



Universidad
Zaragoza

Proyecto Fin de Carrera

Análisis de servicios de Web Mapping y su
integración en componentes de visualización de
mapas genéricos

Autor/es

Carlos Salvador Rabaza Bergua

Director/es y/o ponente

Director: Juan López de Larrínzar Galdámez

Ponente: Pedro R. Muro Medrano

Escuela de Ingeniería y Arquitectura
2013/2014

Agradecimientos:

Quiero agradecer a mis profesores, compañeros, amigos y familia todo el apoyo y paciencia que han tenido conmigo durante el desarrollo de este proyecto.

También quiero expresar mi agradecimiento al personal del IAAA y GeospatiumLab, en especial a Iván y Juan que me han guiado y apoyado a lo largo de todo este proyecto.

Análisis de servicios de Web Mapping y su integración en componentes de visualización de mapas genéricos

Resumen

Este PFC ha sido desarrollado por Carlos Rabaza Bergua en colaboración con el IAAA y GeosLab como un proyecto de investigación y desarrollo con una duración de más de 1561 horas. El proyecto ha estado motivado por el crecimiento en el uso de información cartográfica por parte de los usuarios así como en el incremento de la oferta de diferentes proveedores cada uno con características que lo hacen único dentro de del panorama mundial. El PFC se ha dividido en dos bloques: parte web y parte móvil.

Para la parte web se han analizado diversos proveedores de mapas online, así como sus posibilidades de integración en la librería de referencia en cuanto a visores de mapas, OpenLayers (en concreto su versión 2.12 que era la última disponible en el momento de inicio de este PFC). Tras el análisis de proveedores realizado, se seleccionaron dos (Yandex y Baidu) por presentar cualidades interesantes y que los hacían diferentes del resto. Durante la integración surgieron múltiples dificultades, siendo el caso de Baidu el más problemático y que dio lugar a la creación y presentación de un paper al respecto en las Jornadas Ibéricas de Infraestructuras de Datos Espaciales del 2013 celebradas en Toledo. Una vez finalizada la integración, se añadió al trabajo realizado un control de GeosLab para la generación de rutas y se creó otro control para la exportación de la ruta en formato JSON a otro dispositivo permitiendo así la interoperabilidad entre las dos partes de este sistema que se ha creado.

En cuanto a la parte móvil, se realizó el análisis de varias librerías y aplicaciones de mapas de tipo Open Source. En concreto el análisis se centró en el análisis de características que permitiesen el uso offline de mapas así como la permisividad de la licencia a la hora del uso dentro de otra aplicación. La licencia era también un factor importante ya que el objetivo de este bloque consiste en la integración de mapas offline en una librería de GeosLab S.L. para Android. Tras el análisis realizado, se escogió la librería Mapsforge que ofrecía una licencia de tipo LGPLv3 lo que posibilitaría su uso dentro de aplicaciones privativas. Además de esto, la librería de Mapsforge cuenta con mapas para todo el mundo en un formato propio usando la información de OpenStreetMap y que tienen un tamaño reducido al usar información vectorial. Utilizando la información recogida, se realizó la integración de la funcionalidad offline de Mapsforge en la librería de GeosLab para Android, así como también se le añadió soporte para teselas online de servicios como OpenStreetMap. Por último, la aplicación creada para el caso de uso cuenta con soporte para localización vía GPS y redes. Además puede cargar un fichero de tipo JSON con una ruta que se ha podido generar con la parte Web del proyecto de modo que utilizando el GPS, un mapa de Mapsforge offline y la ruta, el usuario pueda desplazarse si necesidad de usar ningún tipo de conectividad a Internet.

Índice

1. Introducción	9
1.1. Contexto profesional	9
1.2. Contexto tecnológico	9
1.2.1. Sistemas de información geográfica	9
1.2.2. Contexto tecnológico de la parte web	10
1.2.3. Contexto tecnológico de la parte móvil	11
1.3. Motivación y objetivos del proyecto	11
1.4. Estructura de la memoria	12
2. Trabajo realizado	13
2.1. Parte web	13
2.1.1. Estudio de proveedores de mapas online	13
2.1.2. Análisis de OpenLayers 2.12	14
2.1.3. Arquitectura de la integración	14
2.1.4. Caso de uso	18
2.2. Parte móvil	20
2.2.1. Estudio de librerías de mapas offline para Android	20
2.2.2. Análisis de Mapsforge	20
2.2.3. Arquitectura de la integración	22
2.2.4. Caso de uso	25
3. Conclusiones	27
3.1. Cumplimiento de objetivos	27
3.2. Líneas futuras	27
3.3. Valoración personal	28
4. Anexos	29
Anexo A. Introducción a los sistemas de información geográfica	29
Anexo B. Análisis de proveedores de mapas	32
Bing Maps	32
Google Maps	34
Mappy	37
MapQuest	39
NearMap	44
Here Maps (Nokia Maps)	46

OpenStreetMap.....	49
ViaMichelin	51
WikiMapia.....	52
Yahoo! Maps	53
Yandex Maps	54
Baidu Maps.....	57
Anexo C. Análisis de OpenLayers 2.12	59
1. ¿Qué es OpenLayers?.....	59
2. ¿Cómo funciona OpenLayers?.....	60
3. Vías para la integración de proveedores	68
4. Diseño escogido para la integración	70
5. Diagrama de clases de la integración.....	72
Anexo D. Problemática integración Yandex.....	75
1. Introducción.....	75
2. Problemática para la integración de Yandex.....	75
3. Integración de Yandex en OpenLayers	77
Anexo E. Problemática integración Baidu	83
1. Introducción.....	83
2. Problemática para la integración de Baidu	84
3. Integración de Baidu en OpenLayers.....	88
Anexo F. Restricciones al trabajo con información geográfica online en China (Paper JIIDE 2013).....	95
1. Introducción	95
2. Restricciones legales en China.....	96
3. Restricciones de los mapas en China	96
4. Conclusiones	106
5. Agradecimientos	107
6. Referencias	107
Anexo G. Baidu Maps API interfaz conversión coordenadas	108
Anexo H. Análisis de librerías de mapas offline	111
Big Planet Tracks.....	111
AndNav2	111
Osmdroid	112
Mapsforge.....	112

OsmAnd (OSM Automated Navigation Directions)	113
Conclusiones.....	114
Anexo I. Análisis de Mapsforge.....	115
1. ¿Qué es Mapsforge?	115
2. ¿Cómo funciona Mapsforge?	115
3. Vías para la integración en la librería Android de GeosLab.....	121
4. Diseño escogido para la integración	122
Anexo J. Casos de uso.....	130
1. Parte Web	130
2. Parte Móvil.....	138
Anexo K. Resumen de horas del PFC	145
5. Bibliografía.....	150
6. Glosario	151

1. Introducción

El presente documento recopila toda la información de interés del proyecto final de carrera (PFC) con título "Análisis de servicios de Web Mapping y su integración en componentes de visualización de mapas genéricos" desarrollado por Carlos Rabaza Bergua. Esta introducción se divide en varias secciones. En primer lugar se va a ver el contexto profesional y tecnológico del proyecto así como la motivación y sus objetivos para finalmente acabar revisando la estructura que va a tener este documento. El proyecto ha tenido una duración de más de 1563 horas debido a las dificultades y problemas que se han ido encontrando durante el desarrollo.

