

ikiMap

La plataforma social de la cartografía

Alejandro Lamas Perez⁽¹⁾, Ricardo Cal Guance⁽¹⁾, Francisco Sotelo Rua⁽¹⁾, Fernando Gonzalez Cortés⁽²⁾, Victor Gonzalez Cortés⁽²⁾ y Micho Garcia Coya⁽²⁾

⁽¹⁾ Sixtema Edif. Area Central, 1ª planta, L 25-J (Fontiñas) 15707 Santiago de Compostela
info@sixtema.es

⁽²⁾ geomati.co, GIS Freelance Network, www.geomati.co, info@geomati.co

RESUMEN

ikiMap es un servicio web para crear, compartir y publicar mapas en Internet, basada en software libre (como Openlayers o PostGIS).

ikiMap pretende ser no sólo una red social, sino un plataforma de publicación de mapas en la Red a diversos niveles (usuarios, organizaciones, desarrolladores de aplicaciones, etc), siendo a este respecto el feedback y la comunicación con éstos algo imprescindible para la evolución de la plataforma.

Palabras clave: *Redes sociales, compartir mapas, publicar mapas, ikiMap, gvSIG Mini, Android.*

1. INTRODUCCIÓN

ikiMap (www.ikimap.com) es una plataforma gratuita de carácter social para crear, publicar y compartir cartografía en Internet. Desde su apertura al público ikiMap ha ido evolucionando desde un concepto inicial de red social a un concepto más global de plataforma de publicación de mapas en la Red a diferentes niveles (usuarios, organizaciones, desarrolladores, etc), teniéndose en cuenta el *feedback* con la comunidad de usuarios y las tendencias actuales en la Red. Durante todo este tiempo a ikiMap se han ido sumando miles de usuarios registrados, aportando millares de mapas de diversa temática, e ikiMap ha ido aumentando su presencia en la Red y en los medios, intentado llegar a ser un referente en la publicación de mapas en la Web..

2. TECNOLOGÍAS

Para el desarrollo de ikiMap se han ido utilizando diversas tecnologías y software, seleccionándose unas y desechando otras a lo largo del tiempo, teniendo en cuenta las tendencias actuales y de futuro, así como las necesidades específicas de la plataforma, funcionales y de rendimiento. En todo momento se ha intentado

hacer un amplio uso de software de Código Abierto para los desarrollos. A continuación se expone un esquema de la arquitectura de módulos de ikiMap.

