la arquitectura software de gvSIG, así como en la coordinación entre los distintos proyectos en desarrollo.

**gvSIG en la actualidad:
Arquitectura**

En el **Diagrama 1** se muestra una visión conceptual de la arquitectura de gvSIG. En él se aprecian las capas de la aplicación y las categorías a las que pertenecen los componentes de mayor nivel. A continuación se explica brevemente el diagrama, de izquierda a derecha y de abajo hacia arriba.

Base Tools & Architectural Patterns (Herramientas básicas y patrones de arquitectura)

Este bloque contiene algunas de las piezas y moldes elementales que se utilizan por toda la aplicación. La cantidad y calidad de estas piezas ha ido aumentando progresivamente y la intención es que siga creciendo. Las ventajas de disponer de estos elementos son muchas, principalmente fomentan el crecimiento ordenado de la aplicación, la homogeneidad en el diseño y la comprensión del código. Esto deriva en una aplicación más robusta y unos desarrolladores y usuarios más felices. Aquí se encuentran cosas como:

1. Patrones de diseño como Visitor u Observer.
2. La definición base para las excepciones.
3. La definición base para la persistencia.
4. La definición base para los evaluadores de expresiones.
5. La definición base para las operaciones.
6. Los puntos de extensión, que aportan una técnica de inyección de dependencias.
7. El patrón Locator que aporta una técnica para el registro y obtención de librerías.

Logic GEO (Lógica GEO)

Es el núcleo SIG de gvSIG, también conocido como FMap. En él se encuentra la lógica geomática. Se descompone en:

→ **Data Storage (Almacenamiento de datos)**

Esta capa está compuesta de drivers y filtros que leen y escriben los diferentes formatos de datos (CSV, SHP, DWG, TIFF), que pueden estar almacenados en diferentes medios (File, DB, Http, etc.).

Se integran con gvSIG haciendo uso de la interfaz que la capa de acceso ofrece a sus proveedores (lo que puede denominarse SPI -Service Provider Interface-). En el diagrama se han incluido algunos formatos que vienen "de serie" a modo de ejemplo.

→ **Data Access Layer**

Es la capa de abstracción de acceso a datos. Esta capa permite interactuar con las distintas fuentes de datos de forma homogénea, independientemente del sistema físico que haya detrás.

→ **Map Context**

Es el modelo de mapa. Maneja conceptos como las capas, los símbolos o las leyendas.

→ **Map Control**

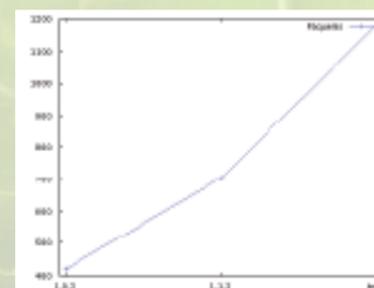
Es el medio de control del modelo del mapa. Se encarga de gestionar la interacción entre el usuario y el mapa mediante controles o herramienta que permiten alterar el estado y propiedades de éste. En mapcontrol se encuentran todas las herramientas de manipulación del mapa y la vista, y al ser extensible los desarrolladores pueden registrar sus propias herramientas.

→ **Geo Data Model (Modelo de datos Geo)**

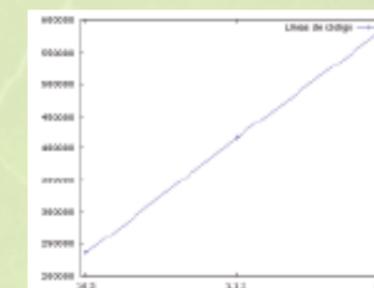
Es el modelo de geometrías. Define la jerarquía de objetos que representan a

Versión	Paquetes	Clases	Métodos	Líneas de código
gvSIG 1.0.2	420	2.797	25.880	235.681
gvSIG 1.1.2	707	5.404	48.812	416.168
gvSIG en desarrollo	1.192	7.782	65.323	589.580

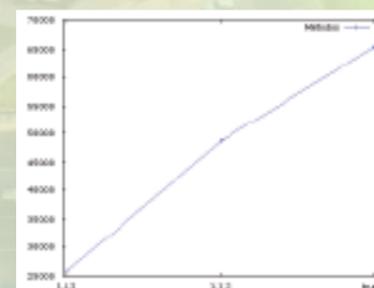
TABLA 2.



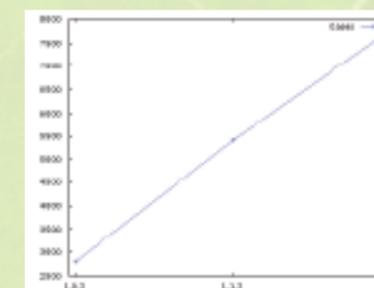
> Evolución del número de paquetes.



> Evolución del número de líneas de código.



> Evolución del número de métodos.



> Evolución del número de clases.

todas las geometrías soportadas, siendo también extensible. La capa de acceso a datos produce objetos de este modelo, unificando los datos procedentes de las distintas fuentes. Mapcontext utiliza estos objetos para componer el mapa y mapcontrol para renderizarlo. Así pues, se trata del modelo de datos sobre el que actúan los procesos geo en todos los niveles.

User Interface

Es la interfaz de usuario. Permite interactuar con la capa geo subyacente y aporta el modelo de documento del proyecto y las herramientas necesarias para su gestión.

Extensions

Llamamos extensión a cualquier componente que aporte nuevas funcionalidades a gvSIG, sin ser una parte del núcleo. Por tanto, dependiendo de a qué capas esté

aportando lógica, la extensión podría situarse a nivel de datos, lógica y/o presentación. gvSIG es extensible en los tres niveles y, por tanto, una extensión podría hacer aportaciones a cualquiera de ellas, o también ser autocontenida e integrarse como un plugin totalmente independiente (es decir, sin adaptarse internamente a las capas de gvSIG).

Cualquier extensión que aporte un nuevo formato/protocolo se situaría a nivel de datos. Como extensión de la lógica geo está, por ejemplo, la parte de 3D que aporta nuevas geometrías y lógica de mapa 3D.

La evolución de gvSIG: entorno de desarrollo

El proyecto gvSIG va a abordar progresivamente una serie de mejoras, en lo que respecta al entorno de desarrollo. Con esto se persigue, por un lado, que el desarrollo de nuevas extensiones de gvSIG

sea lo más sencillo posible. Por otro lado, hay que facilitar que se vaya incorporando un número mayor de proyectos cuyo desarrollo se realiza fuera de los proyectos oficiales.

Algunos de los temas que actualizarán el entorno de desarrollo de gvSIG, desde el punto de vista de la organización del código y los proyectos son:

Maven

Maven es una herramienta de gestión de proyectos, orientada, sobre todo, a aquellos que emplean Java como lenguaje de desarrollo. Es una herramienta que facilita en gran medida la realización de distintas tareas sobre el código, como por ejemplo:

- > Compilación.
- > Generación de archivos .jar.
- > Generación de javadoc.
- > Invocación de tests unitarios.

- > Creación de informes sobre las distintas tareas realizadas.
- > Etc.

A diferencia de otras herramientas, como por ejemplo Ant, se basa en convención sobre configuración por lo que, si se sigue una estructura de proyecto concreta, se minimiza en gran medida la configuración necesaria (aunque podemos adaptarlo a la estructura de proyecto que tengamos a través de dicha configuración).

Con esta premisa, Maven lleva predefinidas las tareas típicas que se suelen realizar en proyectos Java, como los indicados anteriormente, y no necesitaremos crearlas de forma manual.

Ejemplo de estructura de proyecto Maven:

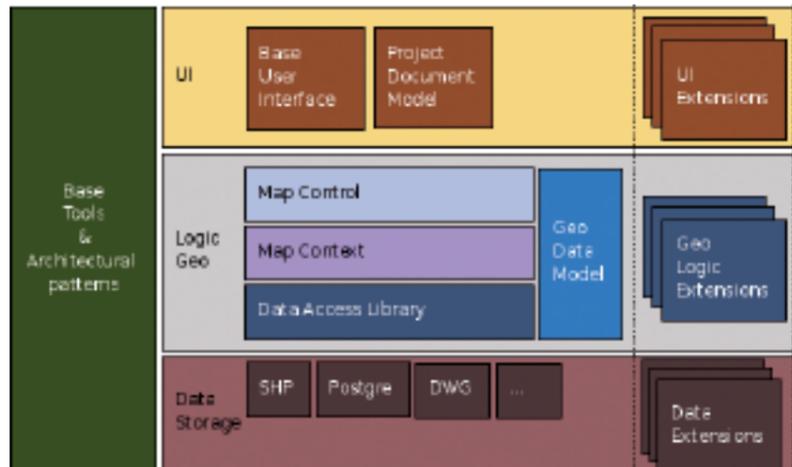
```
my-app
|-- pom.xml
|-- src
|-- main
|   |-- java
|   |   |-- org
|   |   |   |-- project
|   |   |   |-- app
|   |   |   |-- App-
java
|-- test
|   |-- java
|   |   |-- org
|   |   |   |-- project
|   |   |   |-- app
|   |   |   |-- App-
Test.java
```

Otra de las principales ventajas que aporta Maven es la gestión de dependencias. Generalmente, en muchos proyectos, se suelen incluir junto con las fuentes los archivos .jar con las dependencias externas que tenga el proyecto.

Con Maven, en la configuración del proyecto, basta con indicar cuáles son dichas dependencias y él se encarga de descargarlas automáticamente cuando son necesarias, dejando el repositorio de proyecto limpio de archivos binarios.

A continuación se muestra, a modo de ejemplo, cómo definir una dependencia de la librería JUnit, en su versión 3.8.1:

```
<dependencies>
<dependency>
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>3.8.1</version>
</dependency>
</dependencies>
```



> DIAGRAMA 1. Visión conceptual de la arquitectura de gvSIG.

```
<version>3.8.1</version>
<scope>test</scope>
</dependency>
</dependencies>
```

Se puede encontrar más información sobre Maven, documentación e instalables, en la web del proyecto,

<http://maven.apache.org/>

Estructura y versionado de proyectos

Hasta el momento, gvSIG se guía por un versionado completo de toda la aplicación, incluyendo todas las librerías y extensiones que estén disponibles en el momento de publicar cada versión. De hecho, en el repositorio de fuentes, cuando se hace una marca de versión o build de gvSIG, se hace de todos las fuentes disponibles en ese momento.

Sin embargo, dada la evolución del proyecto en el que la aplicación cada vez tiene un número mayor de extensiones disponibles, esta forma de versionado es más costosa y menos práctica.

Por un lado, coexisten al mismo tiempo distintos proyectos para desarrollar nuevas extensiones de gvSIG o mejorar las existentes. Cada uno de esos proyectos tiene una planificación distinta, según el alcance que tenga, por lo que cada vez es más difícil consensuar una fecha común para la liberación de una nueva versión de gvSIG.

Por otro lado, por razones de tamaño, probablemente en un futuro no sea práctico que haya un solo instalable de gvSIG que incluya todas las extensiones disponibles y que además todos los usuarios necesiten tener todas estas extensiones.

Para solventar los dos problemas anteriores, será necesario plantear una nueva organización de proyectos y versionado. Dicha organización de proyectos deberá permitir que cada extensión, o grupo de extensiones por funcionalidad, puedan mantener su versionado y publicación de forma independiente. Así, una extensión podrá publicar nuevas versiones para una misma versión de gvSIG, y a la inversa, una versión de gvSIG será una recopilación de extensiones con su última versión estable.

Otros grandes proyectos de código abierto siguen una política similar, como son, por ejemplo, Eclipse, con sus releases anuales como Ganymede, o también las distintas distribuciones de Linux.

La evolución de gvSIG: Normas de codificación Formato del código

Un código bien ordenado es más fácil de leer y entender. La convención preferida con este fin es la definida por Sun en su "Code Conventions for the Java Programming Language". Una parte de estas convenciones las aplica, por defecto, el propio entorno de desarrollo Eclipse, por ejemplo las extensiones de los ficheros. El resto queda en manos del desarrollador: las tabulaciones, espacios, disposición de bloques anidados o nombres de variables.

Esta convención ya se está aplicando en el núcleo de gvSIG y se pretende seguir en todos los desarrollos relacionados. Lo cierto es que la mayor parte del código ya cumple con una buena parte de estas convenciones, pero hay espacio para mejorarlo. Un ejemplo de código bien formado:

```
SubtypeFeatureType(DefaultFeatureType
parent, String[] names)
throws DataException {
super (parent, false);
DefaultFeatureAttributeDescriptor
attrcopy;
DefaultFeatureAttributeDescriptor
attr;
for (int i = 0; i < names.length;
i++) {
attr =
(DefaultFeatureAttributeDescriptor)
parent
.getAttributeDescriptor(names[i]);
if (attr == null) {
```

```
throw new
SubtypeFeatureTypeNameException(names[i],
parent
.getId());
}
attrcopy = new
DefaultFeatureAttributeDescriptor(attr);
attrcopy.index = i;
}
this.parent = new
WeakReference(parent);
}
```

Normalización de nomenclatura

Se ha definido un conjunto de normas que regulan el uso de prefijos y sufijos en los nombres de las clases o interfaces. Gracias a estas normas, los sufijos y prefijos aportan una semántica coherente al nombre de las clases o interfaces, indicando propiedades de interés para el desarrollador, como por ejemplo si una clase es abstracta o si es la implementación por defecto de una interfaz.

En este sentido se sigue progresando, y según se definen nuevos patrones de arquitectura, se podrán definir nuevas normas que identifiquen el rol de las clases dentro de los mismos, facilitando su comprensión allí donde se utilicen.

Javadoc

Javadoc es una buena herramienta de documentación de código. Permite navegar por la especificación de un componente de forma estructurada y facilita al desarrollador el generar una documentación homogénea ya que establece el conjunto de etiquetas a utilizar, evitando en un extremo el síndrome del papel en blanco y en el otro la "creatividad" excesiva.

Así pues, se está trabajando en la generación de un Javadoc de calidad, en inglés, donde se especifiquen las APIs de que dispone gvSIG. Inicialmente se publicará un Javadoc para el núcleo y progresivamente se irán sumando las extensiones.

Test unitarios

→ JUnit

Para el profano en el tema, un test unitario es un mini-programa que comprueba que un proceso o función de un componente funciona según su especificación. En gvSIG se utiliza JUnit como librería para hacer los tests. Una vez generado el test, éste se puede ejecutar de forma automática siempre

que se desee. Habitualmente se ejecuta automáticamente cada vez que se hace un nuevo build.

Los tests unitarios de código fuente ya se utilizan en gvSIG. De cara al futuro se quiere reforzar su uso, definiendo técnicas para su programación y formas de controlar la calidad del test.

→ Mock Objects

En ocasiones no es fácil hacer un test, especialmente cuando el método a probar requiere de otros objetos complejos de construir desde cero, o que dependen de objetos externos para determinar su estado. Para ayudar a resolver este problema, que puede desanimar al desarrollador a la hora de hacer su test, se quiere fomentar el uso de Mock Objects. Los Mock Objects son objetos 'vacíos' que implementan la misma interfaz que los objetos complejos mencionados anteriormente. Lógicamente para poder utilizarlos hay que programar 'hacia' interfaces, de lo contrario, no podremos suplantar la identidad de los auténticos objetos con los de 'pega'.