1.1. Contexto profesional

El presente proyecto fin de carrera ha sido desarrollado en colaboración con el Grupo de Sistemas de Información Avanzados (IAAA) del Departamento de Informática e Ingeniería de Sistemas (DIIS) perteneciente a la Universidad de Zaragoza y también en colaboración con GeoSpatiumLab S.L. (GeosLab).

El IAAA es un grupo con 20 años de experiencia en I + D, adscrito al Instituto de Investigación en Ingeniería de Aragón de la Universidad de Zaragoza. Una de sus áreas de trabajo son las llamadas Infraestructuras de Datos Espaciales (IDEs).

En cuanto a GeosLab es una empresa que nace en el 2.007 con el objetivo de dar uso a la tecnología desarrollada por parte del IAAA. Se trata de una empresa especializada en el tratamiento digital de la información geoespacial y georreferenciada y sus ámbitos de aplicación. Cuenta con una amplia experiencia en el ámbito de las Infraestructuras de Datos Espaciales (IDE), siendo éstas, sistemas de información integrados por un conjunto de recursos que permite el acceso y la gestión de datos y servicios geográficos disponibles en Internet y que cumplen una serie normas, estándares y especificaciones que regulan y garantizan la interoperabilidad de la información geográfica. Posee como una de sus líneas de negocio el desarrollo de soluciones que facilitan el tratamiento y la visualización de datos de base geográfica tanto en entorno Web como móvil, los denominados sistemas de información geográfica (SIG), así como servicios basados en localización.

1.2. Contexto tecnológico

El proyecto se ha dividido en dos partes diferenciadas, cada una con su propio contexto tecnológico, pero ambas se encuentran basadas en los Sistemas de Información Geográfica. Esta diferenciación se debe al diferente ámbito de uso. La primera parte del proyecto se centra en el uso de tecnologías Web relacionadas con los Sistemas de Información Geográfica, mientras que para la segunda se ha pasado a un contexto de tecnologías móviles, más concretamente Android.

1.2.1. Sistemas de información geográfica

Los Sistemas de Información Geográfica (SIG) permiten mantener y usar datos relacionados con localizaciones geográficas. El tipo de datos que manejan estos sistemas pueden ser imágenes aéreas o de satélite siempre y

cuando se pueda determinar una localización geográfica sobre ellos, planos de ciudades o mapas de carreteras digitalizados, datos relacionados con ubicaciones geográficas como pueden ser tipos de suelos, precipitaciones medias, temperaturas...

Habitualmente un SIG estructura la información en forma de capas superpuestas que aportan información adicional. Actualmente este tipo de estructura se usa en prácticamente todos los servicios de mapas Web que podemos encontrar permitiendo al usuario usar unas capas (ver Figura 1) u otras según la información que desee ver. Estas capas suelen ser de dos tipos: ráster o vectoriales. Las capas ráster ya están dibujadas previamente por lo que se devuelve ya la imagen directamente, sin embargo, en las capas vectoriales se dispone de la información de geometrías para hacer el dibujo y por tanto es necesario realizar el dibujo antes de mostrárselo al usuario. Además, las capas se dividen habitualmente en una cuadrícula de modo que la información se pueda devolver en bloques según sea necesario. Cada elemento de esta cuadrícula se conoce habitualmente como tesela. Se ha adjuntado más información a este respecto que se puede consultar en el Anexo A.

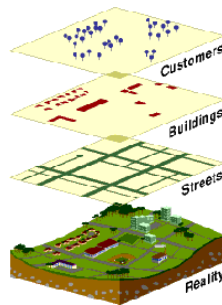


Figura 1 - Ejemplo de estructura en capas de un SIG

1.2.2. Contexto tecnológico de la parte web

Para la parte Web del proyecto se han usado varias tecnologías, entre ellas cabe destacar:

a) JavaScript

Es un lenguaje de programación interpretado (no necesita compilación). Muchos de los navegadores web actuales cuentan con un motor capaz de interpretar código en JavaScript permitiendo a las páginas Web ejecutar código JavaScript en el ordenador del usuario pudiendo crear así funcionalidad sin tener que recurrir al servidor.

b) HTML

Es el principal lenguaje de marcas usado para crear páginas Web y otra información que pueda ser mostrada en un navegador Web.

c) CSS

Las hojas de estilo CSS permiten describir el aspecto que tendrá una página Web al ser mostrada en un navegador. Habitualmente se usa para dar formato a páginas web escritas en HTML o XHTML aunque también puede dar formato a documentos en XML, SVG y XUL.

d) OpenLayers

OpenLayers es una librería JavaScript de código abierto (usa una licencia derivada de BSD) usada para mostrar datos de mapas en navegadores web. La librería permite crear aplicaciones web similares a

las ofrecidas por Google Maps, o Bing Maps. Fue desarrollado originalmente por Metacarta Labs, aunque desde el 2007 es un proyecto de la Open Source Geospatial Foundation.

1.2.3. Contexto tecnológico de la parte móvil

Para la parte Web del proyecto se han usado varias tecnologías, entre ellas cabe destacar:

a) Android

Es un sistema operativo principalmente para móviles que ha cogido fuerza en los últimos tiempos. Actualmente Android supone en torno a un 70-80% de los dispositivos móviles a nivel mundial mientras que iOS su principal competidor supone en torno a un 20% del mercado mundial.

b) Java

Es un lenguaje de programación orientado a objetos pensado para funcionar sobre una máquina virtual de manera que el programador escriba una vez el código y se pueda ejecutar en cualquier dispositivo para el que exista una máquina virtual de Java. En el caso de Android el código es compilado en un archivo APK que es ejecutado por la máquina virtual propia de Android llamada Dalvik.