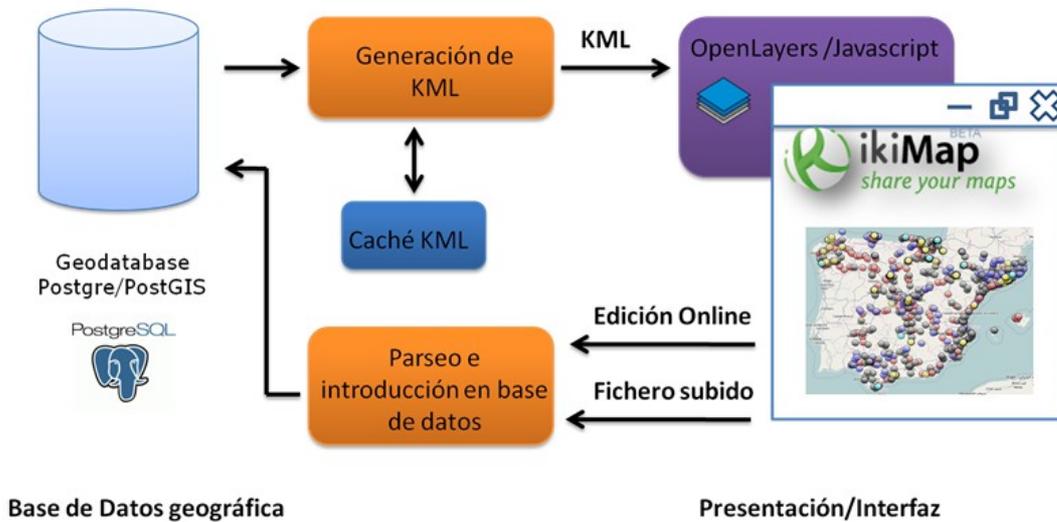


Figura 1: Arquitectura de ikiMap

2.1 BASE DE DATOS POSTGRE/POSTGIS

ikiMap utiliza **PostgreSQL** [1] en combinación con el módulo de soporte datos geográficos **PostGIS** [2], ambos publicados bajo licencia GNU GPL, para la implementación de *geodatabase* del repositorio de mapas de la comunidad de ikiMap, desarrollándose consultas y procedimientos tanto orientados a la explotación de la información cartográfica como a cubrir las necesidades de gestión de usuarios y mapas de la comunidad de ikiMap.

2.2 JAVASCRIPT Y OPENLAYERS

Dado el avance de los navegadores web actuales en cuanto a soporte de Javascript, así como las tendencias actuales de optar por éste sobre otras tecnologías como Flash o Silverlight, se escoge Javascript para la ejecución en cliente. Así mismo, se utiliza tecnología AJAX [3] para la realización de peticiones asíncronas de datos, obteniendo una comunicación cliente-servidor que proporciona una interacción mucho más fluida para el usuario.

Para la presentación e interacción con mapas se utiliza la librería de Javascript **OpenLayers** [4], una librería distribuida bajo licencia BSD que proporciona mapas interactivos con soporte para numerosos servicios de mapas: Google, Bing, Yahoo, Cloudmade, OpenStreetMap [5], etc.

Así mismo, OpenLayers proporciona soporte para la presentación de información en diversos formatos (GML, GeoRSS, GeoJSON, etc), optándose en ikiMap por la utilización de KML para el envío desde servidor.

3. FUNCIONALIDADES

Las posibilidades y opciones que ofrece ikiMap han ido aumentando a lo largo de su vida teniendo en cuenta las necesidades de la comunidad de usuarios y las tendencias actuales en la Red, para cubrir los aspectos de creación y edición de mapas, sin olvidar la componente social.

3.1 MECANISMOS DE CREACIÓN DE MAPAS

Actualmente ikiMap soporta diferentes formatos cartográficos para su importación. Desde el popular KML, el extendido GPX y el preciso y amplio *shapefile*, además de proporcionar herramientas para la edición online de la información cartográfica

- **KML**(Keyhole Markup Language): lenguaje XML utilizado inicialmente por Google Earth para representar datos geográficos en tres dimensiones empleado masivamente en la red para compartir información cartográfica, y que ya en 2008 se convirtió definitivamente en nuevo estándar aprobado por la OGC. Actualmente es uno de los formatos de información cartográficas más popularizados gracias a sus enormes posibilidades y flexibilidad. ikiMap utiliza **libKML** [6], una librería en C++ multiplataforma y de código abierto, específicamente desarrollada por Google, creadores del formato KML. Así, cualquier cambio en el estándar se corresponderá con una actualización de la librería, evitando que el módulo de importación de ikiMap quede obsoleto y permitiendo acceder a nuevas funcionalidades.
- **GPX** (GPS eXchange format): es un formato XML para el intercambio de datos GPS (*waypoints, tracks, routes*) entre aplicaciones ampliamente utilizado. Hoy día es el formato más empleado para el intercambio de información entre dispositivos GPS. Para la incorporación de cartografía en formato GPX, ikiMap utiliza la librería para php **gpxtopgsq** la cual permite extraer la información de ficheros GPX e introducirla en una base de datos de **PostgreSQL**.
- **SHAPEFILE**: es el formato de datos geográficos vectoriales de ESRI. Un shapefile está formado por varios ficheros, siendo los principales los ficheros **.shp** (con las entidades geométricas), **.dbf** (atributos de los objetos) y **.shx** (índice de entidades geométricas). ikiMap utiliza el módulo de PostgreSQL **shp2pgsq** para extraer la información de shapefiles a un formato compatible con PostgreSQL..
- **Edición online**: ikiMap proporciona una interfaz de usuario para la modificación y edición de elementos de mapa, primando en su diseño la usabilidad e utilización intuitiva contra la acumulación de funcionalidades GIS. Así se consigue un editor GIS *on-line* de fácil utilización por parte de usuarios sin conocimientos técnicos, de modo que puedan añadir a su mapa marcadores, líneas, polígonos, etiquetas e imágenes de forma rápida y simple.

