

## Integración de GeoTools en gvSIG CE

F. González (1), V. González (2)

(1) Freelance, geomati.co, fernando.gonzalez@geomati.co

(2) Freelance, victor.gonzalez@geomati.co

### RESUMEN

*En el presente artículo se expone el trabajo en curso que se está realizando para la integración de Geotools en gvSIG CE. Desde su nacimiento, la actividad en gvSIG CE ha consistido en la creación de plugins para determinados usos y en la corrección de algunos defectos encontrados en el software. Estas correcciones han sido aplicadas tanto al código de gvSIG CE como al código de las versiones 1.x de la asociación gvSIG, ya que el código de estas versiones es muy parecido y la razón de ser de gvSIG CE ha sido siempre la de sumar esfuerzos. Sin embargo, estas experiencias han hecho patente el hecho de que las necesidades de mantenimiento de gvSIG 1.x son difíciles de asumir por parte del equipo de desarrollo de gvSIG CE. En este contexto, el equipo de desarrollo de gvSIG CE se plantea la tesis de que reemplazar por Geotools todo el código posible de gvSIG reduciría significativamente la cantidad de código y, por tanto, el coste de mantenimiento. Geotools es una librería Java que proporciona acceso a datos en distintos formatos, un marco de geoprocésamiento, implementación de estándares como WMS, WFS, WCS, SLD y SE, motor de renderizado, etc. Es ampliamente utilizado y, de hecho, es la base de proyectos como uDig o Geoserver. Entre las ventajas de usar Geotools podemos destacar que presenta una gran oportunidad para la formación de los desarrolladores de gvSIG CE ya que es una comunidad con desarrolladores de nivel contrastado y que lleva en funcionamiento muchos años. También parece especialmente interesante las posibilidades de contribuir a una librería tan ampliamente usada y en la que hay gran visibilidad sobre los desarrollos realizados.*

**Palabras clave:** gvSIG, geotools, integración

## INTRODUCCIÓN

Desde su nacimiento, el proyecto *gvSIG CE* ha realizado numerosas aportaciones en el dominio del SIG libre. El sistema está siendo utilizado en diversas organizaciones públicas y privadas y se desarrollan continuamente *plugins* para soportar los distintos casos de uso que se dan en estos entornos.

Sin embargo, durante todo este tiempo, el equipo de desarrollo ha experimentado diversas dificultades que frenan el desarrollo deseable para un proyecto de software libre.

En el presente artículo se expondrán estas dificultades, la estrategia elegida por el proyecto *gvSIG CE* para darles solución, el estado del desarrollo de la solución y sus planes futuros, así como la estrategia adoptada mientras dura el proceso de integración.

## TRAYECTORIA

*gvSIG CE* surgió en 2011 como un *fork* de la versión 1.11 de *gvSIG*. La evolución a nivel estructural de ambas líneas de desarrollo ha sido mínima por lo que estos dos proyectos han compartido una base de código común que ha permitido la colaboración a nivel técnico.

Esta colaboración a nivel técnico incluye correcciones de errores y mejoras integradas en *gvSIG 1.12*, realizadas por *geomati.co* (Fernando González) y *CSGIS* (Ruth Schönbuchner y Jose Canalejo), como por ejemplo:

- El soporte de simbología por intervalos para valores negativos [1].
- Carga automática de leyendas *.gvl* [2].
- Almacenamiento de símbolos con ruta relativa, para facilitar la acción de compartir bibliotecas de símbolos entre distintos ordenadores [3].

Así como mejoras en *gvSIG CE* hechas en *SEXTANTE* por su equipo de desarrollo que facilitaron posteriormente la integración de esta biblioteca de análisis espacial en *gvSIG 1.12*.