1.3. Motivación y objetivos del proyecto

Hoy en día son muchas las personas que utilizan servicios de mapas en su día a día, sobre todo en dispositivos móviles haciendo uso del GPS integrado. Sin embargo, hay gente que necesita recurrir al acceso a mapas sin conexión bien por no tener contratada una tarifa de datos o bien por no disponer de conexión en la zona en la que se encuentre. Los proveedores de mapas son muy heterogéneos ya que aunque algunos cuentan con características comunes entre ellos, también cuentan con servicios adicionales que los hacen únicos (capas de tráfico, de incidencias, ubicación de comercios...).

El PFC se ha dividido en dos partes diferenciadas. Para la primera parte del proyecto los objetivos han sido la realización de un análisis de diferentes servicios de Webmapping así como una valoración de sus posibilidades de explotación y de sus características especiales para su posterior integración en la librería OpenSource de referencia en visores de mapas OpenLayers en su versión 2.12 que era la última en el momento de inicio de este PFC.

En cuanto a la segunda parte del proyecto, los objetivos han sido el análisis de proveedores de mapas con características offline para su posterior integración en la librería Android genérica de GeosLab de visualización de mapas.

Se ha creado un sistema formado por un caso de uso de cada parte y que están diseñados para que puedan trabajar en conjunto. También cabe señalar que este sistema no es un producto final si no que se trata de un prototipo que se pueda usar como base para futuros proyectos.

Actividades correspondientes a la primera parte (parte Web):

- Formación en Sistemas de Información Geográfica y en tecnologías necesarias para el proyecto
- Estudio y análisis de los diferentes proveedores de mapas disponibles.

- Estudio y análisis de la librería OpenLayers.
- Integración en OpenLayers de los proveedores de mapas escogidos tras el estudio previo.
- Creación del caso de uso de la parte Web.

Actividades correspondientes a la segunda fase (parte móvil con Android):

- Formación en la tecnología Android.
- Estudio y análisis de proveedores de mapas orientados a dispositivos móviles.
- Integración de los proveedores de mapas elegidos en una librería genérica de visualización de mapas desarrollada por GeoSpatiumLab.
- Creación del caso de uso de la parte móvil con Android.

Actividades correspondientes a ambas fases:

- Análisis y diseño del caso de uso prototipo a partir del conocimiento adquirido en las fases anteriores.
- Implementación del caso de uso diseñado que integre los desarrollos realizados en las fases anteriores.

1.4. Estructura de la memoria

En este apartado se va a revisar la organización de este documento. Al inicio de este documento se puede encontrar una página resumen de todo el proyecto. En cuanto a la organización, este documento se divide en varias secciones: Introducción, Trabajo realizado, Conclusiones, Anexos, Bibliografía y Glosario.

1. Introducción: se hace un repaso del contexto profesional y tecnológico en el que se ha desarrollado este proyecto. A continuación se analiza la motivación y los objetivos del proyecto para acabar finalmente con la sección en la que se analiza la estructura de la memoria.
2. Trabajo realizado: esta es la sección central de la memoria. Como ya se ha mencionado el proyecto está dividido en dos partes diferenciadas por lo que esa separación se va hacer presente a lo largo del documento. Por ello, esta sección tiene una primera separación en parte web y parte móvil. La parte Web resume el trabajo realizado con la librería OpenLayers 2.12 y la integración de los proveedores de mapas Yandex y Baidu en ella. La parte móvil resume el trabajo realizado con la librería de GeosLab para Android y la integración de la librería Mapsforge en ella para dar soporte a mapas offline de OpenStreetMap.
3. Conclusiones: en esta sección se hace una revisión del trabajo realizado, incluyendo una valoración personal del trabajo realizado así como el nivel de consecución de los objetivos del proyecto. Además se incluye una valoración de las líneas de futuro del trabajo desarrollado en este proyecto.
4. Anexos: esta sección incluye todos los documentos que contengan información relevante respecto al trabajo realizado y que no se ha incluido en la memoria por limitación de espacio.
5. Bibliografía: referencias bibliográficas consultas para la realización del proyecto.
6. Glosario: incluye una referencia de términos usados en este documento y una descripción de ellos para facilitar la comprensión de los mismos.

2. Trabajo realizado

En esta sección se va a mostrar un resumen del trabajo realizado durante el desarrollo de este proyecto. El trabajo se ha dividido en dos bloques, un primer bloque correspondiente a la parte Web del proyecto y su caso de uso y un segundo bloque correspondiente a la parte de desarrollo móvil en Android y su caso de uso.

2.1. Parte web

Esta primera parte consistió en la modificación de la librería OpenLayers 2.12 para la integración en ella de proveedores de mapas adicionales. Para llevar esto a cabo se comenzó realizando un estudio de posibles proveedores a integrar. Una vez realizado el estudio se pasó a analizar la librería de OpenLayers 2.12 para evaluar las posibles vías de integración. Con esta información se procedió a realizar un análisis técnico de los proveedores que se han escogido como candidatos y su integración en la librería. Por último se construyó un caso de uso sobre el trabajo realizado.

2.1.1. Estudio de proveedores de mapas online

Se han analizado varios proveedores (Anexo B.) para evaluar sus posibilidades de integración con OpenLayers, los proveedores analizados son: Bing Maps, Google Maps, Mappy, MapQuest, NearMap, Here Maps (antiguos Nokia Maps), OpenStreetMap, ViaMichelin, WikiMapia, Yahoo! Maps, Yandex Maps y Baidu Maps.

De todos los proveedores analizados, los que han resultado más interesantes por sus características singulares han sido Yandex Maps y Baidu Maps ya que ambos nos ofrecen algunas características interesantes y especiales y que aportando cierta novedad respecto a otros servicios. Ambos proveedores de mapas además representan la competencia directa de Google en sus respectivos países.

Yandex es una compañía con sede en Moscú, Rusia, y que posee el mayor motor de búsquedas en Internet de Rusia. Como se puede ver en la Figura 2, Yandex lidera el tráfico de buscadores en Rusia con un 61.8% del tráfico de buscadores respecto a Google con un 26.6%.

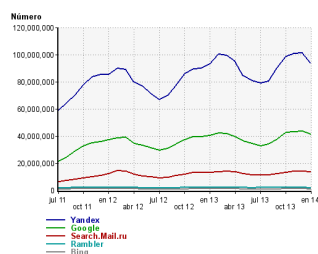


Figura 2 - Tráfico de Yandex en Rusia (Fuente: www.liveinternet.ru)

Baidu (fundada en 1999) es uno de los mayores motores de búsqueda de China. Entre otros servicios dispone de un servicio de mapas. Su motor de búsquedas ocupa la quinta posición a nivel mundial y la primera en China.

Comparando el tráfico de Baidu en China con Google China el tráfico de Baidu es muy superior.