→ Cobertura de los tests

Otra cosa que se quiere promover es el uso de herramientas de cobertura. Éstas miden el nivel de cobertura de código fuente que tiene una batería de tests, o dicho de otra forma, cuánto código está efectivamente probando el test. Esta medida sirve para hacerse una idea de lo bueno que es un test (aunque es necesaria, no es suficiente), porque nos da información de qué partes del código están siendo probadas y qué partes no. Si una batería de tests consigue cubrir un alto porcentaje del código del programa, entonces hay una mayor garantía de que el test, efectivamente, da información fiable sobre lo robusto que es el código.

Herramientas de cobertura disponibles para JUnit está, por ejemplo, jcoverage.

Además existe una herramienta que genera informes, que se llama Cobertura, basado en jcoverage.

→ Organización y diferenciación de los tipos de tests

No todos los tests son iguales, los hay unitarios, de integración, de rendimiento, etc. La automatización de los tests en cada build es algo que se quiere restringir a los tests unitarios, ya que un test de integración, por ejemplo, puede requerir acceso a una base de datos o servidor que puede no estar disponible para todo el mundo en todo momento. Por este motivo, se va a trabajar en esta clasificación de tests y cómo se organizan en los proyectos.

La evolución de gvSIG: Arquitectura

Existen una serie de temas de arquitectura de gvSIG, que el grupo de arquitectura ha ido detectando, sobre los que se plantea una posible actuación actualmente o en un futuro.

Definición de elementos comunes

Dado el número de proyectos y el tamaño que está alcanzando, además del núcleo central de gvSIG, existe gran cantidad de elementos comunes que van apareciendo.

El objetivo consiste en ir identificando estos elementos y extraerlos a proyectos o librerías que puedan ser compartidas por el resto de proyectos gvSIG, evitando la duplicidad de esfuerzos y aprovechando mejoras que se puedan aportar desde los distintos usos de un mismo código.

Aplicación de técnicas de inversión de control o inyección de dependencias

Una de las formas de reducir el acoplamiento en el código es trabajar, siempre que sea posible, con interfaces. Esto,

entre otras cosas, facilita el poder sustituir la implementación de una clase sin afectar al resto de código, así como la realización de tests unitarios.

Existe un punto donde, a pesar de emplear interfaces y referencias a los mismos desde el código, se produce un acoplamiento con la implementación al crear la instancia de un objeto concreto.

Existen diversos patrones de diseño que permiten resolver este problema, como por ejemplo el uso de factorías o del patrón Locator. Sin embargo, aunque se reduce, aún se mantiene un cierto nivel de acoplamiento con dichas factorías y con el Locator.

Otro mecanismo que permite solventar este problema, eliminando el acoplamiento en mayor medida, es el uso del patrón de inversión de control o inyección de dependencias. Éste consiste, básicamente, en cambiar cómo obtiene un objeto las referencias a otros objetos. En vez de ser él mismo quien se encarga de crear u obtener otros objetos, un elemento externo se encarga de asignárselas.

Existen diversos frameworks de inyección de dependencias disponibles actualmente, con un uso cada vez mayor en todo tipo de proyectos. Algunos ejemplos son: Springframework, Guice o PicoContainer.

Sería muy interesante evaluar la posible implantación de uno de estos frameworks, dentro de gvSIG.

Definición de la modularización API-Librería y su uso

Una buena práctica en el desarrollo de aplicaciones consiste en tener una buena separación entre el API y la implementación. Más aún en una aplicación como gvSIG en la que es muy importante la extensibilidad y el poder reemplazar o complementar las distintas funcionalidades de la aplicación. Esto tiene especial interés en las librerías que se desarrollan dentro del proyecto, ya que es donde se implementa la mayor parte de la lógica de negocio de gvSIG.

A nivel de arquitectura de gvSIG, se hará hincapié en hacer que las librerías desarrolladas sean modulares y tengan una definición clara de su API, separada de la implementación, o incluso distintas implementaciones alternativas, como ocurre en algunos casos.

Estrategia para la compatibilidad multidispositivo

Existe un proyecto actualmente cuyo objetivo es desarrollar una versión de gvSIG para dispositivos móviles (PDAs, laptops, etc.), gvSIG Mobile, que se desarrolla en paralelo a la versión actual para escritorio de gvSIG, que llamaremos gvSIG desktop.

Dado que gvSIG está desarrollado empleando el lenguaje de programación Java, que está soportado también en este tipo de dispositivos, la versión Mobile se está desarrollando también con este lenguaje. Sin embargo, se trata de dispositivos con características distintas a las de un ordenador de escritorio o portátil, principalmente con una menor potencia de cálculo y capacidad de memoria y almacenamiento.

Esto significa que gvSIG, tal cual existe para ordenadores personales, no puede ser usado directamente en este tipo de dispositivos. Sin embargo, sí que es posible emplear buena parte del código desarrollado, al menos del núcleo central de lógica geo de gvSIG y del resto de librerías de utilidad.

Además de unificar esfuerzos, esto proporciona una ventaja adicional, que es compartir la arquitectura software base y las buenas prácticas entre las distintas versiones de gvSIG (de escritorio y mobile). Así un desarrollador que tenga experiencia con gvSIG desktop, podrá desarrollar para gvSIG Mobile más fácilmente.

Para conseguir estos objetivos, se está haciendo un esfuerzo adicional, en cuanto a la arquitectura y organización

del desarrollo, que permita compartir el máximo posible entre las dos versiones de gvSIG, con un impacto lo menor posible en el desarrollo de las mismas.

Agradecimientos

En este punto nos gustaría mostrar nuestro agradecimiento a la dirección del proyecto gvSIG, por haber depositado su confianza en nosotros a la hora de realizar la tarea de arquitecto software para el proyecto gvSIG.

También al resto de miembros del grupo de arquitectura y coordinación, con los que trabajamos día a día, y que gracias a sus profundos conocimientos en gvSIG y geomática en general, surgen las principales ideas y necesidades de arquitectura.

Dar las gracias a los distintos proyectos en los que actuamos para coordinación y arquitectura en gvSIG, por dedicarnos su tiempo y su colaboración, así como su participación en muchos de los temas de arquitectura de gvSIG planteados en este artículo.

Y, en general, nuestro agradecimiento para toda la gente que participa en gvSIG, sin cuya ayuda nos sería imposible realizar nuestro trabajo y que hacen que, día a día, este fantástico proyecto sea una realidad.

REFERENCIA

[1] Líneas de código sin tener en cuenta comentarios ni líneas en blanco.

Name	Branch Coverage	Unc
funit.core	58.11% (179/308)	129
funit.core.builders	66.66% (40/60)	20
funit.core.events	100.00% (13/13)	0
funit.core.listeners	0.00% (0/10)	10
funit.core.listeners.events	0.00% (0/9)	9
funit.errors	25.00% (2/8)	6
funit.framework	77.65% (218/280)	62
Assert	45.90% (28/61)	33

> Ejemplo de informe generado con Cobertura.

Internacionalización

Mario Carrera
Responsable internacionalización gvSIG
Conselleria d'Infraestructures i Transport

"La internacionalización es un proceso a través del cual se diseñan productos de software para que puedan adaptarse a diferentes idiomas y regiones sin necesidad de cambios de ingeniería ni cambios en el código" (Wikipedia).

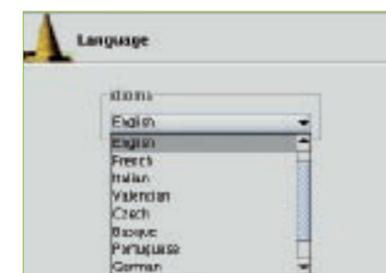
En los últimos años, desde la democratización de Internet, el Software Libre ha crecido considerablemente, permitiendo que las aplicaciones sean accesibles en cualquier parte del mundo. Esto permite que llegue a usuarios de países distintos, que hablan idiomas diferentes.

Gracias a la internacionalización del software es posible que los usuarios puedan trabajar con las aplicaciones en su propio idioma.

En el caso de las aplicaciones SIG, su internacionalización puede ayudar a los nuevos usuarios a introducirse en la materia, siendo más intuitivos los textos si están en su lengua nativa.

Actualmente, la interfaz de gvSIG se encuentra traducida a 13 idiomas en la última versión estable (castellano, valenciano, inglés, alemán, checo, chino, euskera, gallego, francés, italiano, polaco, portugués y rumano), y se van a incorporar dos idiomas nuevos en la nueva versión (ruso y griego).

El proyecto realiza las versiones en idioma castellano, valenciano e inglés. El resto



> Selección de idioma en gvSIG.

de las traducciones han sido realizadas por colaboradores, desde diversas partes del mundo.

Un poco de historia...

La primera versión que se publicó de gvSIG, la 0.2, ya estaba disponible en castellano e inglés, con lo que se conseguía llegar a la mayor parte de los usuarios de SIG.

Posteriormente, en gvSIG 0.3, se amplió la internacionalización de la interfaz con valenciano e italiano, fruto de las traducciones realizadas en la propia Conselleria de Infraestructuras y Transporte de la Generalitat por el Servicio de traducciones de la Subsecretaría y por Andrea Primosich, que se encontraba realizando una beca en el Servicio de Puertos y Costas, respectivamente, y con el francés, que constituía la primera traducción realizada directamente por un usuario externo al proyecto, como fue Erwan Bocher, del Centre National de la Recherche Scientifique de Francia. Desde ésta hasta la actual versión estable 1.1 han ido incorporándose, poco a poco, el resto de idiomas comentados anteriormente. Por otro lado, se ha comenzado la traducción a otros, como son el eslovaco, japonés, vietnamita, turco, danés, groenlandés, tailandés (comenzado recientemente), y swahili entre otros, este último gracias al trabajo realizado por Càrol Puig, de la Universitat Politècnica de Catalunya, dentro de un proyecto de cooperación en Tanzania.

Documentación plurilingüe

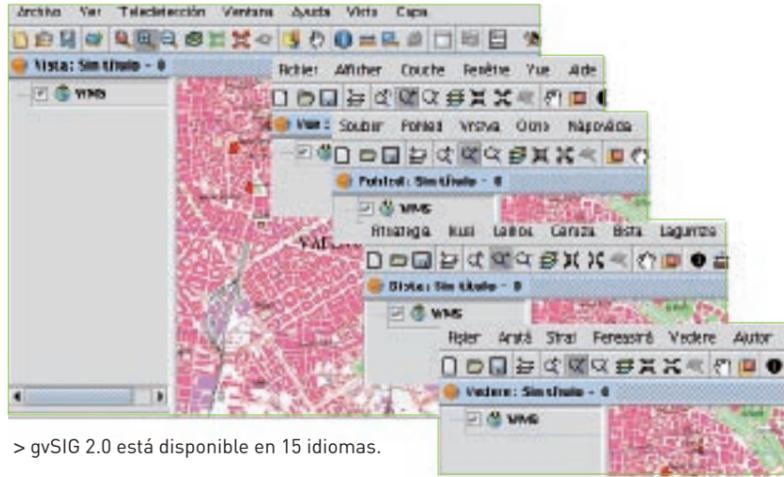
Análogamente a la internacionalización de la interfaz de una aplicación, la traducción de la documentación de una herramienta informática a distintos idiomas permite a un usuario conocer más a fondo la aplicación, sobre todo si se está iniciando en la materia.

En el software abierto, en el que además se dispone del código fuente, una documentación disponible en varios idiomas, o al menos en inglés, facilita el acercamiento de nuevos desarrolladores al proyecto, que pueden desarrollar nuevas extensiones, las cuales pueden llegar a incorporarse a la aplicación.

En el caso de gvSIG, la documentación de usuario ha estado disponible en español desde la versión 0.2, y a partir de la versión 0.3 también lo ha estado en inglés.

Este mismo año se ha hecho público el portal web de la comunidad gvSIG, en www.gvsig.org, en el cual, entre otras cosas, está disponible la documentación de usuario y de desarrolladores del proyecto, y en el que se ha creado un repositorio de extensiones donde todos los interesados pueden publicar sus proyectos relacionados con gvSIG para que estén disponibles para toda la comunidad.

Otra de las novedades de este portal, relacionada directamente con la internacionalización, es la posibilidad de realizar traducciones on line de la documentación. Desde el proyecto gvSIG se ha creado la infraestructura necesaria y se ha hecho



> gvSIG 2.0 está disponible en 15 idiomas.

un llamamiento a la comunidad de usuarios para colaborar en la traducción, tanto de la documentación de usuario como de la de desarrolladores, a distintos idiomas. La respuesta ha sido masiva y se cuenta en la actualidad con cinco grupos de traductores (inglés, italiano, alemán, francés y portugués), cada uno con sus coordinadores y editores, sumando más de 50 traductores en total.

Desde el proyecto gvSIG tenemos gran interés en que las traducciones en inglés tengan una gran calidad, para poder así luego extender gvSIG a otras comunidades lingüísticas, y es por eso por lo que aparte del coordinador y los editores, se ha creado también la figura del supervisor, que revisará las traducciones. La documentación para desarrolladores, que se encuentra únicamente en castellano, se está traduciendo ahora a inglés, gracias a la colaboración de Carlos Galcerán, de Solid Energy New Zealand Ltd., y al grupo de seis traductores entre editores y supervisores. Esto, sin duda, permitirá al resto de grupos trabajar para disponer de una documentación de desarrollo en diversos idiomas.

El italiano fue el primer idioma al que se comenzó a traducir el manual de usuario por colaboradores externos al proyecto,

antes incluso de la creación del portal web. En el mes de febrero de este mismo año, Alessandro Sgambati (Oficina Forestal de Trieste) y Alberto Riva (Dipartimento di Scienze della Terra, Università degli Studi di Ferrara), promovieron la creación de una lista de usuarios de gvSIG en italiano, y ese mismo mes, a través de la propia lista recién creada, se formó un grupo de usuarios dispuestos a traducir la documentación de usuario de gvSIG a italiano. El manual fue traducido completamente hasta la versión 1.1.2, y desde la publicación del nuevo portal web se están publicando los contenidos en él.

Respecto al resto de idiomas, José Antonio Canalejo, consultor SIG residente en Alemania, es el coordinador de las traducciones a idioma alemán, tanto de las cadenas de la interfaz como de la documentación de usuario. Ésta última se está realizando a través del nuevo portal, desde agosto de este mismo año, por un grupo de 13 traductores. En el último trimestre de 2008 también se han comenzado las traducciones de la documentación de usuario a francés y a portugués, gracias a la colaboración de David Martínez Morata y de Eloi Ribeiro, respectivamente, que son los encargados de coordinar las traducciones en estos dos idiomas.