Además de las correcciones realizadas, se han llevado a cabo diversos desarrollos a medida como por ejemplo:

- Integración de *gvSIG CE* en *KKG* (un software propietario utilizado en mas de 900 municipalidades en Alemania para la gestión municipal). Este proyecto fue presentado en el AGIT 2012 por el Ayuntamiento de Münster, el cual ha distribuido oficialmente *gvSIG CE* dentro de la municipalidad.
- Integración de *Survey2gis* en *gvSIG CE* a través del cual es posible convertir automáticamente datos procedentes de taquímetros y estaciones totales en *shapefiles* y cargarlos en *gvSIG CE*. Actualmente se está trabajando en la estabilidad de este software que se publicara en el 2013 con licencia GPL.

## PROBLEMÁTICA

Durante todo este tiempo se han experimentado diversas dificultades, todas ellas relativas a la incoherencia y complejidad estructural del código de *gvSIG*.

Entre otras cosas, estas dificultades hacen difícil la estimación fiable de los desarrollos salvo para los desarrolladores muy experimentados. Sin embargo, dado que la base de código de *gvSIG* es tan grande, es realmente difícil encontrar este perfil.

La base de código de *gvSIG* está compuesta por cerca de un millón de líneas de código documentadas escasamente y con frecuencia en castellano, lo que dificulta la lectura para desarrolladores de otras lenguas. La presencia de casos de prueba es mínima y la documentación a bajo nivel es escasa.

Todos estos factores, no hacen sino encarecer el mantenimiento del sistema y el coste del desarrollo de nuevas funcionalidades, en un momento en el que existen diversas alternativas interesantes para el desarrollo de GIS de escritorio, como por ejemplo *QGIS*.

Por otra parte el equipo de *gvSIG CE* es relativamente reducido y debe ser gestionado óptimamente para poder darle continuidad al proyecto, por lo que surge la necesidad de dar solución a largo plazo a estas dificultades.

Durante los últimos años, la rama *gvSIG* de la asociación está tratando de realizar una reestructuración íntegra del proyecto. Esta solución no es válida en el contexto de *gvSIG CE* ya que, por una parte la reestructuración no ha tenido, a lo largo de los años, fecha de finalización prevista; y por otra, las elecciones a nivel técnico no son satisfactorias para el equipo de *gvSIG CE*, por motivos que se expondrán a lo largo del artículo.

## SOLUCIÓN PROPUESTA

Ante esta problemática, el equipo de desarrolladores de *gvSIG CE* se plantea de qué manera es posible continuar con el desarrollo de *gvSIG CE* de una manera más sostenible, adaptándose a los nuevos estándares, formatos y nuevas necesidades en general pero con el mínimo esfuerzo.

### **GeoTools**

La apuesta de *gvSIG CE* para un desarrollo más sostenible es *GeoTools*. *GeoTools* es una librería Java de código abierto con licencia LGPL que proporciona herramientas conforme a los estándares definidos por el *Open Geospatial Consortium* (OGC) para la manipulación de información espacial.

Entre las distintas funcionalidades que *GeoTools* aporta podemos destacar las siguientes:

- Topología basada en *Java Topology Suite* (JTS).
- Acceso a los formatos vectoriales y raster más comunes: *shapefile*, *Oracle Spatial*, *PostGis*, *GeoTiff*, etc.
- Soporte para transformaciones entre sistemas de coordenadas.
- Filtros espaciales y alfanuméricos.
- Renderizador con soporte para los estándares de simbolización OGC.
- Soporte para formatos basados en XML/GML
- Implementación de varios servicios OGC: WMS, WCS, WFS, etc.

Como proyecto de software libre, la comunidad de *GeoTools* lleva colaborando desde 2002 [4] y actualmente es posible encontrar documentación sobre las políticas del proyecto así como sobre los distintos procedimientos existentes para la colaboración con el proyecto.

*GeoTools* es usado de forma masiva por multitud de proyectos, como *GeoServer* y *uDig*, y cuenta con un amplio grupo de colaboradores [5]. Además, el equipo de desarrollo de *GeoTools* está comprometido con la calidad del código desarrollado y muestra de ello es el desarrollo de baterías de prueba automatizadas que se ejecutan cada pocas horas en servidores de integración continua.

Todos estos factores son indicadores muy significativos sobre la salud de un proyecto de software libre, tanto en cuanto a la fiabilidad de las funcionalidades que ofrece como en cuanto a la calidad del código desarrollado.