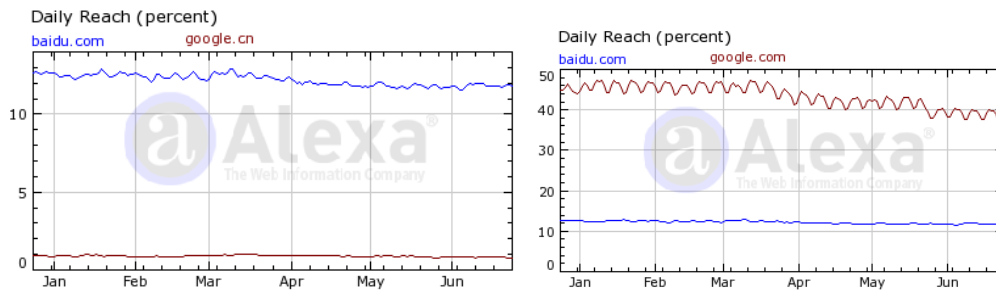


Figura 3 - A la izquierda comparación Baidu y google.cn, a la derecha comparación de Baidu con google.com

2.1.2. Análisis de OpenLayers 2.12

En OpenLayers todos los proveedores se añaden al mapa en forma de capas. Cada una de estas capas se crean usando o bien una clase que gestione un tipo de servicio concreto (como el estándar WMS), o bien una clase que gestione las peculiaridades de un proveedor concreto (como la clase Google, Bing...). A cada clase se le han de pasar además una serie de parámetros para configurarla con la información necesaria.

El diagrama de clases mostrado en la Figura 4 ilustra las interrelaciones entre las diferentes clases de los proveedores. Como se puede ver, todas las clases descienden de la clase Layer, hasta llegar a las clases específicas para proveedores concretos como OSM (OpenStreetMap), Bing y Google. Y también están disponibles algunas clases concretas que pueden ser usadas directamente con múltiples proveedores como pueden ser las capas de tipo WMS (WebMapServer).

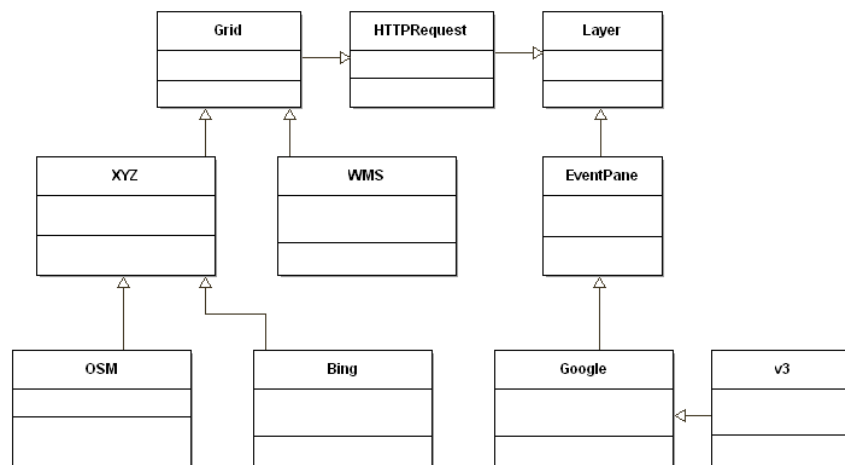


Figura 4 - Diagrama estructura básica OpenLayers 2.12

2.1.3. Arquitectura de la integración

La arquitectura implementada ha consistido principalmente en crear dos clases Baidu y Yandex. Estas dos clases extienden de la clase XYZ de OpenLayers y

tienen sus adaptaciones para su correcto funcionamiento. Para poder hacer uso de esta clase se necesitan varios parámetros. En primer lugar se necesitan las URLs de las teselas que se van a usar. Además de esto, es necesario conocer la proyección en la que se encuentra el mapa, la máxima extensión de la capa y la resolución de la misma. Con estos parámetros ya es posible establecer la relación entre las coordenadas en las imágenes y el mundo que representan.

En cuanto a las URL de las teselas, ni Yandex ni Baidu facilitan información oficial a este respecto, por lo que se analizaron las peticiones realizadas por las API de cada uno de ellos y así obtener el formato de las URL que usan. El análisis se realizó usando el plugin para Firefox "Tamper Data".

Según la información oficial de Yandex, utilizan en sus mapas la proyección Mercator esférica usando como radio de la Tierra 6378137 metros. Sin embargo en ningún lugar identifican la proyección utilizando su correspondiente código EPSG. Tras buscar información no oficial se pudo determinar que la proyección que utilizan es EPSG:3395. Dato que se trata de información no oficial se procedió a contrastarla realizando la conversión de coordenadas por dos vías, por un lado se le pasó a la librería de OpenLayers el código EPSG:3395 y se comparó el resultado con el obtenido usando las fórmulas encontradas en el documento de SNYDER, JOHN P., *Map Projections – A Working Manual*, U.S. Geological Survey Professional Paper 1395, pp 38-47 que se puede consultar en el siguiente enlace de publicaciones del USGS <http://pubs.er.usgs.gov/publication/pp1395>. Concretamente las fórmulas que se encuentran en las páginas 44 y 45 referentes al elipsoide de la proyección Mercator. La extensión máxima del mapa se puede calcular usando la fórmula $Extensión\ del\ mundo = 2\pi * radio$ y la resolución máxima como $(Extensión\ del\ mundo) / (pixels\ del\ lado\ tesela)$.

En cuanto a Baidu, ha sido imposible lograr una integración perfecta, debido a que por motivos de la legislación China, los proveedores de mapas están obligados a incluir un desplazamiento no lineal en sus mapas. Se puede leer más al respecto en el Anexo F. que corresponde al paper presentado en las JIIDE 2013 y que fue redactado por Carlos Rabaza Bergua (autor de este PFC) en colaboración con personal del IAAA y de GeosLab a raíz de los problemas encontrados al trabajar con los mapas en China. Además en el caso de Baidu, añade un desplazamiento adicional al obligado por la legislación China. A este desplazamiento se le conoce también como encriptación geográfica. La Figura 5 ilustra el desplazamiento entre capas de diferentes servicios en China comparados con la capa de satélite de Google. Como se menciona en el paper, las únicas capas que parecen no presentar desplazamientos, son las capas de satélite los servicios de Google y Bing y también OpenStreetMap. Es decir, carecen del desplazamiento aquellos elementos que no dependen del gobierno chino o de su legislación.