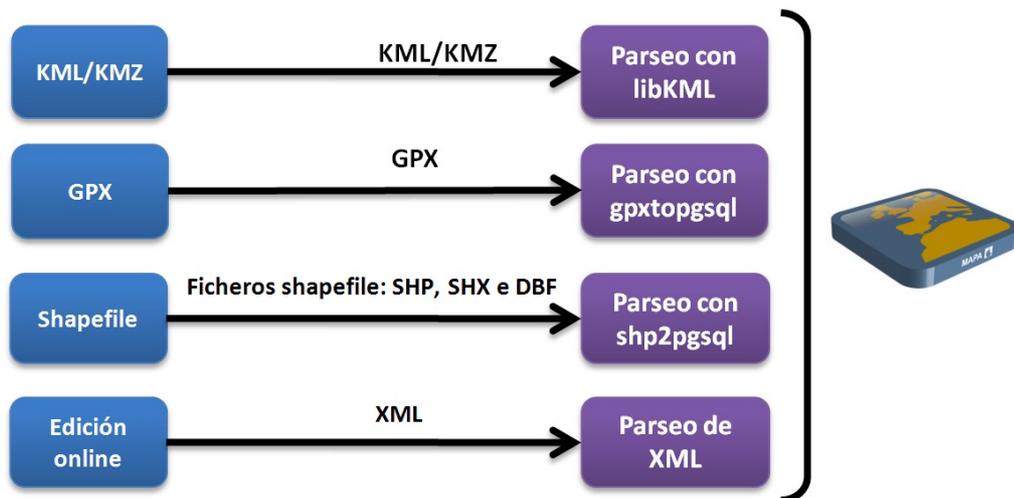


Figura 2: Mecanismos de introducción de datos geométricos

3.2 COMPONENTE SOCIAL

Actualmente es habitual la utilización de redes sociales (Facebook, Twitter, etc), y prácticamente todos los servicios existentes en la Red proporcionan mecanismos de socialización: contactos o amigos, compartir contenidos, votación y comentarios, etc.

Así, ikiMap permite compartir mapas en otras redes sociales (Facebook, Twitter, Google+, etc), votar mapas, hacer comentarios a éstos y marcar mapas como Favoritos. Además ikiMap permite seguir la actividad de otros usuarios que crean contenido de interés para ti, sabiendo cuándo crean nuevos mapas, qué mapas han sido valorados, qué mapas ha hecho favorito o incluso a quien sigue ese usuario.

También es posible elaborar nuevos contenidos sobre contenidos aportados por otros usuarios, ya sea agrupando mapas de temática similar mediante **Canales**, o combinando mapas y creando **ikiMapas** para una visualización conjunta.

4. API DE IKIMAP

ikiMap ha ido evolucionando desde un concepto inicial de red social para compartir mapas a un concepto más global de plataforma de creación y publicación de mapas en la Red a diferentes niveles, desde usuarios sin conocimientos técnicos hasta usuarios con formación GIS y desarrolladores de aplicaciones.