Además, gran parte de la funcionalidad aportada por *GeoTools*, si no toda, solapa con código existente en *gvSIG* que, a diferencia de aquél, no está prácticamente cubierto por casos de prueba automatizados, lo cual dificulta el mantenimiento.

### **VENTAJAS DEL USO DE GEOTOOLS**

Así pues, es posible reemplazar grandes cantidades de código en *gvSIG*, que no puede ser modificado de forma barata porque carece de casos de prueba que validen esas modificaciones (y que por tanto es caro de mantener), por una librería que aporta estas mismas funcionalidades.

Entre las ventajas que esto aporta, está la de abaratar los costes de mantenimiento. Al apoyarse en una librería como *GeoTools*, cualquier nuevo formato o cualquier nuevo estándar implementado por su comunidad podrá ser utilizado inmediatamente en *gvSIG*, con el único coste de actualizar la versión de *GeoTools* utilizada.

En el sentido contrario, es posible que el uso de *GeoTools* saque a la luz algún defecto de *GeoTools*, lo cual representa una oportunidad para colaborar con dicho proyecto, bien con el informe del error o incluso con la propuesta para su corrección. Las correcciones necesarias en *GeoTools* pueden seguir el cauce estipulado para ser aplicadas y sólo es necesario esperar a la publicación de la siguiente versión planificada, que incorporará la corrección.

Desde el punto de vista del desarrollador, el uso de *GeoTools* es muy interesante ya que al colaborar con proyectos de código abierto con tanta visibilidad el trabajo realizado es usado y validado por más personas.

Por último, para el proyecto *gvSIG CE*, el uso de una librería bien conocida y con abundante documentación representa una ventaja de cara a la inclusión de nuevos desarrolladores en el proyecto, ya que actualmente existen multitud de desarrolladores con conocimientos de *Geotools* que podrían aplicar estos conocimientos a desarrollar *plugins* en *gvSIG*.

### **DESVANTAJAS DEL USO DE GEOTOOLS**

Una de las consecuencias de usar un proyecto con tantos usuarios y con varias partes interesadas en el desarrollo es que el proceso de inclusión de parches y modificaciones requiere la aprobación de la comunidad, pasar varios filtros de calidad, etc.

Sin embargo, el equipo de *gvSIG CE* considera este inconveniente muy poco significativo con respecto a las ventajas que *GeoTools* ofrece.

Por otra parte, en el caso de adoptar *GeoTools* finalmente, las líneas de desarrollo de *gvSIG* y *gvSIG CE* se separarán casi por completo por lo que la colaboración será mucho más complicada.

## PROCESO DE INTEGRACIÓN

Uno de los principales problemas a los que el equipo de desarrollo de *gvSIG CE* se ha enfrentado a la hora de integrar *Geotools* es el tamaño inmenso de la base de código de *gvSIG*.

La rama 1.x de *gvSIG*, aquella en la que *gvSIG CE* está basada, tiene cientos de miles de líneas de código, organizadas en varias decenas de proyectos que no sólo no separan completamente la interfaz de la lógica de negocio sino que además están fuertemente acoplados unos con otros. A esto hay que añadir poca o ninguna documentación en el mejor de los casos, documentación engañosa en el peor, código duplicado, nombres de variables poco o nada significativos y más de 18.000 *warnings*.

Sin embargo, el primer intento de integración de *GeoTools* fue engañosamente exitoso. En efecto, en aproximadamente 40 horas de desarrollo fue posible reemplazar la gestión de sistemas de referencia espacial (SRS) existente en *gvSIG CE* por la de *GeoTools*.

En cambio, los siguientes intentos no tuvieron tan buen resultado. En el *codesprint* realizado en Munich en octubre de 2012, varios desarrolladores del proyecto trataron de realizar la misma operación con el cliente de WMS y con la capa de acceso a datos. Sobre todo en el segundo caso, los resultados distaron mucho de los obtenidos con la parte de SRS.