La mejora del sistema de traducción de gvSIG, a través del nuevo portal, va a hacerse extensiva también a las traducciones de las cadenas de la interfaz de la aplicación. Hasta ahora, la traducción se realizaba a través de una aplicación que recibía cada traductor, en la que se cargaba los ficheros de traducción que se le iba facilitando para cada nueva versión o actualización de gvSIG. A partir de ahora se realizará dentro del mismo portal web, en el que para cada versión de gvSIG se actualizará el fichero, y en el que cada traductor trabajará directamente sobre él, indicándose en todo momento cuántas cadenas hay traducidas y cuántas quedan por traducir.

También se está mejorando la actualización de idiomas desde el propio gvSIG, de forma que cuando se cree la traducción de un nuevo idioma no haya que esperar a la siguiente versión que se publique, sino que se pueda actualizar desde la propia interfaz de gvSIG. Este comportamiento será también para la actualización de un idioma ya existente en una versión pública.

Por último, desde el proyecto gvSIG queremos agradecer públicamente la participación de todos los traductores, ya que sin ellos este artículo no hubiera sido posible.

<https://gvSIG.org/web/community/agradecimientos/traductores/view>

En los últimos años, desde la democratización de Internet, el Software Libre ha crecido considerablemente, permitiendo que las aplicaciones sean accesibles desde cualquier parte del mundo. La internacionalización de gvSIG puede ayudar a los nuevos usuarios SIG a introducirse en la materia, siendo más intuitivos los textos si están en su lengua nativa

1 Ignacio Bernardo // Vasco



2 Pavel Dergel // Checo



3 Jose Canalejo // Alemán



4 Tudor Castravet // Rumano



5 Javier Alvarez // Gallego



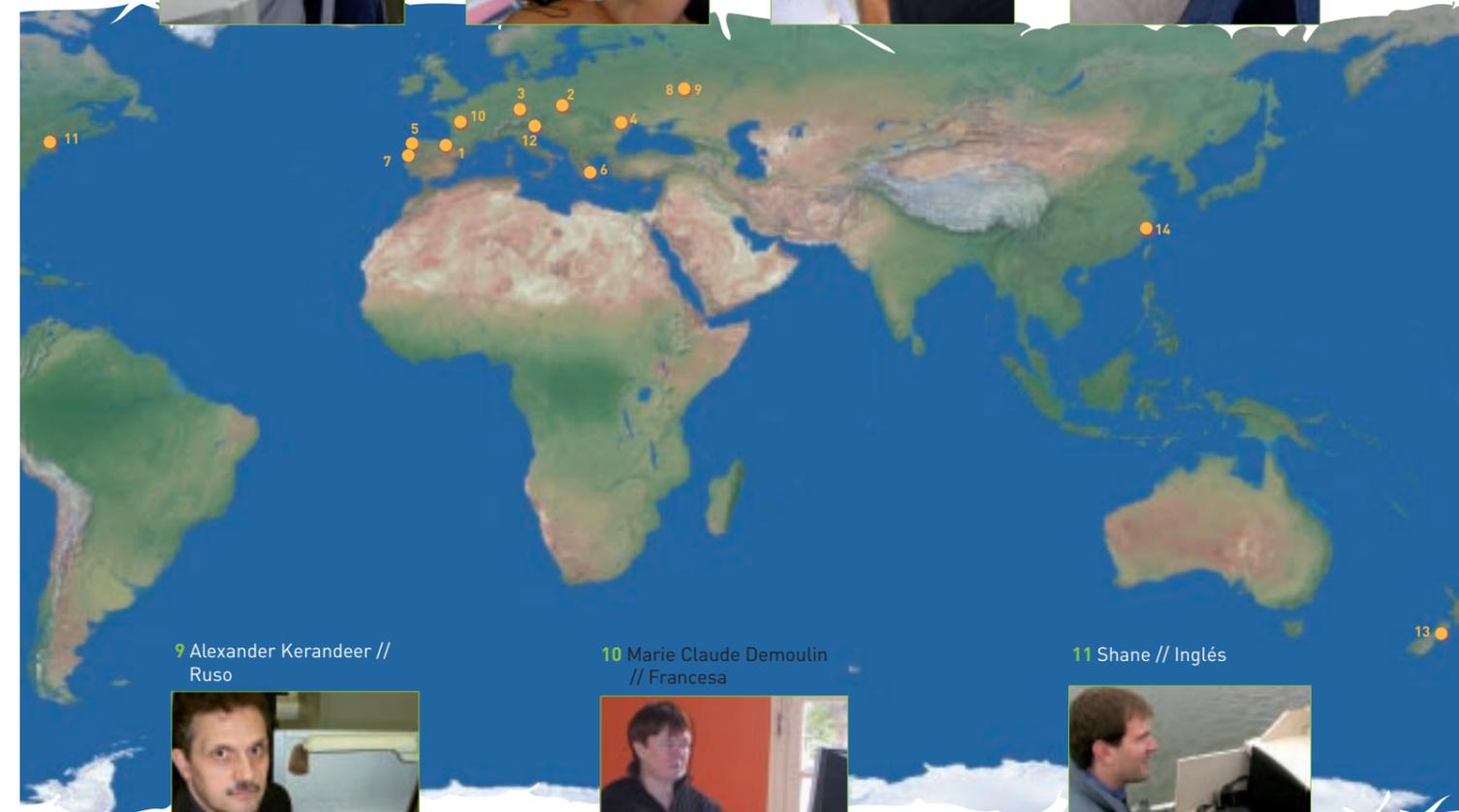
6 Evangelia Kappa // Griega



7 Eloi Ribeiro // Portugués



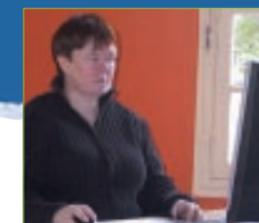
8 Serguei Mikhailov // Ruso



9 Alexander Kerandeer // Ruso



10 Marie Claude Demoulin // Francesa



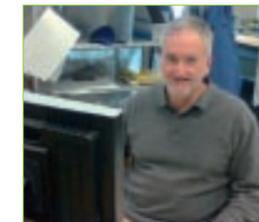
11 Shane // Inglés



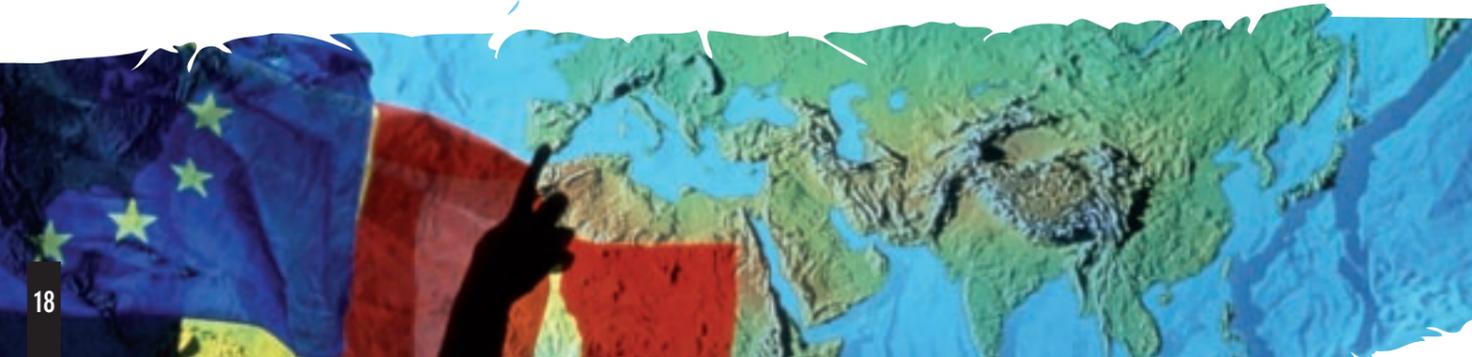
12 Alessandro Sgambati // Italiano



13 Carlos Calceran // Neozelandés



14 Qi Li // Chino



Testeo colaborativo

Manuel Madrid

Responsable calidad y testeo proyecto gvSIG
Asistencia Técnica Externa, Conselleria de Infraestructuras y Transporte de la Generalitat de la Comunidad Valenciana

Victoria S. Agazzi

Coordinadora testeo proyecto gvSIG
Indra Software Labs

El Software Libre se concibió bajo la idea de que entre varios participantes se podía construir algo mejor de lo que cada uno por separado sería capaz de hacer, por lo tanto, todo lo que signifique fomentar la participación va en la dirección de conseguir un producto mejor. Plenamente conscientes de ello, en el proyecto gvSIG se viene trabajando desde hace unos meses en la construcción de una plataforma colaborativa que facilite y maximice la participación de la comunidad gvSIG en todas las áreas posibles. Y una de las áreas en las que pensamos que más nos pueden ayudar nuestros usuarios es en la del testeo. El objetivo: asegurar la calidad del software en base a un modelo de testeo colaborativo.

El presente trabajo expone las diferentes formas en las que los usuarios pueden colaborar dentro del campo del testeo y las herramientas disponibles para ello.

Introducción

gvSIG es un proyecto relativamente joven y en constante evolución. En estos momentos nos encontramos en una fase en la que queremos ceder el protagonismo a la comunidad gvSIG, la cual representa un potencial que hasta el momento no se ha explotado al máximo, tal y como suele hacerse en los proyectos de Software Libre, donde la comunidad desempeña un papel fundamental. Y es que una de las principales diferencias entre el Software Libre y el propietario es precisamente la participación. El objetivo más importante del proyecto gvSIG, en estos momentos, es dotarnos de la infraestructura necesaria para que nuestros usuarios y desarrolladores participen en masa y consigamos construir, entre todos, un producto de calidad. Uno de los aspectos más relacionados con la calidad es el testeo. En gvSIG hemos trabajado, junto con la empresa Dimensión Informática (actualmente Indra), la cual cuenta con una gran experiencia en el campo del testeo de software, en el diseño de un modelo de testeo colaborativo basado, principalmente, en la participación de los usuarios y cuyo objetivo es hacer de

gvSIG un software lo más fiable posible, un software de calidad. El presente artículo describe brevemente los posibles tipos de colaboración en los que los usuarios y desarrolladores de gvSIG pueden participar con respecto al testeo, y la infraestructura colaborativa en la que se está trabajando para facilitar al máximo esta labor.

Tipos de testeo en función del ciclo de vida de una versión

En función de los tipos de ejecutables a testear y de la fase dentro del ciclo de vida de una versión en la que se realice, podemos diferenciar tres tipos de testeo. Por orden cronológico de sus fases son: testeo de pre-estabilización, testeo de estabilización y testeo de post-publicación. A continuación se describen las fases y el tipo de testeo asociado a cada fase.

Testeo en fase de pre-estabilización

La fase de pre-estabilización (también llamada fase de desarrollo) es aquella a lo largo de la cual se implementan los requerimientos planificados para una determinada versión. Los ejecutables que se testean en esta fase son sucesivas distribuciones (builds) inestables que se liberan de manera periódica.

Lo más interesante de esta fase es poder conocer la opinión de los usuarios

sobre las nuevas funcionalidades, cuando aún se está a tiempo de rectificar. Las indicaciones de los usuarios podrían significar desde una pequeña modificación de un requerimiento hasta la inclusión de uno nuevo, previo estudio y aprobación por parte de los responsables. En esta fase son más importantes las sugerencias de mejora que la detección de errores, puesto que al tratarse de una fase muy prematura los errores son naturales. El cauce establecido para que los usuarios interesados en este tipo de testeo nos transmitan sus sugerencias será un formulario situado en la web de la comunidad y que posteriormente también estará disponible en el propio gvSIG Desktop (ver apartado "Formulario de comunicación de errores y sugerencias").

Con este objetivo, recientemente se han comenzado a publicar las distribuciones de gvSIG en fase de desarrollo en la web colaborativa del proyecto:

<https://gvSIG.org/web/projects/gvsig-desktop/devel>

Existe otro tipo de colaboración asociado a esta fase de pre-estabilización que consiste en participar en la definición de los planes de pruebas de las nuevas funcionalidades (ver apartado "Gestor de planes de pruebas").

Conforme se van implementando las nuevas funcionalidades, se deberán diseñar las pruebas que luego se pasarán en la fase de estabilización. Ello se hace siguiendo unos criterios definidos y bajo la dirección de un coordinador con el que, en este caso, se trabaja en estrecha colaboración. Se ha decidido desarrollar un plan de pruebas piloto que cubra las funcionalidades que la versión 1.x de gvSIG tiene ya implementadas. Este trabajo se está llevando a cabo desde la propia Conselleria de Infraestructuras y Transporte, con el objetivo de generar la documentación necesaria para que la comunidad pueda comenzar a colaborar. En la siguiente URL se puede consultar la información relativa al plan de pruebas piloto que se está desarrollando actualmente:

<https://gvSIG.org/web/production/docs/procedures/procedimientos/testing/introduccion/tipos-de-testeo-2/por-metodologia-de-testeo/testeo-mediante-planes-de-prueba/>

Por último conviene remarcar que los builds en desarrollo son altamente inestables, con lo que se recomienda encarecidamente no utilizarlos para un uso normal o en entornos de producción. Incluso podrían introducir algún tipo de inestabilidad en el sistema operativo sobre el que se ejecute.

Testeo en fase de estabilización

La fase de estabilización es aquella que comienza una vez que se supera la fase de desarrollo y todos los requerimientos de la aplicación se dan por implementados. Se trata de la fase inmediatamente anterior a la publicación y en la que, de manera intensiva, se somete a la aplicación a una serie de pruebas con el objetivo de detectar y solucionar el máximo número de errores, antes que la versión llegue a manos de los usuarios. Los ejecutables sobre los que se testea en esta fase son, al igual que en la fase anterior, sucesivas distribuciones (builds) inestables que se liberan de forma periódica.

El testeo en esta fase consistirá en ejecutar planes de test gestionados mediante una aplicación que permite desde el diseño hasta la ejecución de las pruebas. Al tester colaborador se le asignarán una serie de tests para probar aquellas funcionalidades con las que esté más familiarizado o en las que tenga un interés especial, los cuales ejecutará mediante

el gestor mencionado al que se accede vía web. Los errores derivados de la ejecución de los tests se reportarán, previa comprobación de que no ha sido reportado ya por otro tester (ver apartado "Seguimiento de incidencias"), a través del formulario mencionado en el apartado anterior y que estará disponible tanto en la web de la comunidad gvSIG, como en la propia aplicación (ver apartado "Formulario de comunicación de errores y sugerencias"). Por último se encargará de comprobar si el error ha sido solucionado o no en la siguiente distribución o build de la aplicación.

Este proceso se haría de forma cíclica para cada nueva distribución, hasta que los responsables determinen que la aplicación ha alcanzado un nivel de estabilidad apropiado para su publicación.

El objetivo es dotarlos de la infraestructura necesaria para que nuestros usuarios y desarrolladores participen en masa y consigamos construir, entre todos un producto de calidad

Ésta es quizá la forma más comprometida (y más importante) de colaborar, pero siempre desde la voluntariedad. El colaborador es, en todo momento, el que marca la dedicación en función de su disponibilidad.