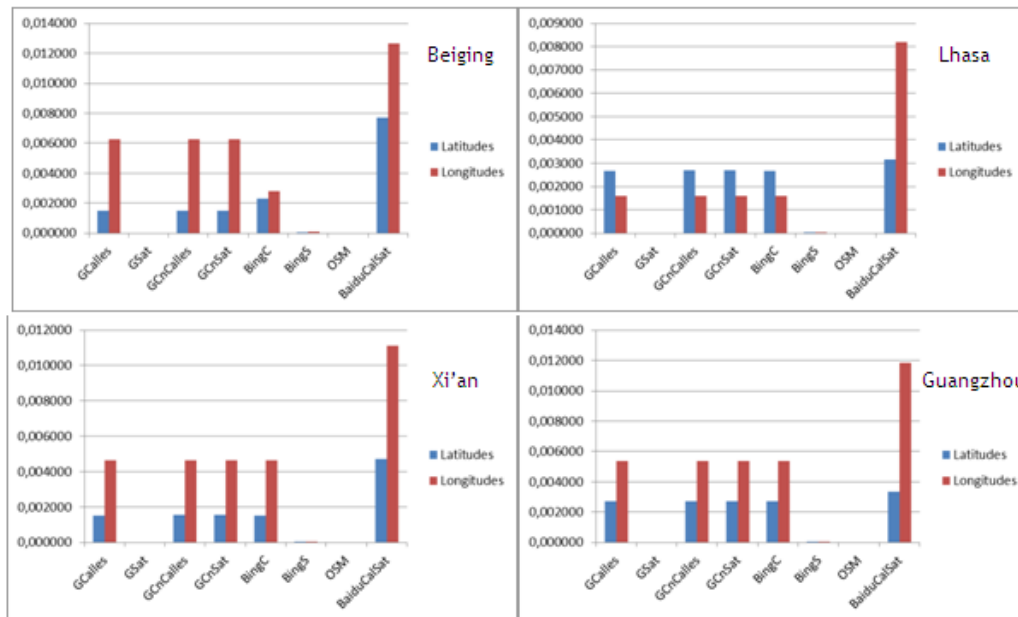


Figura 5- Gráficas de los desplazamientos de cuatro ubicaciones diferentes en China respecto a la capa de satélite de Google. Gcalles = Google Streets; GSat =Google Satélite; GCnCalle = Google China Calles; GCnSat = Google China Satélite; BingC = Bing Calles; BingS = Bing Satélite; OSM = OpenStreetMap; BaiduCalSat = Baidu Calles y Satélite

Para la integración de Baidu, ha sido necesario en primer lugar realizar ingeniería inversa a su librería para obtener las fórmulas que usan para convertir coordenadas del mapa a coordenadas del mundo, así como para obtener los valores de la extensión máxima y de la resolución del mapa. Con esta información se pudo construir un prototipo y dado que no se podía comparar con capas de otros servicios para validar la integración se procedió a realizar la integración de la API de Baidu, para de esta manera poder comparar la capa de Baidu, con la capa de la API de Baidu. De esta manera se pudo comprobar que ambas capas coincidían perfectamente, por lo que se pudo dar por válida la integración.

Además de las capas básicas de ambos servicios (Yandex y Baidu), se integró también el servicio de tráfico de Yandex y el servicio de Alertas de Baidu. El servicio de tráfico de Yandex dispone de información del nivel de congestión del tráfico en ciudades de Rusia y algunos países de sus alrededores. La información general de la congestión del tráfico se muestra en niveles lejanos de zoom, mientras que para niveles más cercanos dispone de teselas con transparencias que muestran la congestión por calles dentro de la ciudad. Además, el servicio de Yandex cuenta con una capa que muestra información de las velocidades en las calles, pensada para funcionar en conjunto con la capa de teselas de tráfico, de modo que al pasar el ratón sobre una calle aparezca la velocidad en ese tramo, esto requirió bastantes optimizaciones para hacerlo usable, debido a la diferente forma de trabajar de OpenLayers y la API de Yandex (Anexo D.). En cuanto al servicio de Baidu, muestra un icono en el que al hacer clic sobre él muestra información adicional sobre el suceso ocurrido. En caso de que no haya ningún suceso, no se muestra nada, y de haberlo puede tener también una imagen para ilustrarlo (Anexo E.).

2.1.4. Caso de uso

Como ya se ha comentado se ha creado un sistema compuesto por dos casos de uso que pueden trabajar en conjunto. En concreto, la funcionalidad offline ha sido uno de los puntos en qué más se ha centrado este PFC. Por ello este caso de uso de la parte Web está pensado para que utilizando las funcionalidades de tráfico de Yandex o las alertas de Baidu se puedan crear rutas online que puedan ser usadas posteriormente en el dispositivo móvil sin ningún tipo de conectividad.

Para ello se ha realizado la integración en OpenLayers 2.12 de un componente de creación de rutas de GeosLab que utilizan en su librería de producción. Este componente permite la creación de una ruta utilizando la API de Google Maps para ello. Una vez realizado esto, se creó un control para permitir la exportación de esa ruta en un fichero de modo que pueda ser usada en la segunda parte (parte móvil del proyecto). Para este fichero se eligió usar el formato de archivo JSON, ya que es un formato habitual en tecnologías web y permite el intercambio de información entre diferentes plataformas por tratarse de un fichero en texto plano que puede ser procesado por la plataforma de destino.

En la siguiente imagen (Figura 7) se puede ver una muestra del uso de las capas de Yandex integradas en la librería. En este caso se muestran capas combinadas de diferentes servicios. Además en la parte superior izquierda de la imagen se pueden ver los controles que permiten la creación de las rutas tal y como se ha mencionado. En la Figura 8 se puede ver la información mostrada en ciudades que disponen de un semáforo con niveles de congestión y las que no. Todas las ciudades que tienen un icono de semáforo sobre ellas disponen del servicio de tráfico de Yandex.

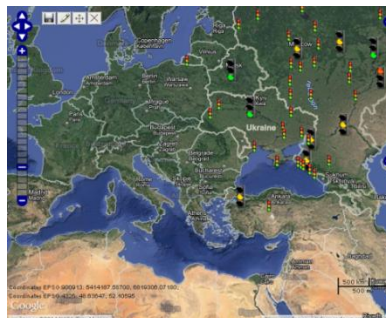


Figura 7 - Captura de pantalla de la aplicación web mostrando como capa base la capa de satélite de Google en conjunción con la capa overlay de nombres de Yandex y la capa de tráfico de Yandex.