Para ello se ha desarrollado una API de acceso a los servicios de ikiMap. Gracias a estos servicios cualquier desarrollador tiene la posibilidad de utilizar los datos almacenados en ikiMap para sus propias aplicaciones, desde la creación y recuperación de datos geométricos, hasta la obtención de los comentarios que se realizan en un mapa, datos de perfil de usuarios o actividad de los mismos. Mencionar los 6 bloques en donde se aglomeran los métodos de la API de ikiMap:

- Usuarios.

- Mapas.
- ikiMapas.
- Canales.
- Búsquedas.
- Comentarios, favoritos y otros.

La API de ikiMap trabaja mediante estándares de comunicaciones (HTTP GET/POST e XML), sobre arquitectura REST (*Representational State Transfer* ou *Transferencia de Estado Representacional*), de modo que pueda ser utilizado por aplicaciones en diferentes plataformas, ya sean web, de escritorio o para móviles. Para la salida de la información obtenida mediante los métodos se utilizan los formatos JSON o XML según se solicite mediante parámetro, haciendo que el acceso a los datos de ikiMap se realice de una manera sencilla desde cualquier tipo de cliente.

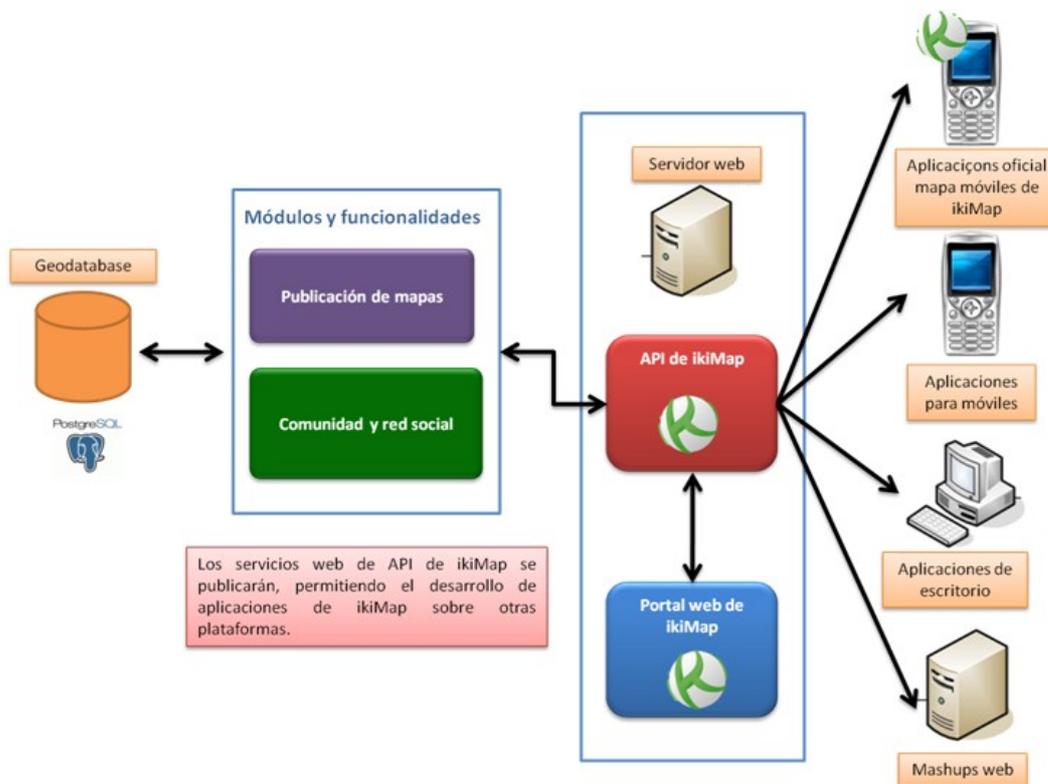


Figura 3: arquitectura de la API de ikiMap.