También será posible, como en las demás fases del ciclo de vida de la aplicación, reportar errores no provenientes del plan de pruebas de gvSIG que hayan sido identificados y acotados por un usuario de la comunidad. Como se ha indicado al principio de este apartado, en esta fase los requerimientos están ya cerrados por lo que las sugerencias de mejora se valorarán en general de cara a las siguientes versiones de la aplicación.

Testeo en fase de post-publicación

El testeo en fase de post-publicación consiste en testear la última versión estable de gvSIG publicada. Esta fase se prolonga de forma indefinida en el tiempo, ya que siempre hay una última versión estable publicada susceptible de ser testeada.

El testeo en esta fase consiste en utilizar de forma normal la aplicación hasta que se detecte un error.

Lo que el usuario tiene que hacer a continuación es intentar reproducir el error con

el fin de acotar el origen del problema. Una vez acotado, el usuario debe comprobar que no se trata de un error detectado con anterioridad, para lo cual se deberá consultar la lista de errores desde la web de la comunidad gvSIG (ver apartado "Seguimiento de incidencias"). Es importante evitar la duplicación de incidencias (errores o sugerencias).

Una vez comprobado que el error no forma parte de la lista de errores detectados, debe documentarlo 'mínimamente' rellenando el formulario mencionado en los dos apartados anteriores (ver apartado "Formulario de comunicación de errores y sugerencias"). A partir de ahí podrá hacer un seguimiento de la evolución del posible arreglo accediendo al boletín de la incidencia en cuestión.

Lo interesante de este tipo de testeo es que, mientras que la batería de pruebas a la que se somete la aplicación durante la fase de estabilización cubre un número determinado de casos de uso, los casos de uso reales que nos aportan los usuarios son infinitos. De esta manera se prueban cosas que de otra manera no se probarían nunca. Por otro lado se amplía de forma espectacular el juego de datos utilizado en las pruebas ya que los usuarios nos aportan datos de todo tipo, origen y formato.

Por último, quizá la forma más sencilla de colaborar con el testeo del proyecto posiblemente sea el simple hecho de expresar la opinión que el usuario de gvSIG tiene sobre la aplicación a través de la lista de usuarios del proyecto, como se viene haciendo desde siempre. Hacer llegar a los responsables aquellos aspectos que el usuario identifica como mejorables desde su punto de vista.

Conocer la opinión de nuestros usuarios es fundamental de cara a obtener un producto usable y de calidad. Al igual que en el caso de los errores, las sugerencias de mejora se podrán comunicar de forma sencilla mediante el formulario antes mencionado.

Lógicamente en esta fase los requerimientos de la versión también están cerrados por lo que las sugerencias de mejora se estudiarán de cara a las próximas versiones de la aplicación. Lo mismo ocurre, en este caso, con los errores detectados, salvo que se trate de un error muy importante, en cuyo caso se estudiaría su arreglo. En caso de abordar el arreglo, se publicaría una versión de actualización (comúnmente conocida como parche).

Herramientas para el testeo colaborativo

Registro de colaboradores

En la fase de estabilización de una versión de gvSIG, como se ha comentado en el apartado anterior, los equipos de desarrollo y testeo trabajan de manera intensiva y coordinada en la detección, documentación y arreglo de errores, con el objetivo de que la mayoría de estos no acaben afectando al usuario final. Para poder realizar una correcta planificación de esta fase es necesario contar con cierta información sobre los testers, como por ejemplo qué tipo de funcionalidades de gvSIG estarían dispuestos a probar o cuánto tiempo estiman que podrían dedicar a ello. Por ello éste es el único tipo de colaboración que, de alguna manera, se formaliza mediante un simbólico compromiso. En caso de que el usuario de gvSIG quiera ser tester colaborador, se le solicitarán algunos datos de contacto, además de la configuración de su entorno de trabajo y las funcionalidades de gvSIG que desea testear.

Posteriormente, en el momento en que estemos en disposición de abordar una fase de estabilización, nos pondremos en contacto con los testers colaboradores para preguntarles sobre su disponibilidad durante el tiempo que estimamos durará la estabilización. Por supuesto, y dado que es una actividad voluntaria, contamos con que la cantidad de tiempo a dedicar puede variar mucho de unos testers a otros o incluso ser nula en un momento dado.

Gestor de planes de pruebas

Una de las herramientas más importantes dentro de la fase de estabilización es el gestor de planes de pruebas, el cual permite, por un lado, definir la batería de pruebas a la que se va a someter a la aplicación, y por otro, ejecutar estas pruebas de forma ordenada y coordinada. Al mismo tiempo permitirá monitorizar el nivel de fiabilidad de la aplicación.

Cada prueba a ejecutar tendrá asociado una serie de pasos y un conjunto de datos definidos en el gestor de planes de prueba. Además se tendrá definido el comportamiento esperado de la aplicación como resultado de la ejecución de cada prueba. En función de este resultado, la prueba se evaluará de forma positiva o negativa. En caso de que el tester colaborador detecte un fallo en gvSIG durante la ejecución de una prueba, deberá reportarlo como inci-

dencia mediante el formulario correspondiente (ver apartado siguiente). De esta forma, gracias a los resultados del conjunto de pruebas se podrá monitorizar la calidad de la versión de gvSIG susceptible de ser publicada.

La herramienta que hemos seleccionado en este caso es Salomé-TMF, que es una aplicación en Software Libre cuya distribución se hace bajo licencia GNU/GPL. Para el correcto uso de Salomé-TMF dentro del proyecto gvSIG, se ha decidido implementar algunas mejoras que simplificarán la ejecución de las pruebas. Estos cambios se harán de forma coordinada con el equipo de desarrollo de Salomé-TMF para que reviertan a todos los usuarios del proyecto.

Formulario de comunicación de errores y sugerencias

Comunicar un error es una tarea que podrá realizar cualquier usuario a través de dos vías: desde web de la comunidad gvSIG y a través de la propia aplicación. Antes de reportar un error es importante comprobar que éste no ha sido ya reportado, razón por la cual se recomendará hacer una búsqueda en función de palabras clave que identifiquen la incidencia. Será posible ejecutar la búsqueda desde el propio formulario de comunicación de errores y sugerencias. En caso de ser un error "nuevo", el usuario deberá registrarlo rellenando el formulario con la información mínima necesaria para que la incidencia pueda ser reproducida por el desarrollador encargado de corregir el error.

Seguimiento de incidencias

El usuario de gvSIG tendrá la capacidad de hacer un seguimiento de aquellas incidencias en las que esté interesado, las haya reportado él mismo o no. Además de la posibilidad de hacer búsquedas contra la base de datos de las incidencias, existen búsquedas predefinidas para los errores corregidos en las distribuciones en desarrollo que gvSIG libera periódicamente.

Conclusiones

Uno de los principales objetivos dentro del proyecto gvSIG en la actualidad es promover la participación de nuestra co-

munidad de usuarios y desarrolladores. Para ello es necesario, en primer lugar, identificar las distintas modalidades de colaboración, de manera que los potenciales colaboradores las conozcan y puedan elegir la más adecuada a sus circunstancias.

En segundo lugar, debe existir una infraestructura con las herramientas necesarias para que la colaboración se lleve a cabo de la manera más cómoda y eficaz posible y con las instrucciones o procedimientos necesarios para ello debidamente documentados. En este momento esta infraestructura se encuentra en plena construcción.

En tercer y último lugar, para que haya colaboración ha de existir motivación. ¿Por qué deberían animarse a colaborar con el proyecto los usuarios y desarrolladores de gvSIG? Básicamente por dos razones. La primera, tal vez la necesidad de devolver, de alguna manera, parte de lo que antes has recibido sin coste (el software, la documentación, el soporte, etc.). La segunda, que es probablemente la más extendida, el beneficio propio y a la vez mutuo. Participar en un proyecto de Software Libre es una estúpida oportunidad de modificar o mejorar la herramienta en tu propio beneficio. Pero lo realmente interesante es que la aportación de uno repercute en los demás y las de los demás en uno mismo, de manera que el resultado final es la suma de todas las aportaciones en una misma dirección.

Cualquiera de los tipos y niveles de colaboración identificados en el presente artículo redundan en aumentar la calidad de gvSIG, lo que definitivamente beneficia al usuario. En la medida en que los usuarios de gvSIG entiendan que participar en el testeo les beneficia, conseguiremos contar con una significativa comunidad de testers y, por tanto, con una aplicación usable y de calidad.



> Salomé-TMF: aplicación web desde la que se gestionan los planes de pruebas.

Documentación

Víctor Acevedo Royer
Responsable coordinación áreas de desarrollo y documentación gvSIG
IVER

Antecedentes

Antes de comenzar con el proyecto de documentación de gvSIG, el manual se generaba a partir de un único documento de OpenOffice.org (el manual de usuario completo en aquel entonces no tenía 50 páginas). El problema de este método de trabajo es que hacía muy difícil controlar un equipo que trabajase simultáneamente sobre ese documento y, por otro lado, teníamos un único requisito que cumplir, el documento final tenía que tener el mismo aspecto que el que tenía por aquel entonces.

Adoptada la decisión de dotar de recursos a la documentación del proyecto, teníamos que plantearnos cómo lo íbamos a abordar. La primera decisión fue hacia dónde nos dirigíamos primero, atendíamos las necesidades de los usuarios documentando el uso de la aplicación o centrábamos nuestros esfuerzos en generar documentación que sirviese a la comunidad de desarrolladores.

En ese momento decidimos que lo que más nos urgía era la documentación de usuario, teníamos funcionalidades, teníamos usuarios, pero carecíamos de una guía para iniciarse en el uso de la aplicación y comenzamos a trabajar en las necesidades de documentación en este área.

Los comienzos siempre son duros

El proyecto de documentación de gvSIG se inicia como tal para el TECNIMAP del 2006. En esas fechas teníamos la intención de presentar las funcionalidades de edición de gvSIG y, puestos a comenzar por algún sitio, porqué no liberar las nuevas funcionalidades con un manual de usuario que las describiese y explicase su funcionamiento.

El problema, como en casi todo proyecto, eran los plazos en los que teníamos que ejecutar la tarea, con el añadido de que, paralelamente, teníamos que definir cómo se iba a realizar y seleccionar las herramientas con las que se confeccionaría y gestionaría el proyecto, siempre pensando que teníamos que obtener el mismo formato de salida con el mismo

aspecto y trabajando sobre prototipos de las funcionalidades, lo que implicaba que muchas veces o la interfaz de usuario no estaba totalmente acabada o no era estable la funcionalidad, por lo que no se podía realizar todo el proceso porque en algún punto fallaba.

Fue una época en la que muchos días teníamos que deshacer el trabajo que habíamos realizado y volver a empezar.

Cómo trabajábamos

La primera decisión que tomamos fue la de dividir el manual en distintos documentos agrupados por área funcional, de esta manera conseguimos que varias personas trabajasen paralelamente sin necesidad de trabajar todos a la vez en el mismo archivo y contar con un control de versiones de estos documentos.

Otro de los puntos en los que hacemos más hincapié es en el tratamiento de imágenes. Por regla general las capturas que se realizan del programa son con una paleta de colores de 24 bits. Al principio esto nos suponía un problema porque en el paso de la documentación a formato .PDF nos producía manuales con un tamaño en disco inaceptable, ahora el problema es que las páginas on line tardan en cargarse, por lo que tratamos la imagen reduciendo la paleta de colores, normalmente con 32 basta para que la imagen se vea correctamente, pero conservamos la imagen original a color verdadero por si queremos volver a utilizarla en otro tipo de documento y tratarla sin perder la calidad original.

Los retos del presente

Actualmente, dentro del proyecto de documentación de gvSIG el reto más grande que tenemos son las traducciones de los contenidos a los diferentes idiomas que solicita la comunidad. Estas traducciones se realizan actualmente por equipos de colaboradores y voluntarios que trabajan on line directamente sobre los documentos. Para nosotros ha supuesto un reto enorme facilitar los mecanismos necesarios para poder realizar esta labor, tanto de infraestructura como de

coordinación. El resultado es altamente satisfactorio, ahora mismo hay tres equipos trabajando en la traducción de la documentación de usuario a alemán, francés y al italiano, y esperamos que en breve podamos tener un cuarto equipo que trabaje en la traducción al inglés de la documentación de desarrolladores.

La respuesta que hemos recibido por parte de la comunidad y el apoyo no sólo con palabras sino con trabajo nos ha sorprendido muchísimo, en algunos momentos incluso nos hemos visto desbordados por la cantidad de gente que quiere participar en el proyecto de una u otra forma.

El futuro

De cara al futuro nos gustaría hacer la web del proyecto aún más abierta a la comunidad..., que la propia comunidad tenga un espacio donde poder dejar sus escritos relacionados con gvSIG, trucos, experiencias, peticiones, cualquier cosa que sea de interés. En definitiva, queremos ser un lugar de referencia no sólo para usuarios de gvSIG en particular, sino del mundo de los SIG en general.

Por último

No me resisto a dejar pasar esta oportunidad de agradecer a todos los que, de una manera u otra, han participado en la evolución de este proyecto de documentación gvSIG. Tanto a los miembros de las empresas participantes, sobre todo a los desarrolladores por la paciencia y el esfuerzo de 'aguantar' a documentadores y testers, más aún en periodos de liberación en los que todos estamos más nerviosos, a todos los documentadores por el gran trabajo realizado en una de las partes más oscuras de cualquier gran proyecto informático, a las empresas que han participado en la traducción de partes de la documentación, no sólo por las traducciones sino también, y especialmente, por su implicación en la corrección de los textos y sus sugerencias, y por encima de todo a todos los voluntarios que se han volcado en este proyecto.

Muchísimas gracias a todos.



Tour 2008

Desde sus comienzos gvSIG ha estado presente en eventos nacionales e internacionales, con el objetivo de dar a conocer el proyecto, contribuir en la difusión de las tecnologías y datos libres, establecer contactos con la comunidad de usuarios y desarrolladores de SIG libre y recibir información directa sobre la visión del proyecto desde el exterior.

En 2008 gvSIG ha formado parte en diversos congresos, algunos de visita obligada todos los años, como las JIDEE (Jornadas de la IDE de España) o las Jornadas de SIG Libre de Girona, y otros nuevos que nos llenan de ilusión, como el Primer encuentro de usuarios de gvSIG de Italia. Observamos con entusiasmo que son ya muchos los eventos en los que gvSIG tiene presencia, incluido en ponencias sobre desarrollos o proyectos

que está realizando la comunidad, ya sean empresas, administraciones o universidades.

Además de los congresos y jornadas aquí citados, gvSIG ha impartido diversos seminarios y talleres en el ámbito universitario: Universidad Politécnica de Valencia, Universidad de Valladolid, cursos de verano de la Universidad de Extremadura, Máster de Universidad Politécnica de Madrid/LatinGEO, Máster de Sostenibilidad de la Universidad Politécnica de Cataluña...

Este Tour 2008 no estaría completo sin citar las visitas que hemos recibido de entidades de otros países interesadas por el proyecto, como la reunión celebrada en las dependencias del IGN con la Asociación de GIS de Rusia (GIS-Association Russian Federation) o el encuentro con el Gobierno de Corea del Sur.