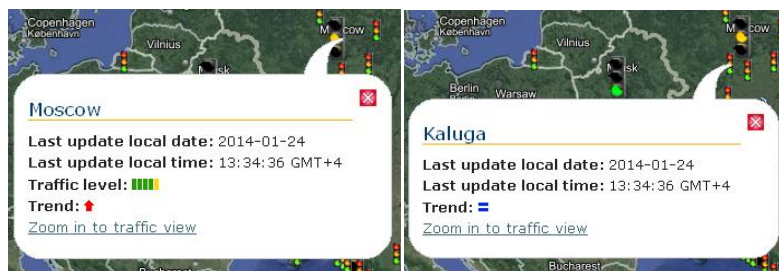


Figura 8 - La imagen de la izquierda muestra la información disponible al hacer clic sobre una ciudad con semáforo con código de colores. La imagen de la derecha

muestra la información disponible al hacer clic sobre una ciudad que no dispone de información sobre el nivel del tráfico.



Figura 9–La imagen de la izquierda muestra de la capa de tráfico de Yandex en niveles de zoom cercanos mostrando el tráfico por calles y la velocidad en el tramo. Teniendo esta información en cuenta se puede crear una ruta considerando el tráfico para exportarla (imagen de la derecha).

Respecto a la integración de Baidu la Figura 10 muestra la combinación de capa de satélite de Baidu y su capa de nombres así como la capa normal del mapa de Baidu. Como ya se ha mencionado debido a la problemática de los mapas en China, no se compara la capa de Baidu con la de otros servicios. La Figura 11 muestra la integración del servicio de alertas de Baidu y la información obtenida de las incidencias.

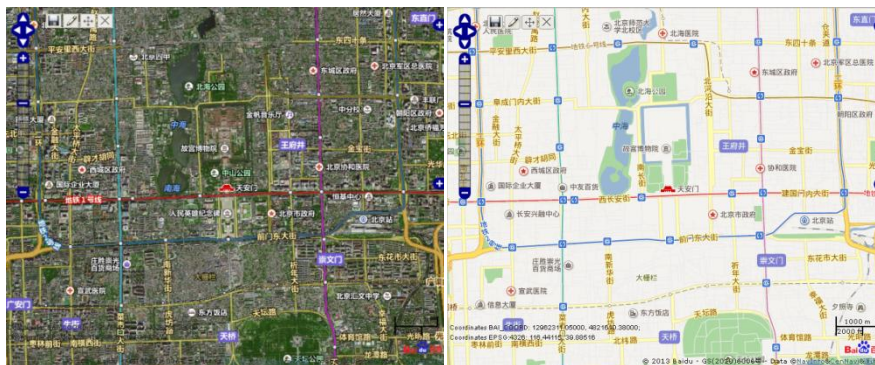


Figura 10 - La imagen de la izquierda muestra como capa base la capa de satélite de Baidu en combinación con la capa de overlay de nombres de Baidu. La imagen de la derecha muestra la capa del mapa de Baidu.



Figura 11 - La imagen de la izquierda muestra en el mapa dos incidencias de Baidu. La de la derecha muestra un popup con información de una incidencia de Baidu con imagen.

2.2. Parte móvil

La segunda parte del proyecto consistió en añadir soporte para mapas sin conexión en una librería de GeosLab que han desarrollado para Android y que usan en fase de producción. Para realizar esta parte del proyecto se comenzó con un análisis de diferentes librerías que puedan hacer uso de mapas sin conexión. Tras el análisis se consideró que Mapsforge era la que mejor respondía a las necesidades por lo que se evaluaron posibles vías de integración de Mapsforge en la librería de GeosLab. Por último se creó un caso de uso que utilizara el trabajo realizado.

2.2.1. Estudio de librerías de mapas offline para Android

Se realizó un estudio de diferentes librerías existentes que permiten el uso de mapas sin conexión. El estudio completo, se encuentra en el Anexo H. .

Para realizar el estudio se buscaron librerías o aplicaciones de código abierto que tuvieran funcionalidad para la visualización de mapas sin conexión a Internet. Uno de los aspectos a valorar fue también la licencia. Se analizaron las siguientes librerías o aplicaciones de código abierto: Big Planet Tracks, AndNav2, Osmdroid, Mapsforge y OsmAnd. Tras el análisis la opción escogida fue Mapsforge ya que dispone de sus propios mapas vectoriales basados en la información de OpenStreetMap y alojados en sus servidores y además cuenta con una licencia que permitiría que la aplicación que la utilice use el tipo de licencia que desee ya sea Open Source o cerrada.

2.2.2. Análisis de Mapsforge

Mapsforge es una librería para Android de código abierto (usa GNU Lesser GPL v3). Una de las principales ventajas de Mapsforge es que usa mapas vectoriales que ocupan relativamente poco, como ejemplo el mapa para toda España ocupa 208 MB.

La librería tiene soporte para visualización de un archivo que contiene el mapa sin conexión y que está en un formato propio de Mapsforge o bien visualización de mapas de OpenStreetMap online. La actividad del renderizado de tiles se encuentra centralizada en la clase "DatabaseRenderer" a la que es necesario pasarle la ruta del fichero del mapa sin conexión. La librería crea las tareas de renderizado de teselas según sean necesarias. Esas tareas se colocan en una cola de trabajo gestionada por la clase MapWorker que se ejecuta en un hilo aparte. De esta manera el renderizado se lleva a cabo por un único hilo aparte del hilo principal. Esta estructura impide el uso del renderizador por múltiples hilos de forma que se pudiese separar el dibujado de múltiples teselas en varios hilos. Además, tampoco es posible ir mostrando al usuario la tesela conforme se va dibujando, ya que hasta que no se ha terminado de pintar no se devuelve la imagen por parte de la librería. Los colores que usa la librería para el dibujado de las teselas se pueden modificar variando el tema que usa (el tema por defecto se encuentra en un archivo XML dentro de la librería de Mapsforge). En la imagen de la Figura 12 se puede ver la estructura básica de Mapsforge.