Los servicios de API utilizan un mecanismo de autenticación, tanto para identificar las aplicaciones que usan la API, como para la autorización de acceso a datos privados de usuario. Por un lado, para la utilización de la API de ikiMap por parte de una aplicación se solicitará una clave de desarrollo que identificará las peticiones de ésta. Y por otra parte, el acceso a datos privados de usuario se requerirá de una acción de *log-in* (autenticación) para garantizar la identidad del usuario accediendo a información protegida.

5. CLIENTE DE IKIMAP PARA ANDROID

Una vez implementada la API, se plantea la idea dentro de la estructura de ikiMap del desarrollo de las versiones oficiales de aplicaciones para móviles. Una de estas

versiones se realizará para el sistema operativo Android. La idea es implementar la funcionalidad del portal web dentro de estos dispositivos móviles, permitiendo al usuario acceder desde ellos a las mismas funcionalidades. Además se pretende realizar un testeo de la API y comprobar que su definición es correcta y permite el desarrollo de aplicaciones por terceras partes.

5.1 Tecnologías

Para el desarrollo de la aplicación para Android[7], se ha utilizado el entorno común de desarrollo de este tipo de aplicaciones, Android SDK con Eclipse[8] y Git[9] como sistema de control de versiones. La versión de la API de Android que se ha utilizado ha sido la 7.

5.2 El uso de OpenLayers como primera opción

En una primera aproximación al proyecto tal y como se puede comprobar dentro de la arquitectura de ikiMap, se definen claramente dos partes. Por un lado todos los servicios que se encargan del módulo de comunidad y red social y por otra parte los encargados de la publicación de mapas. Para el primer módulo, el de comunidad y red social, la API de Android permite la gestión de estos servicios, ya que existen las funcionalidades necesarias para la comunicación mediante peticiones GET, POST y el tratamiento de formato XML o JSON.

Sin embargo para el desarrollo del segundo módulo, el de publicación de mapas y el tratamiento de información geográfica, la API de Android si se presenta excasa, ya que cuenta unicamente con los servicios que proporciona la API de Google Maps para este tipo de dispositivos[10]. En una primera aproximación al problema, se planteó realizar este módulo mediante el uso de la versión de OpenLayers para móviles[11] actualmente en desarrollo. Para esta integración era necesario la insercción de un control web dentro de la aplicación y la comunicación de este mediante llamadas a través de Android a la funcionalidad implementada en JavaScript. Tras algunas pruebas se comprobó que el rendimiento, sin ser desmerecido, no era el adecuado, y además algunas de las funcionalidades no estaban implementadas para este tipo de dispositivos por lo que se descartó seguir avanzando en esa línea de desarrollo.

Llegados a este punto se presentaba la necesidad de obtener unas librerías que cubriesen la funcionalidad de la parte de publicación de mapas, principalmente:

- Presentación de mapas sobre diferente cartografía base, GMaps, Bing, OpenStreetMap... como en el caso del portal web
- Permitir la carga de formato KML dentro del control para la presentación de los mapas de los usuarios.
- Permitir la creación y edición esos mapas de usuario.
- Permitir el manejo de formato GPX extraído del GPS del dispositivo móvil para la creación de mapas.

Revisando el estado actual del arte en cuanto a proyectos que utilizasen librerías que permitiesen realizar estas operaciones se encontró que existían varios proyectos que podían aportar parte de esa funcionalidad, tales como osmdroid[12] o gvSIG Mini[13]. osmdroid dispone de librerías para incluir cartografía de distintos proveedores como Bing Maps, GMaps y como no OpenStreetMaps. Por esa parte cubría las necesidades de presentación de diferentes capas base, pero dejaba sin resolver el resto de temas. gvSIG Mini ofrecía una cobertura mayor de las necesidades del

proyecto ya que además de permitir el manejo de cartografía de diferentes proveedores, se había desarrollado durante el GsoC de 2011 la posibilidad de añadir soporte para capas vectoriales[14]. Teniendo en cuenta la compatibilidad de las licencias de ikiMap con gvSIG Mini, se decidió optar por continuar el desarrollo del módulo de publicación de mapas utilizando las librerías de gvSIG Mini.