Fuera de España

Primer Simposio GIS Educación en América Latina



Lugar	Quito [Ecuador].
Fecha	Del 26 al 28 de marzo.
Descripción	Reunión de expertos internacionales en aplicación de la geoinformación y SIG al aprendizaje.
gvSIG	→ Ponencia "gvSIG como herramienta de educación".

INTERGEO



Lugar	Bremen [Alemania].
Fecha	Del 30 de septiembre al 2 de octubre.
Descripción	Evento de geodesia, geomática y topografía más importante de Europa.
gvSIG	→ Espacio en Open Source Park. → Ponencia "gvSIG - Aktuelles im Überblick". → Ponencia "Publikation OGC. Metadateneditor mit gvSIG".

FOSSGIS 2008



Lugar	Friburgo [Suiza].
Fecha	Del 1 al 3 de abril.
Descripción	Congreso de SIG libre de los países de habla germana.
gvSIG	→ Ponencia "gvSIG - ein Desktop GIS für Verwaltungen, IT-Firmen und Universitäten". → Taller "Advanced geographical analysis using gvSIG-SEXTANTE".

FOSS4G 2008



Lugar	Ciudad del Cabo [Sudáfrica].
Fecha	Del 29 de septiembre al 4 de octubre.
Descripción	Congreso anual de OSGeo sobre Software Geomático libre y abierto (Free and Open Source Software for Geospatial).
gvSIG	→ Ponencia "gvSIG: Open Source for handling Geospatial Information Status Report. Towards an Open Organization". → Ponencia "gvSIG Mobile: How to code for desktop and mobile GIS/SDI". → Ponencia "Improving open source GIS-SDI integration: the web service publishing extension for gvSIG". → Taller "Introduction to gvSIG/SEXTANTE as desktop GIS with advanced spatial analysis and SDI publishing tools".

Latinoware '08



Lugar	Foz de Iguazú [Brasil].
Fecha	Del 30 de octubre al 1 de noviembre.
Descripción	Evento anual latinoamericano de Software Libre más destacado.
gvSIG	→ Ponencia "gvSIG como herramienta de educación". → Taller "gvSIG Desktop".

Primo incontro degli utenti italiani di gvSIG



Lugar	Trieste [Italia].
Fecha	Del 9 al 10 de mayo.
Descripción	Primer encuentro de usuarios de gvSIG en Italia.
gvSIG	→ Seminario "gvSIG, uno strumento valido per gli specialisti GIS". → Seminario "gvSIG Mobile 0.1, un prototipo di Mobile GIS".

1er Encuentro Geomática Libre Venezuela

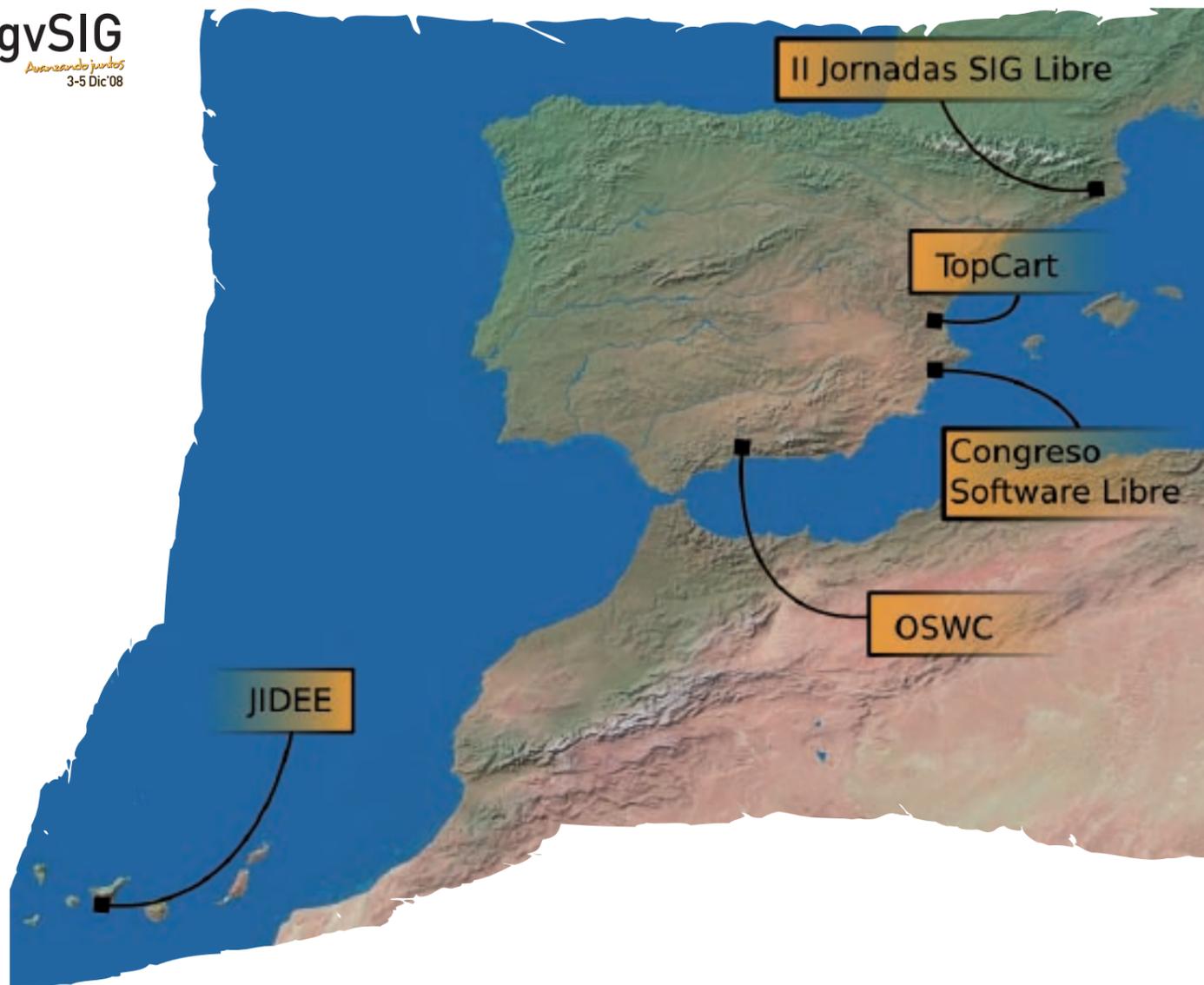


Lugar	Mérida [Venezuela].
Fecha	10 de julio.
Descripción	Primer encuentro de geomática libre de la comunidad venezolana, en el marco de las II Jornadas de Reflexión, Investigación y Desarrollo de Tecnologías Libres.
gvSIG	→ Ponencia "gvSIG, SIG y cliente IDE libre".

AGIT 2008



Lugar	Salzburgo [Austria].
Fecha	Del 2 al 4 de julio.
Descripción	Simposio de geomática aplicada.
gvSIG	→ Ponencia "gvSIG: Aktuelles im Überblick". → Ponencia "gvSIG Mobile".



Dentro de España

TOP-CART 2008



Lugar	Valencia.
Fecha	Del 18 al 22 de febrero.
Descripción	Congreso Internacional sobre Ingeniería Geomática y Topográfica.
gvSIG	→ Ponencia "gvSIG: el Sistema de Información Geográfica en Software Libre".

XXI IGSM 2008



Lugar	Valencia.
Fecha	Del 4 al 10 de mayo.
Descripción	XXI Encuentro Internacional de Estudiantes de Geodesia (International Geodetic Students Meeting).
gvSIG	→ Ponencia "gvSIG a real tool for GIS technicians".

JIDEE



Lugar	Tenerife.
Fecha	Del 5 al 7 de noviembre.
Descripción	Jornadas Técnicas de la Infraestructura de Datos Espaciales de España.
gvSIG	→ Ponencia "gvSIG Mobile como cliente móvil de IDE".

III Congreso Software Libre Comunitat Valenciana



Lugar	Alicante.
Fecha	Del 5 al 7 de noviembre.
Descripción	Congreso de Software Libre en la Tecnologías de la Información y Comunicación (TIC), especialmente en el mundo de la educación.
gvSIG	→ Taller "EduSIG: gvSIG como herramienta de enseñanza de la geografía". → Ponencia "EduSIG: gvSIG aplicado a la educación".

OSWC



Lugar	Málaga.
Fecha	Del 20 al 22 de octubre.
Descripción	La Conferencia Internacional de Software Libre es el evento más importante relacionado con tecnologías de fuente abierta de Europa.
gvSIG	→ Ponencia "gvSIG-Infraestructuras de Datos Espaciales". → Ponencia "gvSIG Mobile: gvSIG en dispositivos móviles".

II Jornadas SIG Libre II Jornadas de SIG Libre

Lugar	Girona.
Fecha	Del 3 al 5 de marzo.
Descripción	Jornadas dedicadas a la programación libre en el campo de los Sistemas de Información Geográfica.
gvSIG	→ Ponencia "gvSIG: Informe de estado". → Ponencia "gvSIG Mobile. Cliente GIS/IDE móvil". → Taller "gvSIG y Sextante". → Mesa de debate "Software Libre y Proyectos de Cooperación".

Comunidad gvSIG: Universidades

Entrevista

Antoni Pérez

Estudis d'Informàtica, Multimèdia i Telecomunicació
Universitat Oberta de Catalunya



¿Cómo ves el panorama actual de implantación de soluciones de Software Libre en la Universidad? ¿Está entrando gvSIG en ambientes universitarios?

Antoni Pérez. En cuanto a la penetración del software en la Universidad habría que distinguir dos tipos de software: herramientas utilizadas para llevar a cabo el trabajo del día a día, como procesadores de texto u hojas de cálculo; y herramientas curriculares de las asignaturas, al estilo de Matlab, Scilab, Mathematica, Octave, Pspice, Cucs, etc.

Personas con cargos institucionales, profesores y estudiantes son los actores que juegan algún papel en la Universidad y cada uno tiene un efecto distinto sobre la extensión del Software Libre. Así, las personas con cargos institucionales pueden marcar una política dirigida a utilizar software propietario o libre. En este sentido, en los últimos años se ha detectado un tímido impulso al Software Libre en la Universidad. Tímido porque si bien desde los órganos directivos se apuesta abiertamente por el Software Libre, a la hora de la verdad en los campus se encuentran tanto Software Libre como propietario y, además, muchas aplicaciones funcionan sólo con software propietario.

Un ejemplo que cabe destacar es el de la Universitat Oberta de Catalunya (UOC), que desde 2006 ofrece a los estudiantes un punto de trabajo en open source. Así, los estudiantes recibían en casa, junto con el material de los estudios, un LiveDVD con una distribución basada en Ubuntu que contenía el software necesario para los estudios. Esta opción les permitía probar el funcionamiento del campus con el punto de Software Libre y, eventualmente, instalarlo. El DVD contenía también las

alternativas libres a varias opciones de software propietario.

El DVD se ofreció en un principio sólo a los estudiantes de las titulaciones tecnológicas (Ingeniería Técnica en Informática de Gestión, Ingeniería Técnica en Informática de Sistemas, Ingeniería Informática, Ingeniería Técnica de Telecomunicación, especialidad telemática), unos 7.000. Sin embargo, en el curso 2007 ya se suministró a todos los estudiantes de la UOC, unos 34.000. El resultado fue que, en un semestre, el número de peticiones al servidor provenientes de Firefox pasó del 23 por ciento al 31 por ciento, y el número de peticiones provenientes de Linux pasó del 4 al 15 por ciento.

Por otro lado, a nivel interno, la UOC inició en 2007 un proyecto parecido para que todos los profesores pudieran contar con un punto de trabajo en Software Libre. Se inició un proyecto piloto pero, en este caso, el gran número de aplicaciones propias de la UOC, desarrolladas sobre

“En cuanto a la implantación del Software Libre en la Universidad, en un primer momento prima la cuestión ética, desde tres puntos de vista: político, económico y filosófico”

software propietario, ha llevado a que la iniciativa no haya tenido el mismo éxito que con los estudiantes.

Los profesores, en algunos casos, pueden elegir utilizar Software Libre como herramientas curriculares en sus asignaturas y así promocionarlas. Es decir, en ciertas ocasiones hay alternativas abiertas que pueden suplir las necesidades docentes del curso con toda garantía. Sin embargo, a menudo estas alternativas no existen: un caso claro es, precisamente, el de las herramientas CAD propietarias que están tan implantadas en el mercado.

Por otro lado, a pesar de la implantación de Software Libre en muchos ámbitos, y cada día más, aún quedan nichos de mercado exclusivos del software propietario o en los cuales cierto software propietario tiene mucho prestigio. En estos casos,

los profesores acostumbran a escoger estas opciones que se ajustan de una forma más clara a las necesidades del mercado.

Sin embargo, en asignaturas en las que interesa estudiar intrínsecamente el software, lo que prima es el Software Libre, precisamente por la libertad que proporciona.

En cuanto a la faceta de investigación de los profesores, parece que el software de código abierto está algo más extendido, pero en este aspecto depende mucho del ámbito en que se encuentre y de decisiones personales.

Los estudiantes a menudo tienen varias opciones ante sí, libres o propietarias, y pueden escoger unas u otras, en general en lo que respecta al software del día a día. Su apuesta acostumbra a ser por el software propietario, que es “el que suena”.

Se ha visto anteriormente que en un semestre, la utilización de Linux pasó del 4 al 15 por ciento. De estos datos hay dos elementos destacables: el primero, la poca penetración de Linux entre los estudiantes y, el segundo, el efecto que una acción institucional puede tener en la difusión del Software Libre en este colectivo. Hay además un tercer efecto, y es la implicación de algunos estudiantes en la búsqueda de alternativas libres a software propietario. Es decir, en algunas asignaturas donde parecía no haber alternativa libre a la herramienta curricular que se utilizaba, han sido los mismos estudiantes los que, por iniciativa propia, se han ofrecido a buscar y probar alternativas, con el beneplácito y complicidad del profesor.

Por tanto, la conclusión a la que se llega es que el Software Libre está empezando a penetrar en la Universidad. Sin embargo, es necesario que haya acciones institucionales y una actitud proactiva por parte de los profesores, no sólo para promocionarlo, sino incluso para darlo a conocer.

Hasta aquí la implantación del open source en general. En cuanto a gvSIG, en particular, se trata de un software sectorial en un ámbito que, desgraciadamente, aún no tiene la penetración que sería

Lluís Vicents

Miembro del equipo de proyectos SIG del Servicio de Sistemas de Información Geográfica y Teledetección (SIGTE)

Universitat de Girona

Miembro del equipo docente del Máster UNIGIS

deseable en los currículos universitarios: el de los Sistemas de Información Geográfica (SIG). Además, gvSIG tiene que competir con los excelentes programas de colaboración con universidades que ofrecen ESRI o INTERGRAPH, además de disponer ambas compañías de muy buenos productos, con muchos años de experiencia y con muchas capacidades.