Así pues, se hace patente la necesidad de otra metodología para afrontar la integración. Tras intentar distintas aproximaciones y empezar desde cero cada vez, se acabó por dar con una metodología que:

- Puede ser definida formalmente.
- Puede ser llevada a cabo por más de un desarrollador en paralelo y de forma independiente.
- El resultado de la integración puede verse y ejecutarse en cualquier momento, por lo que se aporta valor desde el principio. Inicialmente el producto no tendrá muchas funcionalidades ya que se irán incorporando poco a poco.

Las dos primeras características son fundamentales de cara a permitir que haya más gente que se implique en el desarrollo, mientras que la última es fundamental para poder evaluar el coste del desarrollo, para poder rentabilizar desde el principio cualquier esfuerzo realizado que realmente aporte valor.

Así, cualquiera podrá ver el progreso de la integración en un momento dado y, aún con la integración no finalizada, es posible que haya usuarios que puedan beneficiarse del estado del programa en un momento dado.

## Metodología

Con el fin de facilitar la colaboración de otros desarrolladores en el proyecto y minimizar los esfuerzos, se ha definido formalmente una metodología de desarrollo para la migración de *gvSIG CE* a *GeoTools* [6]. La idea básica de esta metodología consiste en mantener intacto el sistema de *plugins* de *gvSIG CE* e ir integrando cada una de las extensiones en un único *plugin* de *gvtools* a medida que sea necesario o se considere oportuno.

Así, esta metodología comprende principalmente dos componentes. Por un lado, un componente o librería que contiene todo el código relacionado con *GeoTools* y que proporciona acceso a datos, soporte para leyendas, manejo de capas y mapas, etc. Este componente se ha de implementar por completo y es requisito indispensable que esté bien documentado y testeado.

Y, por otro lado, otro componente que incorpora el código de las extensiones de *gvSIG CE*, copiado tal cual está disponible en el repositorio. Como resultado de copiar las extensiones es de esperar que se produzcan errores de compilación que pueden ser solucionados de diferentes maneras. En función del problema, y siguiendo unas determinadas directrices que se detallan en la descripción completa de la metodología [6], se podrá eliminar el código que contiene el error, copiar más código necesario para la correcta compilación de la extensión, o bien implementar la funcionalidad requerida en la librería de soporte que utiliza *GeoTools*.

Además, la metodología define unas directrices básicas como resultado de la experiencia de migrar las primeras extensiones. Entre estas directrices es posible destacar la consulta de cualquier modificación necesaria sobre la librería de soporte. Esto es debido a que un objetivo primordial de esta metodología es que dicha librería sea coherente, estable y comprensible, por lo que cualquier cambio debe ser lógico, y estar bien documentado y testeado. Por ello, es importante discutir estos cambios entre todo el equipo de desarrollo para llegar a la mejor solución.

Por otra parte, se hace especial hincapié en que el objetivo principal de *gvtools* es la integración de *GeoTools* en *gvSIG CE*. Es por esto que, cuando se copien nuevas extensiones a *gvtools* se ha de evitar siempre que sea posible correcciones sobre el código copiado. Esto es debido a que existe una gran cantidad de casos en los que se verá de manera sencilla cómo se pueden corregir ciertos defectos en el código, por lo que corregir todos estos defectos requeriría una gran cantidad de tiempo que nos alejaría del objetivo principal de integrar toda la funcionalidad lo antes posible.

Por último, es importante destacar que, puesto que es deseable que el número de extensiones se vaya incrementando de manera gradual y constante, se hará necesario generar distribuciones de *gvtools* constantemente. Para esto se ha hecho posible la generación de estas distribuciones con *Maven* en un simple comando, lo cual también hace que la incorporación de nuevos desarrolladores al proyecto sea mucho más sencilla.

Así, como resultado de esta metodología, no sólo es posible reemplazar gran parte del código de *gvSIG CE* por *GeoTools*, sino que el resultado será una librería de soporte para *gvSIG CE* bien diseñada, estable y comprensible para nuevos desarrolladores. Además, el código de las extensiones, proveniente de la versión actual de *gvSIG CE*, estará aislado en un único componente que se podrá revisar y actualizar de manera más sencilla y ordenada. Todo esto contribuye a que la colaboración en el proyecto por parte de otros desarrolladores sea más fácil.