5.3 Utilización de librerías de gvSIG Mini en ikiMap para Android

En una primera fase previa a la integración se valoró llevar a cabo una refactorización de las librerías de gvSIG Mini para desacoplar estas lo máximo posible de la aplicación y poder ser así mejor utilizadas por terceras partes. Se ha orientado esta refactorización de tal manera que permitiese un manejo de los controles similares a los ofrecidos por las librerías de OpenLayers, pudiendo crear y añadir controles mediante código facilitando la tarea de desarrollo.

5.3.1 Preparación del entorno

Desde los repositorios de gvSIG Mini[15] es posible descargarse los fuentes necesarios para utilizar el proyecto. Las herramientas que serán necesarias a parte de las ya indicadas en la sección tecnologías son Maven y el plugin de integración de este en Eclipse. Mediante el uso de las diferentes funcionalidades de Maven podremos compilar e instalar lo necesario para poder utilizar dichas librerías en nuestro proyecto ikiMap para Android.

5.3.2 Algunos de los paquetes usados

Dentro del código fuente de gvSIG Mini encontraremos cantidad de paquetes que ofrecen diferente funcionalidad. Principalmente los usados por la aplicación ikiMap para Android son:

- **es.prodevelop.gvsig.mini.map:** donde encontraremos la vista MapView encargada de mostrar los mapas y a la que se le añaden los diferentes controles, listeners, controladores, ... Es la base de desarrollo. Dispone de la funcionalidad necesaria para añadir y quitar controles y capas, controlar los eventos del usuario, presionado de teclas, touchcontrol, gestión del zoom...
- **es.prodevelop.gvsig.mini.control:** paquete con los controles para el uso de la vista, y con una clase abstracta para el desarrollo de controles personalizados.
- **es.prodevelop.gvsig.mini.overlay:** contiene todas las clases referentes a las capas. Contiene a MapOverlay clase de la que heredan el resto de capas como TileOverlay o PointOverlay.

Estos paquetes se encuentran dentro del proyecto gvsig_mini_base que englobaría las librerías del proyecto gvSIG Mini. Otro proyecto importante se trata de TileRasterCache, que engloba las librerías encargadas del manejo de los renderizados de las diferentes capas de los distintos proveedores. Dentro encontraremos dos paquetes importantes para la implementación de la aplicación:

- ***es.prodevelop.tilecache.layers***: que engloba la clase Layers encargada de dar persistencia a la información de las diferentes capas almacenadas en formato texto.
- ***es.prodevelop.tilecache.renderer***: contiene las clases que se encargan del manejo de la información recibida a modo de Tiles de los servidores de los proveedores de cartografía.

Estos paquetes son los encargados del manejo de las imágenes recibidas que mostrarán las diferentes capas base disponibles. Es interesante remarcar la facilidad de la librería para dar de alta nuevos proveedores de cartografía mediante la modificación de un archivo de texto, incluyendo en él los parámetros de las capas de dichos proveedores. Como último proyecto de referencia se nombrará la API GPE que se encarga de la gestión de la información vectorial dentro de gvSIG Mini. Esta contiene la funcionalidad necesaria para el manejo de geometrias en formato KML y GPX necesarias dentro de la aplicación de ikiMap para Android

5.3.3 Desarrollo de la aplicación ikiMap para Android

Actualmente se encuentra totalmente desarrollada la aplicación. Se ha creado una MapActivity que hereda de la clase abstracta AbstractIkiMapActivity e implementa la interfaz MapViewListener de gvSIG Mini. Esta activity dispone de una view, IkiMapView que a su vez hereda de la MapView de gvSIG Mini pero implementando sobre ella métodos que ofrecen funcionalidad diferente adaptados a las condiciones de la nueva aplicación. La gestión de las diferentes capas base, se realiza mediante una ListActivity que implementa una ListView y un flujo típico de navegación entre activities de Android. La detección de la posición actual del dispositivo mediante GPS está implementada utilizando las clases LocationManager y LocationListener de la libreria android.location y GPSPoint del paquete *es.prodevelop.gvsig.mini.geom.android*, como contenedor de la información obtenida del dispositivo.