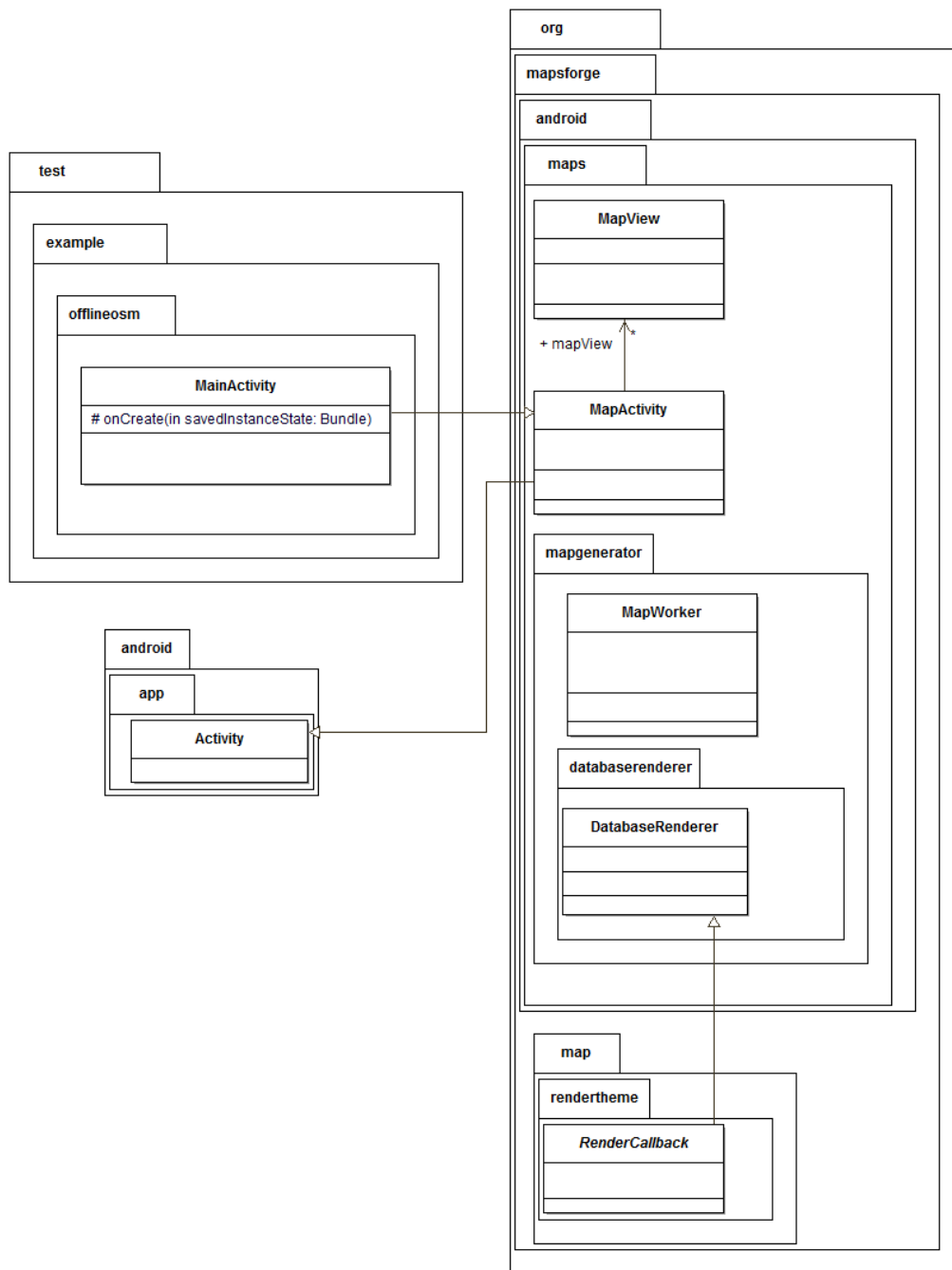


Figura 12 - Diagrama básico de la estructura de Mapsforge

La clase "MainActivity" que se muestra en la Figura 12 corresponde a la clase principal de una aplicación Android cualquiera que quiera utilizar la librería de Mapsforge. De acuerdo con el análisis realizado de Mapsforge, existen principalmente dos vías de integración en la librería. Una posible vía sería tratar de integrar directamente la clase "MapActivity" que usan las aplicaciones que utilizan la librería. La otra posibilidad sería integrar directamente el renderizador de teselas sin conexión "DatabaseRenderer" de modo que se realice el acceso a las clases necesarias dentro de la librería. Ambas opciones tienen sus ventajas e inconvenientes, ya que cuando más profundo es el acceso dentro de la librería más difícil puede ser por ejemplo realizar actualizaciones de versión de la librería, pero también permite una mayor flexibilidad en la integración.