Finalmente, estas extensiones que proporcionan la funcionalidad al usuario se mantendrán prácticamente intactas, por lo que el usuario apenas percibirá ningún cambio en la interfaz.

## ESTADO DEL DESARROLLO

Actualmente la integración se encuentra en un estado muy temprano del desarrollo. Básicamente consiste en un visor de datos espaciales con posibilidad de seleccionar algunas entidades en el mapa.

Sin embargo, como se usa *GeoTools* como librería base, es posible cargar todas las fuentes de datos que ésta es capaz de acceder: *shapefile*, *Oracle spatial*, *PostgreSQL*, etc.

Para cuantificar el estado del desarrollo se han utilizado algunas métricas sobre el código Java de las extensiones de *gvSIG*. Así, se han medido el número total de extensiones en *gvSIG* y se ha comparado con el número de extensiones ya migradas a *gvtools*. Además, con el fin de tener una métrica distinta y poder comparar, también se han medido el número total de líneas físicas (sin tener en cuenta líneas en blanco ni comentarios) de código Java y se ha comparado con el número de líneas existentes en *gvtools*. Los resultados se pueden observar en la Tabla 1.

Tabla 1: Porcentaje de extensiones migradas a *gvtools*

Métrica	gvSIG	gvtools	%
Nº líneas de código	277,781	11.288	4.06
Nº de extensiones	140	7	5.00

En dicha tabla se puede observar como el porcentaje de ambas métricas es bastante similar. Es importante notar que este porcentaje no tiene asociado un coste constante, sino que el coste de migrar las primeras extensiones es mucho mayor que el de las últimas. Esto es debido a que las extensiones requieren de una librería que dé soporte a las operaciones requeridas, como el acceso a datos, el manejo de leyendas, etc. Así, para la primera extensión migrada dicha librería no existe y debe implementarse por completo, mientras que para la última extensión es de esperar que la librería ya esté completamente implementada.

Así, el estado del desarrollo es muy temprano, con apenas un 5% de la funcionalidad integrada. No obstante, por lo que se ha comentado en el párrafo anterior, el porcentaje de trabajo ya realizado es superior al 5%. Además, también es importante destacar que ya es posible ejecutar la aplicación y utilizarla para cargar capas desde fichero, utilizar las herramientas de zoom y realizar selecciones, tal y como se puede apreciar en la Figura 1.