Fig 4. Ejemplo de implementación de clases en ikiMap para Android

Para el manejo del formato KML en el que se comunica la API con la aplicación, ha sido necesario implementar diferentes funcionalidades que aún no estaban desarrolladas en gvSIG Mini. A partir de componentes como MapOverlay, se han desarrollado capas que permiten la visualización de la información KML manteniendo los estilos de las geometrias. En el caso de ikiMap, la mayor parte de las geometrías de tipo punto tienen asociado un icono (bitmap) que representa a este. Ha habido que implementar un tipo de JTS[16] feature que nos permita el manejo de esta información. Debido a los problemas de memoria que han ido apareciendo durante el desarrollo por el manejo de este tipo de elementos, se ha creado una estructura que permite el cacheado de los elementos del KML y que la aplicación puede manejar en caso de falta de memoria.

5.4 CONCLUSIONES

Como conclusiones en el estado actual del proyecto indicaríamos:

- Siempre es beneficioso el uso de funcionalidades desarrolladas al amparo de otros proyectos para no invertir en desarrollos que pueden ser evitables, no reinventar la rueda.
- Obtener sinergias entre proyectos permite que ambos se beneficien de las inversiones realizadas.
- Como desarrolladores somos conscientes de lo que nos cuesta documentar, y a su vez, de lo importante que es tener esa documentación cuando se quiere que ese desarrollo sea usado por terceras partes.
- La posibilidad de que parte de la funcionalidad de un desarrollo sea mantenido dentro de un proyecto de gran calado.
- El desarrollo de aplicaciones para terminales móviles implica un control más exhaustivo de la gestión de los recursos. Cualquier pequeña pérdida de eficiencia que no tengamos en cuenta puede generar problemas en el funcionamiento de la aplicación
- Dentro del entorno de Android, existen ya varias soluciones para el manejo de cartografía, como las expuestas en este artículo anteriormente, sin embargo ciertos temas aún están poco desarrollados, como la edición vectorial dentro de terminales móviles. En ese aspecto el desarrollo de ikiMap aporta ya un punto de partida.

6. REFERENCIAS

1. Portal Web de PostgreSQL: <http://www.postgresql.org/>
2. Portal Web de PostGIS: <http://postgis.refractory.net/>
3. Entrada sobre AJAX en la wikipedia: <http://es.wikipedia.org/wiki/AJAX>
4. Portal Web de OpenLayers: <http://openlayers.org>
5. Portal Web de OpenStreetMap: <http://www.openstreetmap.es/>
6. Referencia a libkml el Portal Web de Google:
<http://code.google.com/p/libkml/>
7. Portal web de Android <http://www.android.com/>
8. Portal web de Eclipse <http://www.eclipse.org/>
9. Portal web de Git <http://git-scm.com/>
10. Google Projects for Android: Google APIs
<http://code.google.com/android/add-ons/google-apis/maps-overview.html>
11. Development Environment for OL Mobile
<http://trac.osgeo.org/openlayers/wiki/mobile>
12. Google Code project osmdroid <http://code.google.com/p/osmdroid/>
13. Portal gvSIG Mini <https://confluence.prodevelop.es/display/GVMN/Home>
14. Add support to vector data formats gvSIG Mini GsoC 2011
<https://confluence.prodevelop.es/display/GVMN/Add+support+to+vector+data+formats>
15. How to build the gvSIG Mini workspace in Eclipse
<https://confluence.prodevelop.es/display/GVMN/How+to+build+the+gvSIG+Mini+workspace+in+Eclipse>
16. JTS Topology Suite <http://sourceforge.net/projects/jts-topo-suite/>