Aún así, respecto a departamentos de investigación universitaria dedicados a desarrollos SIG y que ya trabajaban con Software Libre, la penetración de gvSIG parece ser excelente. Así, es destacable su penetración en algunos centros universitarios de prestigio en España como las universidades Jaume I de Castellón, de Girona, de Extremadura o la propia UOC. En el caso de la Jaume I, por ejemplo, gvSIG forma parte estructural de los estudios de máster, a pesar de que en la Ingeniería Informática, por ejemplo, su penetración es menor, debido sobre todo al coste de migración de las asignaturas que ya usaban otros software. En cuanto a la UOC, como se verá más adelante, existe una línea de trabajos y proyectos finales de carrera con gvSIG. En cualquier caso, a pesar de estos ejemplos, no queda claro si esta apuesta es coyuntural o estructural.

Lluís Vicents. Sería fantástico poder responder que sí, que efectivamente las soluciones open source están a la orden del día en todas las universidades, pero tristemente, y desde nuestra experiencia y conocimiento personal, debemos decir que aún queda mucho camino por recorrer. Basta con pasearse por las aulas de Informática de muchas de las facultades y universidades para comprobar que el sistema operativo GNU/Linux (con distribuciones para todos los gustos) es, cuanto menos, una rara avis, mientras que el sistema predominante continúa siendo propiedad de Microsoft. Al menos, ésta es la realidad, creemos, en cuanto a sistemas operativos. En el campo de las aplicaciones o programas libres, la realidad es algo diferente, mucho más optimista, ya que si bien es cierto que las aplicaciones privadas continúan siendo (en la mayoría de los

casos) dominantes, no es menos cierto que el auge e implantación de soluciones libres en las universidades es cuanto menos, notable, con algunos ejemplos que son paradigmáticos como el paquete de ofimática OpenOffice.org, o el explorador web Firefox, el gestor de correo Thunderbird o el Calendario Sunbird.

Muchos de estos paquetes básicos utilizados en las tareas que diariamente se realizan en cualquier universidad y más concretamente en las tareas cotidianas de alumnos, personal docente, etc., han conseguido ya una masa crítica de usuarios, suficientemente amplia como para forzar un posible cambio de modelo, pasando del uso de paquetes de ofimática y herramientas de Internet privadas a soluciones libres.

En el campo de la Geomática, para que las soluciones libres consigan esa misma masa crítica de usuarios que permita consolidar las alternativas libres, es preciso la formación de alumnos y profesorado en la utilización de SIG libres. Creemos sinceramente que la Universidad tiene la obligación moral y ética de formar a futuros profesionales y docentes mediante herramientas libres. De otro modo, tal y como muy gráficamente se apuntó en una mesa de debate durante las pasadas Jornadas de SIG Libre, estaremos creando pequeños delincuentes en potencia, ya que, ante la casi segura imposibilidad de obtener licencias de productos privados -generalmente de alto coste económico- buscarán la manera de obtenerlas de forma, podemos decir, ilícita. Y este argumento tan sólo hace hincapié en el aspecto económico del problema, aunque podrían mencionarse muchísimos más.

En concreto, y sobre la entrada de gvSIG (sin olvidar el proyecto SEXTANTE) en ambientes universitarios, cabe destacar que su irrupción meteórica (nos atreve-

ríamos a decir que sin precedentes en el campo de la Geomática, tanto por su velocidad como por su popularidad) tendrá que suponer, a la fuerza, un cambio en el uso de los programas de SIG en la docencia. GvSIG, además, en poco tiempo se ha convertido en uno de los abanderados del SIG Libre en las universidades y, sin duda alguna, ha supuesto la puerta de entrada a un universo de alternativas libres en el campo de la docencia universitaria.

En UNIGIS, un ejemplo muy próximo para nosotros, se ha pasado de un modelo en el cual se utilizaba 100 por cien software privado, a un modelo prácticamente inverso, empezando desde la propia plataforma educativa, pasando por los programas que se utilizan para el desarrollo normal del curso y para la consolidación de todo el contenido y conceptos teóricos.

¿Qué prima más en la implantación del Software Libre en la Universidad: la ética o la técnica?

A.P. En cuanto a la implantación del Software Libre en la Universidad, quizás en un primer momento prima la cuestión ética, desde tres puntos de vista: político, económico y filosófico. Desde un punto de vista político porque hay un claro interés por parte de las administraciones (regionales, nacionales y europeas) de fomentar el desarrollo de Software Libre. Con ello se busca tanto el desarrollo de un tejido empresarial local como una supuesta independencia de fabricantes de software "no locales". Esto lleva a que haya un buen número de proyectos financiados, subvencionados, contratados, etc.



Desde un punto de vista económico, por el ahorro en licencias que representa: ¿hasta qué punto es ético gastar dinero en algo para lo que existe un equivalente gratuito? Y esta pregunta es especialmente relevante en entidades públicas en las que se gasta el dinero de todos.

En cuanto al punto de vista filosófico, por lo que representa el Software Libre en sí: la libertad de poder estudiar, modificar y adaptar el código (con toda la carga pedagógica que ello conlleva); pero también la certeza que se puede conocer todo aquello que se instala. Y es que a menudo las entidades se preguntan, respecto al software propietario, si hace sólo aquello que se supone que está haciendo.

Sin embargo, una vez superado el primer paso (instalar el código), que es el trigger que marcaría el punto de partida para utilizar Software Libre, la condición necesaria para que se consolide es la calidad técnica del mismo. Al fin y al cabo, la Universidad tiene la obligación moral de dar la mejor enseñanza posible a los estudiantes y procurar proporcionar las mejores herramientas. Por tanto, cuando trabajar con Software Libre implica un descenso de la calidad, el que prima es el propietario.

Otro punto de vista desde el que habría que responder a la pregunta es el de la Investigación y Desarrollo (I+D) propiamente dicho. En este sentido cabe decir que en España hay pocos centros que hagan realmente I+D en áreas ligadas a los SIG. Hay pocos estudios de tercer grado ligados al desarrollo de sistemas SIG, geometría computacional, agentes inteligentes, lógicas difusas, computación GRID, generalización automática, etc. Es decir, aquellas áreas de interés en las que otros países del entorno están ya investigando. Por poner un ejemplo, el proyecto gvSIG mismo, en el que participa directamente la Universidad Jaime I, tiene una componente más importante de desarrollo que de investigación, a pesar que poco a poco va convirtiéndose en soporte para ciertos grupos de investigación.

Cambiar este sesgo a la parte de D requiere un cambio de escenario que implique políticas en investigación y financiación parecidas a las de otros países comunitarios, capaces de generar un tejido empresarial que incluya empresas spin-off que faciliten la transferencia tecnológica Universidad-empresa y viceversa. Y sobre todo es necesario un cambio

de mentalidad de la sociedad en general. No hay que olvidar que la parte de I implica inversiones con resultados a medio y a largo plazo. Aún son loables los pasos que se están haciendo para llevar a cabo desarrollos con empresas en que éstas ponen la componente de D y la Universidad la componente de I.

En cuanto a los aspectos éticos del Software Libre en I+D en la Universidad, es cierto que existen grupos de investigación centrados en la explotación y desarrollo del mismo y es aquí donde puede esperarse un desarrollo mayor, aunque en el caso concreto de SIG, una vez más, es difícil saber si es una apuesta estratégica de la Universidad, u obedece más bien a una cuestión coyuntural o, incluso, personal del interés de las personas implicadas.

“Creemos sinceramente que la Universidad tiene la obligación moral y ética de formar a futuros profesionales y docentes mediante herramientas libres”

L.V. Por lo general, según nuestra propia experiencia, en la Universidad han primado desde siempre los aspectos técnicos. En un momento concreto, allá por 1997, se hizo una apuesta de futuro en base a tres criterios: económico, tecnológico y organizativo. Una vez valoradas todas las alternativas en base a los criterios ya mencionados, se decidió apostar por el software privativo en lugar de adoptar soluciones libres. Las razones que en su momento adujo la Universidad para justificar dicha elección fueron que, desde un punto de vista económico, las empresas productoras de software privativo ofrecían ofertas difícilmente rechazables. Desde el punto de vista organizativo, se intentó que el backoffice y frontoffice de todas las aplicaciones pertenecieran a una misma tecnología (empresa). Ya por último, desde el punto de vista tecnológico, primó la elección de la tecnología que ofreciera el menor número de problemas posibles... Efectivamente, todas estas razones planteadas pueden ser ampliamente debatidas y rebatidas y, sin duda alguna, invitan a la reflexión y al debate acerca del papel que deben jugar las universidades en la adopción y promoción de las soluciones libres.

Muchas de las razones que condujeron a la Universidad a apostar por un

modelo de software no libre, hoy en día carecen de validez incuestionable pero, lamentablemente, nada hace pensar que esto vaya a cambiar en un período relativamente corto de tiempo, a no ser que exista una apuesta firme y clara por parte del órgano de gobierno de la Universidad. En otras palabras, estamos a la espera de una apuesta política para iniciar un verdadero cambio de modelo. Creo que en ningún momento se ha planteado un debate profundo acerca de los aspectos éticos de la utilización de uno u otro modelo. Y pienso sinceramente que la Universidad no debería dejar a un lado, bajo ningún concepto, los criterios éticos en cualquier decisión, a menos que esté dispuesta a renegar de su componente o vertiente más social, la de la Universidad como formadora de personas.

¿Qué papel crees que debe jugar el Software Libre en la Universidad? ¿Y la Universidad en el Software Libre?

A.P. Respecto al papel que el software de fuentes abiertas debe jugar en la Universidad, siempre que exista una alternativa viable al Software Libre, la Universidad debería apostar por ella, por las razones éticas anteriormente comentadas. Si bien es cierto que en algunos casos, por razones profesionales y de mercado, interesa enseñar software propietario, también lo es que cuantos más estudiantes se habitúen a utilizar Software Libre, más profesionales habrá después habituados a trabajar con él.

También hay que decir que ciertos aplicativos libres juegan, desde un punto de vista pedagógico, el mismo papel que la alternativa propietaria. En estos casos las aportaciones del software propietario son sobre todo de rendimiento: necesarias en un entorno profesional, pero posiblemente no en uno docente.

No hay que olvidar, sin embargo, que ciertos nichos de mercado están copados por alternativas propietarias y, en estos casos, la comunidad es tan pequeña que no se da la masa crítica necesaria para llegar a obtener una alternativa libre viable.

Sobre el papel que debería jugar la Universidad en el Software Libre, debería ser el de implicarse directamente en el desarrollo. La Universidad es un importante foco de generación de software: prácticas de estudiantes, proyectos finales de carrera, proyectos de investigación, etc. Si se detectan errores en un software, si se implementan mejoras o desarrollos, etc.

III Jornadas de SIG Libre

11, 12 y 13 de Marzo de 2009
Girona, España

**"Ponencias, debates, presentaciones,
talleres y reflexiones alrededor
del software libre aplicado a
los Sistemas de Información
Geográfica"**



www.sigte.udg.es/jornadassiglibre

todo ello debería contribuir a la comunidad y al desarrollo de estos productos. Además, el ahorro en licencias que implica utilizar software de código abierto se podría dedicar al desarrollo de proyectos con lo que todo ese dinero revertiría directamente en la sociedad.

De hecho, la Universidad lleva desde hace años produciendo Software Libre, incluso SIG, con resultados que han tenido un éxito notable: GRASS o Map-Server son quizás los ejemplos más conocidos.

Éste es el tipo de acciones que habría que potenciar. Aunque eso sí, poniendo más el acento en la "I" que en la "D": a pesar que la Universidad tiene que tener un ojo puesto en el mercado, no puede ser cautiva del mismo.

Por otro lado, desde el punto de vista de los estudiantes, no hay duda que es mucho más gratificante llevar a cabo un desarrollo que luego utilizará la comunidad y, desde el punto de vista de los investigadores, compartir con la comunidad los propios desarrollos sería una forma de llevar a cabo directamente el traspaso de ciencia a sociedad.

L.V. Mi opinión personal es que deberían ser elementos indisolubles. Universidad y Software Libre tienen muchos puntos en común y es que ambos, como dice la célebre cita (ya fuera Bernardo de Chartres o Isaac Newton quien la verbalizara por primera vez), han viajado en el tiempo a 'hombros de gigantes'. Si bien algunos de los materiales docentes utilizados hoy en día en las universidades puede que hayan sido creados por el propio profesorado, no hay que olvidar que estos beben de otras fuentes, obras y creaciones... Lo mismo sucede con el Software Libre. Sólo por esta simple razón, ambos deberían ir de la mano.

Así pues, parece de pura lógica que por ésta y más razones que podríamos ir esgrimiendo a lo largo de estas líneas, relacionadas con los valores intrínsecos a la Universidad en general, ésta se erigiera como centro difusor de las alternativas libres en todos sus campos, a la par que los proyectos libres podrían beneficiarse a su vez del poder de innovación y creación de nuevos materiales, ideas, proyectos ... existentes en las universidades.

Iniciativas como la Declaración Institucional de la Universidad de Zaragoza so-

bre el Software Libre, debería estar a la orden del día y, sin duda, viene a confirmar que lo que tan sólo hace unos años podía etiquetarse como una utopía, va tomando cuerpo y fuerza. Eso sí, resulta interesante comentar que declaraciones como la anteriormente mencionada no pueden, ni deben, quedarse en un plano teórico, sino que debe materializarse a la práctica, apostando con decisión por las alternativas libres.

A modo de conclusión sobre las posibles relaciones y sinergias existentes entre la Universidad y el Software Libre, me gustaría plantear una idea que invite a la reflexión acerca del conocimiento libre... ¿Cuánto tiempo deberemos esperar para que todas las universidades, decididamente, liberen el conocimiento encerrado en los materiales docentes, tal y como el Software Libre libera sus líneas de código?

“La Universidad no debería dejar a un lado, bajo ningún concepto, los criterios éticos en cualquier decisión, a menos que esté dispuesta a renegar de su componente o vertiente más social, la de la Universidad como formadora de personas”

Concretando en el proyecto gvSIG, ¿cómo se plantea la colaboración?

A.P. En cuanto a la colaboración de la UOC con gvSIG está en la línea que se apuntaba al final de la pregunta anterior. En la UOC se desarrollan proyectos y trabajos finales de carrera (TFC y PFC) de SIG en las titulaciones de Ingeniería Técnica en Informática de Gestión, Ingeniería Técnica en Informática de Sistemas, Ingeniería Informática, Ingeniería Técnica de Telecomunicaciones, especialidad Telemática.

Los proyectos finales de carrera de la UOC están asignaturizados y en ellos es el profesor quien propone los proyectos a llevar a cabo. Estos se desarrollan a lo largo de un semestre y, si el estudiante no termina, cuando vuelve a matricularse del proyecto, debe empezar desde cero un proyecto distinto. Dado que los estudiantes no conocen el proyecto hasta el día que empieza el curso, y que todos terminan al mismo tiempo, varios estudiantes pueden llevar a cabo el mismo proyecto, o un proyecto mayor de forma coordinada. La poca duración del proyecto, y el hecho

que la asignatura tiene una parte importante de gestión y redacción, deja poco hueco para la investigación. Pero aún así, con proyectos claros y bien acotados, los estudiantes han demostrado ser capaces de hacer trabajos realmente extraordinarios.