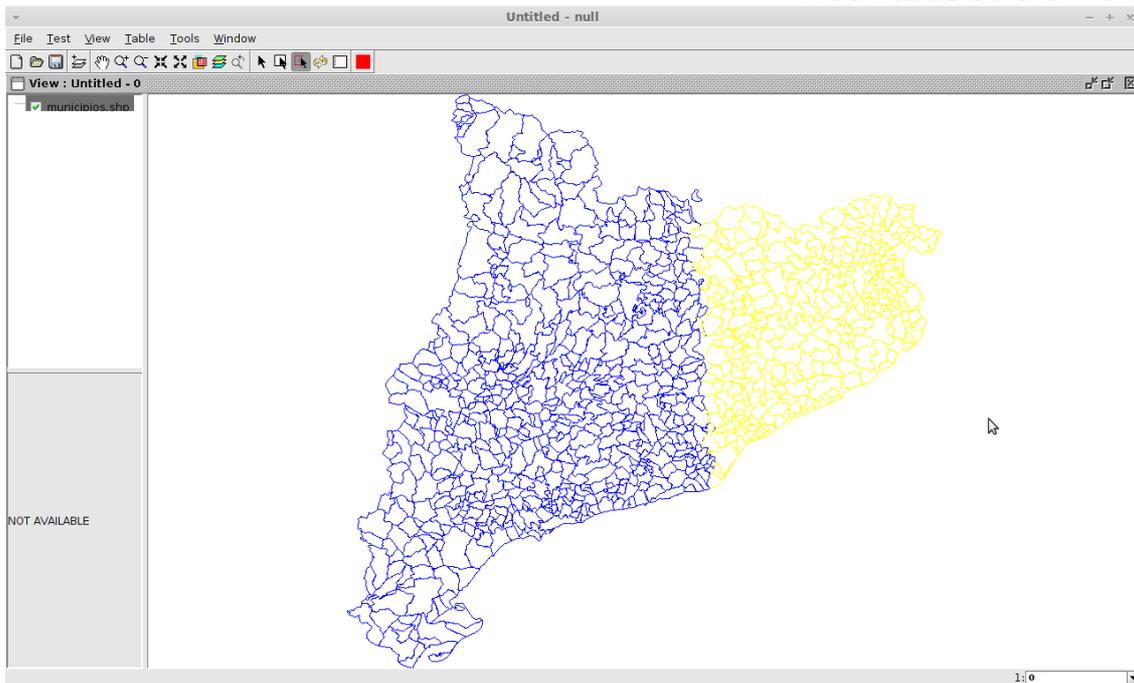


Figura 1: gvtools

## CONCLUSIONES

El desarrollo de la integración con *GeoTools* no tiene una fecha de finalización. Como se expone anteriormente, una de las ventajas de la metodología seguida es que se está aportando algo de valor desde el inicio, aunque sea poco. De hecho, actualmente ya es posible utilizar el resultado de la integración.

Así pues, ¿cuándo podemos considerar la integración con *GeoTools* finalizada? En el equipo de *gvSIG CE* consideramos que dicha pregunta no tiene sentido. La pregunta coherente es ¿Cuándo puede un usuario comenzar a utilizar *gvSIG CE* con *GeoTools*? La respuesta es obvia, cuando incluya las funcionalidades que dicho usuario necesita.

Por tanto, la integración de *GeoTools* se irá haciendo de forma progresiva y se irá desplegando en aquellos usuarios que puedan realizar sus flujos de trabajo con el estado actual de la solución.

En aquellos que necesiten funcionalidades no existentes todavía se desplegará la versión actual, asumiendo las dificultades existentes como se ha hecho desde que surgió *gvSIG CE*.

### El código

El desarrollo de la integración se lleva a cabo en un proyecto en *GitHub* llamado *gvtools* [7]. Es posible encontrar una pequeña introducción [8] y la guía del desarrollador [6].

## REFERENCIAS

- ◆ [1] GONZÁLEZ, F.; SCHÖNBUCHNER, R. y CANALEJO, J. (2012), *Simbology with negative values doesn't work*. gvSIG CE Bug tracker, <http://gvsigce.sourceforge.net/mantis/view.php?id=139>
- ◆ [2] GONZÁLEZ, F.; SCHÖNBUCHNER, R. y CANALEJO, J. (2012), *Automatic loading and applying of vector layer styles*. gvSIG CE Bug tracker, <http://gvsigce.sourceforge.net/mantis/view.php?id=180>
- ◆ [3] GONZÁLEZ, F.; SCHÖNBUCHNER, R. y CANALEJO, J. (2012), *Symbol library*. gvSIG CE Bug tracker, <http://gvsigce.sourceforge.net/mantis/view.php?id=199>
- ◆ [4] *GeoTools*, Wikipedia (2013), <http://en.wikipedia.org/wiki/GeoTools>
- ◆ [5] *Contributors*, The GeoTools Open Source Project on Oloh (2013), <http://www.ohloh.net/p/geotools/contributors/summary>
- ◆ [6] *Developer Guide*, gvtools on Github (2013), [https://github.com/gvtools/gvtools/blob/master/doc/source/developer\\_guide.rst](https://github.com/gvtools/gvtools/blob/master/doc/source/developer_guide.rst)
- ◆ [7] *gvtools*, Github (2013), <https://github.com/gvtools/gvtools>
- ◆ [8] *Introduction*, gvtools on Github (2013), <https://github.com/gvtools/gvtools/blob/master/doc/source/introduction.rst>