Desde principios de 2007 hay una línea de proyectos abierta en que se utiliza gvSIG, que se ha revelado como una plataforma ideal para este tipo de proyectos. Durante los primeros semestres se llevaron a cabo una comparativa de gvSIG con software propietario y un desarrollo de topología para gvSIG, entre otros.

Dada la buena sintonía con el equipo de gvSIG, se les propuso definir los proyectos coordinadamente con ellos, de manera que el trabajo desarrollado por los estudiantes de la UOC estuviera en línea con el camino a seguir por gvSIG. Así, desde este semestre, los proyectos finales de carrera de la UOC que utilizan gvSIG, se llevan a cabo en coordinación con el equipo del mismo y en la línea del proyecto. gvSIG propone algunas opciones, pero deja opción a las aportaciones de estudiantes y profesores.

No cabe duda que es una simbiosis ideal, por un lado porque el proyecto gvSIG se ve enriquecido y, por el otro, porque los estudiantes llevan a cabo proyectos reales de utilidad práctica.

L.V. La colaboración de la Universitat de Girona, en estos momentos, se basa en el testeo de las versiones que se encuentran actualmente en las cocinas de gvSIG. A corto plazo, nuestra manera de contribuir al proyecto será ayudar en todo lo posible, al igual que decenas o centenares de usuarios realizan a nivel personal o colectivo, en el testing de versiones y pilotos, reporte de errores, propuestas de mejora, etc. Contribuir en la medida de lo posible con estas actividades, es nuestra manera de aportar algo al proyecto gvSIG.

Además, gvSIG conjuntamente con SEXTANTE, son dos alternativas libres ampliamente utilizadas en nuestro currículum académico (como nodo español de la red UNIGIS) contribuyendo a la formación de nuevos profesionales y futuros formadores en el uso de herramientas SIG Libres. Así pues, el binomio gvSIG-Sextante se ha convertido en nuestra principal herramienta de trabajo en nuestros proyectos docentes y educativos, ya sean on line o presenciales.



Comunidad gvSIG: Europa

Alessandro Sgambati

Responsabile dell' Ufficio cartografico
Ispettorato ripartimentale foreste di Trieste e Gorizia
Regione Autonoma Friuli Venezia Giulia

Chris Puttick

CIO Oxford Archaeology: Exploring the Human Journey



¿Cuál crees que debe ser el papel que juegue el Software Libre en el desarrollo de la Administración Pública?

Alessandro Sgambati. El Software Libre permite un considerable ahorro de dinero público que puede ser invertido en otras necesidades.

Al ser gratuito, puede ser utilizado por cualquier Administración. Si se estandarizara el uso de Software Libre, se fomentaría la colaboración entre las administraciones.

En Trieste, anteriormente, habíamos intentado colaborar con Bomberos, Policía, Ayuntamientos, también en Eslovenia. El problema siempre fue que cada Administración tenía su propio programa SIG. Ahora, con gvSIG podemos colaborar, considerando gvSIG como un estándar de uso, compartido por todas las Administraciones.

Como el Software Libre es modificable y adaptable a las necesidades de cada Administración, se convierte en una herramienta perfecta que permita a los empleados técnicos de las Administraciones crecer profesionalmente, desarrollando de manera autónoma y gestionando con independencia, sin depender de empresas privadas para administrar asuntos públicos.

Chris Puttick. For free/open source software generally, I think it is clear it will become the normal approach over the next few years. It is simple economics; as the market buyers become more knowledgeable (and more knowledgeable people make the ICT strategy decisions in organizations), people will start acquiring the best value product. Free upgrades for life and a zero price of acquisition is difficult to argue against without resorting to "but it is not what we do now" as your counterargument.

Open source also gives an additional advantage to companies, the ability to "have your cake and eat it" i.e. all the benefits of a custom app that fits your organization's needs with an off-the-shelf product as the base.

For gvSIG particularly I think it gives a new approach to the use of GIS a bit like how the introduction of the world wide web put knowledge into everyone's reach, gvSIG gives the opportunity to put GIS on every desktop and therefore geographic knowledge into every member of staff's reach. Combining this with other open source GIS products, such as PostGIS, Mapserver and OpenLayers leads to a highly capable SDI solution, where desktop users can easily modify and analyse geospatial information and provide that information via the web in realtime.

Además de usar gvSIG, ¿sois colaboradores del proyecto. ¿Por qué pensáis que la Administración debe no sólo utilizar sino también apoyar los proyectos libres?

A.S. Creo que la difusión de proyectos libres y el uso de estándares permite una mejor transparencia de la acción administrativa y ayuda a la participación de la ciudadanía. Las Administraciones Públicas tendrían que apoyar los proyectos libres.

C.P. Because you would be stupid not to. Free software is not no cost, and some of the advantages are only realisable if you are part of the community. Support does not necessarily mean contributing cash, code or bugfixes, or even end-user documentation, but you need to be part of the community to learn how to get the most from the product. By being an active member of the community, even if just by posting the odd feature request, you help to sustain and grow the community, which in essence is a community that lives entirely on information exchange. Even if your primary contribution is to tell others about the product, that is supporting the project and helping develop it.

Being a part of the community gives you a chance to contribute to the roadmap, a chance to gain plugins that help enhance your workflows and productivity, a chance to see bugs that annoy you quashed faster. Not being part gives you no chance.

Contributions of any kind help the community to grow and the related software to thrive and become better. The software becomes better, the community thrives; more people start to use the product and contribute. It's called a virtuous circle.

Concretamente. ¿Cómo colabora vuestra administración con gvSIG?

A.S. La Administración de la Regione Autonoma Friuli Venezia Giulia aún no colabora de forma oficial con el proyecto gvSIG. Hay una docena de personas que están muy interesadas, tenemos cinco personas testeando el programa y dos o tres que lo utilizan frecuentemente. El descubrimiento de gvSIG ha sido verdaderamente una pequeña revolución en nuestra forma de trabajar, impulsando una mayor colaboración entre compañeros. Probablemente gvSIG Mobile será la primera herramienta libre utilizada por los empleados técnicos forestales de forma oficial, abriendo así la puerta al uso de gvSIG.

Personalmente he colaborado en traducir el "Manual del usuario" a italiano y en probar las varias ediciones beta de gvSIG Mobile. La difusión entre mis compañeros de qué es gvSIG y de lo que hace, ha sido también una tarea que me ha quitado bastante tiempo, pero lo hago siempre con mucho entusiasmo.

C.P. We are active members of the community in the ask (and occasionally answer) questions sense; we also are working closely with the gvSIG mobile team and have contributed (OpenMoko and keyboard) hardware to this effort. We've also recently released an integrated installer "gvSIG OA Digital Edition", originally designed for internal quick deployment of the software with plugins and latest JRE wrapped in, and are working on end-user training packages. And some translation work.

Oh, and we talk about it to other people.

A lot.



Comunidad gvSIG: Latinoamérica

Doctor Carlos Figueira

Presidente del Centro Nacional de Tecnologías de Información
 Ministerio del Poder Popular para las Telecomunicaciones
 y la Informática
 Gobierno Bolivariano de Venezuela

Latinoamérica está siendo, a nivel global, uno de los núcleos más activos en la apuesta por el Software Libre. Microsoft, que factura en venta de licencias unos 800 millones de dólares anuales en Latinoamérica, advirtió que el open source daña la economía latinoamericana. ¿Qué papel piensan que puede jugar el Software Libre en el desarrollo de Latinoamérica?

Carlos Figueira. El Software Libre fue adoptado en 2004, como normativa en el Estado Venezolano, como instrumento fundamental en el logro de la Soberanía y la Independencia Tecnológica. En marzo de 2008, Ecuador hizo lo mismo. Brasil tiene regulaciones similares a nivel regional. Otros países están considerando, igualmente, tomar una medida similar.

Además de las claras ventajas del Software Libre en la apropiación del conocimiento de los países, se está convirtiendo en una herramienta para la cooperación entre países aliados de la región.

Juan Ernesto Rickert. El Software Libre juega un papel muy importante en el desarrollo tecnológico de Latinoamérica ya que posibilita a un sinnúmero de instituciones y personas de los mas diversos niveles y recursos económicos, acceder a una tecnología muy estable, desarrollada por miles de personas de todo el mundo, con acceso a su código fuente para realizar las modificaciones que se consideren necesarias y sin necesidad de realizar erogaciones de dinero.

Esto permite utilizar las últimas herramientas disponibles en el mundo, en

forma permanente, aunque no se disponga de presupuesto para invertir en software. Esto es, sin duda, un factor muy importante que facilita y favorece el desarrollo.

“El CNTI promueve el uso de gvSIG como una herramienta en Software Libre y con un modelo de desarrollo completamente alineado con las políticas del Estado en materia de apropiación del conocimiento”

¿Existen iniciativas que permitan la colaboración en materia de Software Libre entre los distintos países? Y en el caso de gvSIG, ¿se percibe la existencia de una comunidad latinoamericana?

C.F. Recientemente, en el congreso CONSEGI (Brasilia, Brasil, agosto 2008) (www.consegi.gov.br/) se establecieron compromisos para establecer una cooperación sobre Software Libre, con participación de Argentina, Brasil, Cuba, Ecuador, Paraguay y Venezuela.

Adicionalmente, Venezuela tiene actividades de cooperación directas con Cuba y Bolivia en Software Libre.

J.E.R. Efectivamente existen algunas iniciativas que permiten la colaboración en materia de Software Libre entre los distintos países. Nada mejor que gvSIG como ejemplo, que cuenta con una gran comunidad mundial de usuarios/colaboradores. Quisiera remarcar la palabra “colaboradores”, ya que una parte importante de los usuarios colaboran con nuevos

desarrollos complementarios para sumar funcionalidades a gvSIG.

Es realmente sorprendente lo que son capaces de lograr estos entornos colaborativos a nivel mundial donde cada parte, cada eslabón de la cadena, aporta lo que sabe y puede hacer en beneficio de la comunidad, que gracias a la gran conectividad o “globalización” esa comunidad es el mundo entero.

En lo que hace a la comunidad latinoamericana, ésta se encuentra en un constante crecimiento en el empleo de gvSIG y estimo que esta tendencia se incrementará con el correr del tiempo. Hay varios países en los que se desarrollan cursos sobre el empleo de este software que posibilita y posibilitará el acceso a gestionar información geográfica a un gran número de instituciones y personas con las conocidas ventajas del Software Libre.

¿Se usa gvSIG en su país? ¿Qué impacto tiene en la actualidad y qué expectativas de avance ven en los próximos años?

C.F. gvSIG se usa en varias instituciones públicas y empresas en Venezuela. Esperamos que aumente dramáticamente a través de la instrumentación de las políticas en materia de Software Libre, y del proyecto de Infraestructura de Datos Espaciales Venezolana IDEGEOVEN.

J.E.R. En Argentina, se utiliza gvSIG y, como mencionaba anteriormente, permite gestionar información geográfica sin necesidad de contar con una licencia de pago.



Ingeniero Juan Ernesto Rickert

Especialista en Infraestructuras de Datos Espaciales Jefe de Tecnología del Instituto Geográfico Militar de la República Argentina

Codirector del proyecto PROSIGA

La realidad en mi país es que existen una gran cantidad de municipios, en lugares con escasos o muy escasos recursos económicos, también investigadores o estudiantes de carreras de grado relacionadas con la información geográfica, a quienes les sería totalmente imposible acceder a cubrir sus necesidades con software de pago. Gracias a gvSIG esto se encuentra solucionado y resuelto.

En los próximos años la tendencia se mantendrá en crecimiento. gvSIG es muy útil y casi permanentemente se le incorporan nuevas funcionalidades, lo que permite satisfacer nuevas exigencias.

Desde su organización, ¿cuál es el grado de colaboración con el proyecto? ¿Cómo le gustaría que avanzara esta colaboración?

C.F. El CNTI (Centro Nacional de Tecnologías de Información) promueve el uso de gvSIG como una herramienta en Software Libre y con un modelo de desarrollo completamente alineado con las políticas del Estado, en materia de apropiación del conocimiento. Estamos dispuestos a financiar y promover desarrollos sobre gvSIG, así como de establecer las alianzas que permitan el crecimiento y sostenibilidad de este importantísimo proyecto.

J.E.R. Desde la organización a la que pertenezco, nuestra colaboración no es muy grande ya que contamos con muy poco personal como para participar en

el desarrollo de nuevas funcionalidades a gvSIG.

Nos hemos convertido en grandes difusores de este proyecto, haciendo en forma permanente referencias a sus ventajas.

“En 2009 organizamos en Argentina las I Jornadas gvSIG Latinoamérica”

Dentro del marco del proyecto PROSIGA que se encuentra conformando la Infraestructura de Datos Espaciales de la República Argentina, se están planificando cursos de gvSIG para el año 2009.

También organizaremos en Argentina las I Jornadas gvSIG Latinoamérica.

En lo personal, me gustaría aportar un grupo de programadores que colaboren activamente en el desarrollo del proyecto, pero hoy me resulta difícil poder asumir este compromiso. Actualmente soy usuario, colaboro en el comité científico de las jornadas gvSIG, estoy convencido de que es un gran proyecto y difundo lo más que puedo sus funcionalidades y ventajas.



Comunidad gvSIG: España

Juan Antonio Bermejo Domínguez

*Técnico GIS (Sistemas de Información Geográfica)
 Dirección Insular de Nuevas Tecnologías y Energías Renovables
 Área de Planificación y Servicios Públicos
 Exmo. Cabildo Insular de La Palma*



¿Cuáles fueron los motivos que llevaron a su organización a apostar por las tecnologías abiertas en materia de SIG?

Juan Antonio Bermejo. El Sistema de Información Geográfico Corporativo (SIG) así como la implantación de la Infraestructura de Datos Espaciales (IDE) de la Isla de La Palma se planteó inicialmente como un proyecto a medio plazo, de progresión constante y orientado no sólo a la publicación de información siguiendo la Directiva INSPIRE, sino también a la explotación del propio sistema mediante herramientas SIG avanzadas. El proyecto nace hace solo tres años, y en un principio estaba encaminado únicamente a suplir las carencias en materia de Información Geográfica dentro de nuestra Administración Local, el Cabildo Insular de La Palma. Con el devenir del proyecto nos hemos dado cuenta de las necesidades del resto de Administraciones Locales insulares, y hemos puesto en marcha un proyecto más ambicioso, de carácter Insular, que aglutine al resto de Administraciones y Organismos con competencias en la Isla.

Inicialmente, se analizaron las tecnologías y los medios disponibles, se estudiaron las distintas alternativas y se decidió por implantar tecnologías abiertas basándonos en el menor coste de implantación, la optimización de la inversión y la no dependencia tecnológica de terceros. El comienzo se produce en un momento fundamental, ya que partíamos prácticamente de cero, lo que en un principio parecía una desventaja se convirtió en una fortaleza porque no heredábamos modelos, formatos ni licencias.

En ese instante gvSIG se afianzaba como Software Libre SIG a nivel nacional y presentaba varios elementos fundamentales para nosotros: código abierto, multiplataforma, estabilidad, excelente adaptación a los estándares OGC y a los diversos formatos vector-raster, rápida evolución, amplia comunidad participativa de usuarios y varios casos de éxito en la

Administración Pública. Sin duda, era el software a utilizar. El problema del cliente de escritorio para la explotación del SIG y la IDE quedaba resuelto. Respecto a las bases de datos espaciales, optamos por implantar PostGIS como solución open-source y Mapserver, como servidor de mapas de nuestra IDE.

¿Colaboran con proyectos de SIG libre?

J.A.B. En la medida de nuestras posibilidades participamos en el proyecto gvSIG, comunicando sugerencias y mejoras del programa, difundiendo la aplicación como solución óptima en la administración, testeando nuevos desarrollos, dando charlas y cursos en Universidades, en definitiva, participando como usuarios y difusores de este magnífico proyecto colaborativo.

¿Están colaborando con otras Administraciones o Universidades en materia de SIG? ¿Qué problemas encuentran en la colaboración?

J.A.B. Uno de los pilares básicos de nuestro proyecto de SIG e IDE es el establecimiento de acuerdos entre los distintos productores, integradores y usuarios de datos en el ámbito insular. El Cabildo Insular de La Palma, consciente de este problema, está estableciendo un marco común en La Isla de La Palma, a través de la firma de convenios de colaboración con las distintas Administraciones con competencias en la Isla. Se intenta con esto promover una serie de líneas de actuación de diversa índole, como son:

- > Promover y Desarrollar el Sistema de Información Geográfica y la Infraestructura de Datos Espaciales de La Isla de La Palma.
- > Promover la formación de técnicos en el uso de herramientas de gestión y tratamiento de información geográfica.
- > Coordinar la producción cartográfica de la Isla de La Palma.

- > Intercambiar información territorial en las condiciones que cada uno de los distintos sectores económicos y sociales precise, así como la utilización óptima de la información acumulada por ambas entidades.
- > Promocionar la modernización de las Administraciones Públicas y entidades locales y el aumento de la calidad de los servicios a los ciudadanos.
- > Introducir la cultura geográfica en la sociedad como un elemento complementario cada vez más imprescindible en el análisis, la planificación y la gestión del territorio insular.
- > Potenciar la Información Territorial como infraestructura de desarrollo.
- > Promover el intercambio de iniciativas y experiencias con otros territorios.

Para poder crear un marco común de actuación, se ha desarrollado un portal web de conocimiento e interacción www.siglapalma.es, que sirve de nexo entre los distintos actores, los datos y las herramientas, a la par de servir de portal de referencia para la interconexión, formación y la difusión.

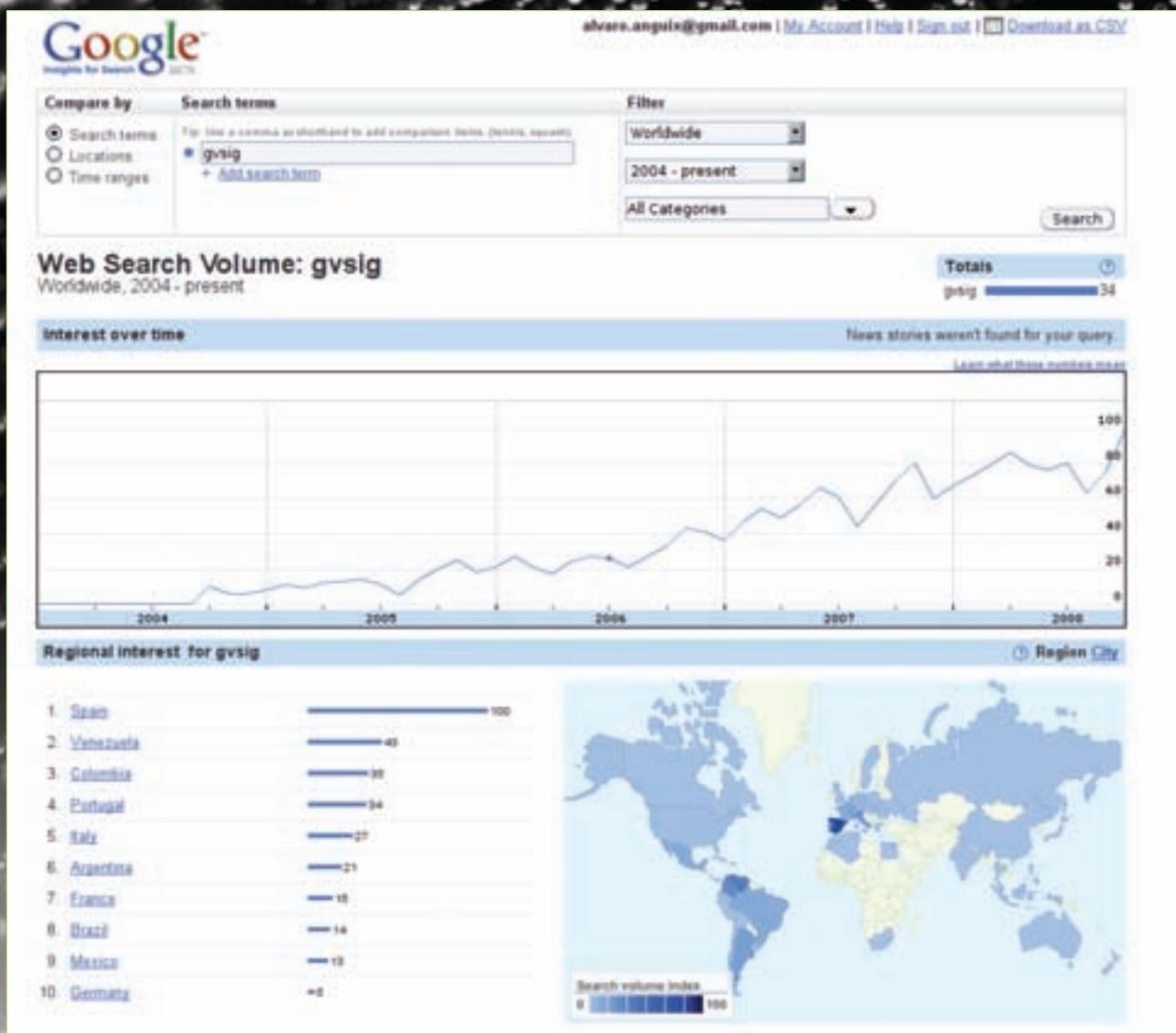
“Avanzando Juntos” es el lema de las 4as Jornadas de gvSIG, ¿qué debe mejorar el proyecto para facilitararlo?

J.A.B. A mi entender debe hacerse un mayor esfuerzo en dos líneas fundamentales:

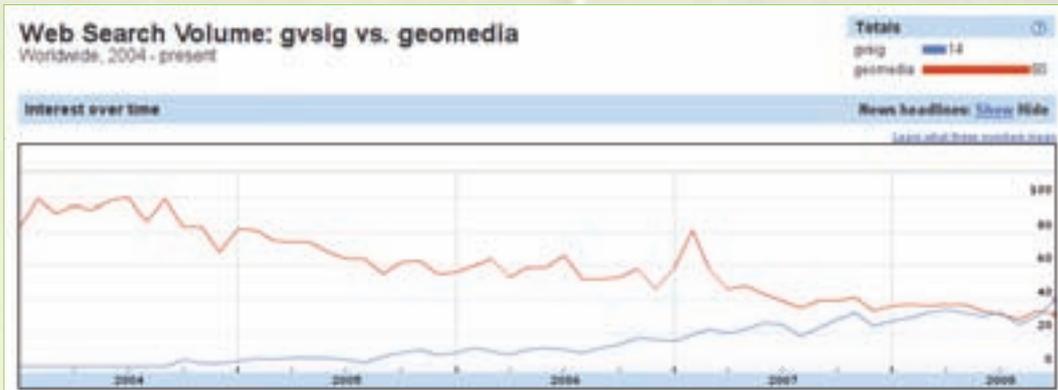
1. Llegar más y mejor a los centros educativos a todos los niveles, pero especialmente difundir y hacer partícipes de la aplicación a las Universidades, incentivando a desarrollar y crear proyectos con gvSIG a través de programas de becas y premios.
2. Afianzar gvSIG a través de una Organización estable de Administraciones, Instituciones y empresas que de alguna manera le den más solidez y peso al proyecto, y que posicionen gvSIG como un referente de desarrollo SIG a nivel mundial.

gsSIG en la Web

“Google Insights for Search” es una nueva herramienta de la factoría Google que muestra cuáles son los términos más buscados en la Red dentro de cada país o en períodos concretos de tiempo.



Veamos la evolución del proyecto en cuanto a búsquedas realizadas por los internautas.



Otra de las características de esta herramienta es que permite comparar distintos términos. Es curioso como gvSIG llega a superar en los últimos meses a herramientas privativas tan consolidadas como Geomedia.



Si comparamos con los productos más extendidos en el mercado, ArcView y MapInfo, vemos que se está produciendo un descenso fuerte frente al constante avance de gvSIG (y poco apreciable en esta gráfica).



Sin embargo, si centramos la gráfica en el año 2008, podemos observar cómo la relación de gvSIG con ambos productos privativos se ha acercado considerablemente.

gvSIG en la Blogosfera

Blogosfera, blogsfera, blogalaxia o blogósfera es el término bajo el cual se agrupa la totalidad de weblogs y se deriva de la palabra inglesa blogosphere. Debido a que los blogs o las bitácoras están conectadas por medio de enlaces, comentarios, históricos y referencias han creado y definido su propia cultura. Por lo tanto, la blogosfera como palabra y concepto es inherente a los weblogs.

Podemos considerar la blogosfera como un sistema virtual, en el que se establecen comunidades de weblogs, categorizadas temáticamente o por perfiles de interés. Estos conforman, pues, el mensaje y la blogosfera, el lugar para habitar en Internet.

(Cabero, 2006, p.234)

Mientras que los blogs, por sí mismos, son sólo un formato en la web, la interconexión de estos es un fenómeno social: al verlos como un todo se pueden determinar claramente tendencias, gustos, popularidad de sitios, objetos, productos, música, películas y libros, como si fuera un ente colectivo.

(Wikipedia)



The collage features several elements related to gvSIG and its community:

- Blog Post:** A screenshot of a blog entry titled "gvSIG y Mac: problemas y a..." (gvSIG and Mac: problems and a...). The post includes an "Archivo del blog" (Blog Archive) section and "Datos personales" (Personal Data).
- Coverage Report:** A screenshot of a "Coverage Viewer" interface showing a table of branch coverage data.
- Blog Archive:** A screenshot of a blog archive showing a post from "Lunes 11 de septiembre de 2006" titled "Grilla creada con..." (Grid created with...).
- Blog Post:** A screenshot of a blog post titled "Disponible el prototipo de estadísticas para gvSIG 4.1.1" (Available the prototype of statistics for gvSIG 4.1.1), dated "6 September 2006".
- Blog Post:** A screenshot of a blog post titled "Grilla creada con..." (Grid created with...), dated "Lunes 11 de septiembre de 2006".

Branch	Line	Name	Branch Coverage
+	+	Función core	58.11% (179/308)
+	+	Función core.fuitero	84.84% (146/172)
+	+	Función core.evento	100.00% (11/11)
+	+	Función core.fuitero.evento	8.88% (28/315)
+	+	Función core.fuitero.evento.evento	8.88% (28/315)
+	+	Función core.evento.evento	25.00% (12/48)
+	+	Función framework	77.89% (119/153)
+	+

La Blogosfera es, sin duda, el colectivo más activo y fresco de Internet. gvSIG ha visto incrementada su difusión, en gran parte, gracias al trabajo y entusiasmo de muchos 'bloggers', que sin estar sujetos a ninguna organización, han decidido estudiar, analizar, criticar o publicitar gvSIG.

A continuación, como una pequeña muestra de agradecimiento a todos ellos, queremos hacernos eco de algunos de ellos, como muestra representativa de lo que se puede encontrar "ahí fuera".

The collage consists of several elements:

- Software Architecture Diagram:** A block diagram showing the structure of gvSIG. It includes layers for 'Base Tools & Architectural patterns', 'UI' (with sub-components like Base User Interface, Project Document Model, and UI Extensions), 'Logic Geo' (with sub-components like Map Control, Map Context, and Data Access Library), and 'Data Storage' (with sub-components like SHP, Postgres, and DWG).
- Data Table Screenshot:** A screenshot of a table with columns 'Name', 'Branch Coverage', and 'Use'. The data is as follows:

Name	Branch Coverage	Use
funit.core	78.11% (179/228)	129
funit.core.builders	56.86% (148/260)	20
funit.core.events	100.00% (1/1)	0
funit.core.listeners	8.89% (2/23)	10
- Blog Post Snippet:** A snippet from a blog post titled 'Krimskrams' with a date of '2008-10-22'. It features a small image of oranges and a white flower.
- Magazine Cover:** The cover of a magazine titled 'TECNOLOGIA SIG EN COLOMBIA' with the subtitle 'UNA REVISTA DE INFORMACIÓN, ASESORÍA Y FORMACIÓN PARA EL SECTOR SIG'. The cover features the gvSIG logo and a map.

Editorial

“En tan sólo cuatro años las jornadas de gvSIG se han convertido en un referente internacional en el mundo de los Sistemas de Información Geográfica y de las Infraestructuras de Datos Espaciales. Un lugar de encuentro donde cada vez un mayor número de expertos nos enseña no sólo que es posible ofrecer soluciones profesionales en estos campos basándose en la integración de tecnologías libres, sino que estas tecnologías terminan por resultar mejores que las soluciones clásicas basadas en modelos privativos que no permiten el acceso al conocimiento.

Se presentan soluciones que siguen fielmente los estándares de interoperabilidad internacional y que, además, resultan más sostenibles ya que, es importante recordarlo tantas veces como sea necesario, la utilización de tecnologías libres, aquellas que nos permiten el acceso al conocimiento, son las que nos van a proporcionar la independencia necesaria que nos permita ser libres de marcar la evolución de nuestros Sistemas de Información, aspecto fundamental para toda organización.

El lema de estas jornadas es el de: gvSIG. Avanzando Juntos. Recordamos que las anteriores jornadas fueron con el lema Consolidar y Avanzar, en ellas la reflexión que planteábamos era la siguiente: gvSIG había crecido mucho y muy rápido, era el momento de poner orden, de consolidar todo ese crecimiento de manera que nos permitiera seguir avanzando. Pues bien, ya estamos preparados para seguir avanzando. Nuestra propuesta ahora no es la de recorrer un camino ya marcado, sino que entre todos construyamos ese nuevo camino. Que nos veamos, que hablemos entre nosotros, que intercambiamos ideas, opiniones, etc., con la finalidad de seguir avanzando. Pero como decimos en el lema de las jornadas: Avanzando juntos.”



Ilustrísimo Sr. D. Gaspar Peral Ribelles
Subsecretario Conselleria de Infraestructuras
y Transporte de la Generalitat de la
Comunitat Valenciana