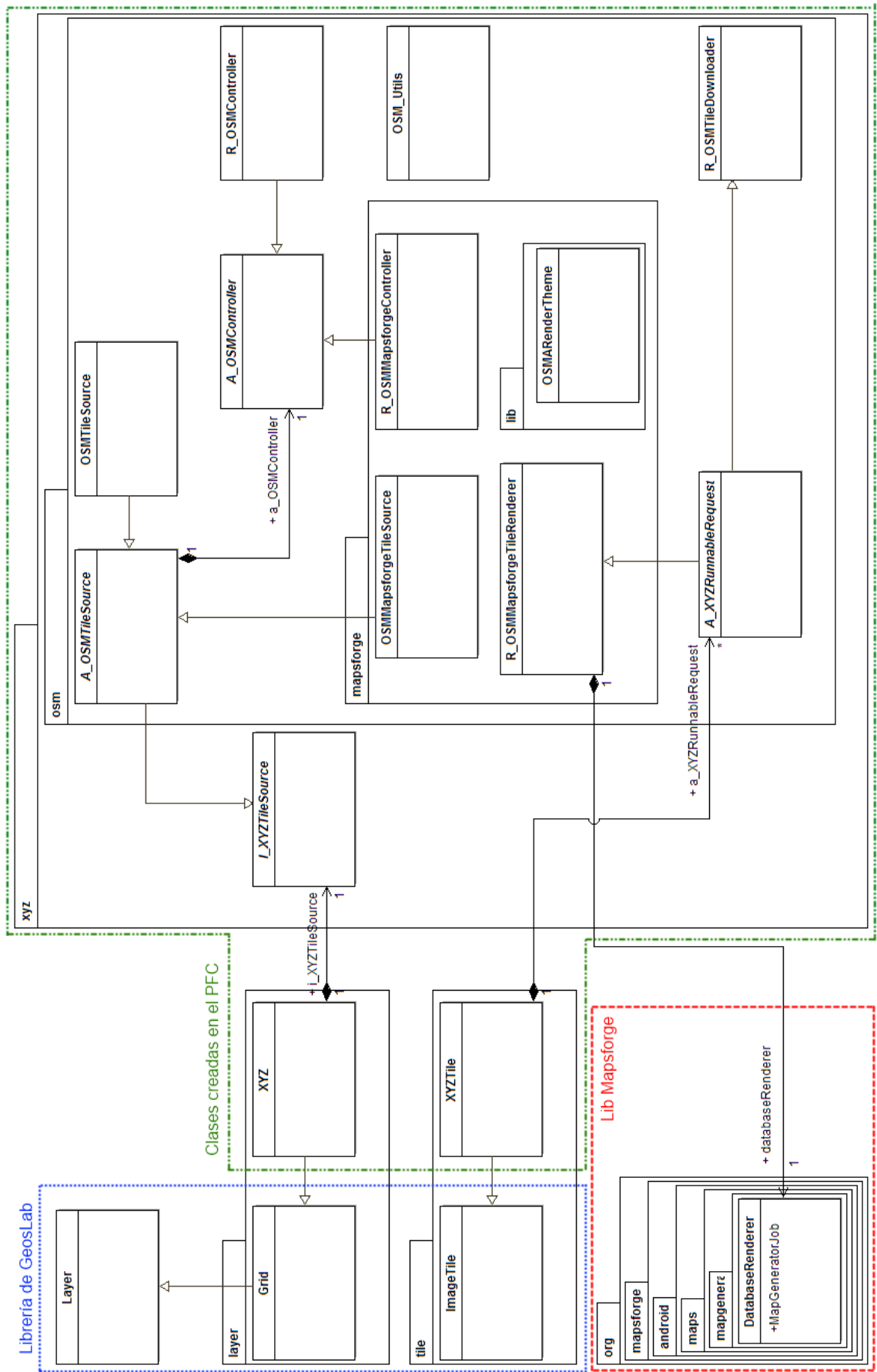


Figura 14 - Diagrama de clases de la arquitectura de las clases creadas en la librería de Android de GeosLab

El siguiente diagrama muestra las clases que se han implementado para hacer uso de la arquitectura creada en la librería. Las clases que se han implementado en este PFC son todas las que pertenecen al paquete "com.android.gsl_mobilemap" (Figura 15).

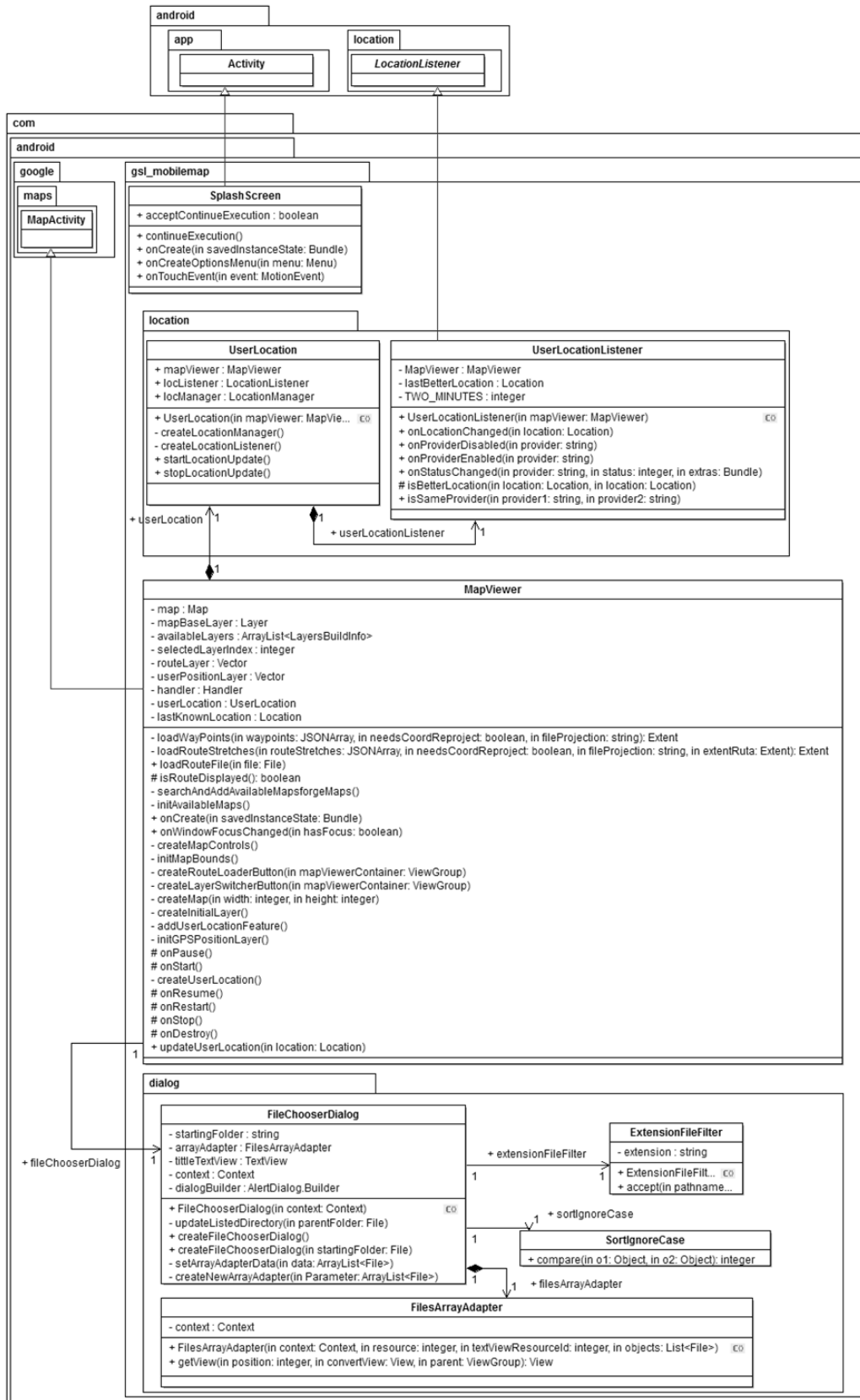


Figura 15 - Diagrama de clases de la aplicación creada en Android

2.2.4. Caso de uso

Para el caso de uso de la parte móvil de Android se ha creado una aplicación capaz de usar tanto capas de mapas de OpenStreetMap Online como mapas de Mapsforge. Además la aplicación creada para el caso de uso permite el uso de mapas del proveedor Mapquest que ofrece teselas de mapas de OpenStreetMap así como también teselas de vistas aéreas de licencia abierta (tienen cobertura mundial en los niveles de zoom del 0 al 11 y en EEUU también a mayores niveles de zoom).

Además de la integración de las capas, se ha añadido soporte para mostrar una ruta sobre el mapa y que se carga desde un archivo en formato JSON que tenga el usuario en el teléfono y que haya sido generada por la primera parte del proyecto. Para la aplicación del caso de uso también se han implementado clases para gestionar la ubicación del usuario, tanto utilizando GPS como localización basada en redes (Figura 16). Como ya se ha mencionado, las rutas se puede generar con la primera parte del proyecto (parte Web) para luego ser cargadas en el dispositivo. Una vez que la ruta está en el dispositivo se puede usar como referencia aunque no se disponga de conexión a internet.

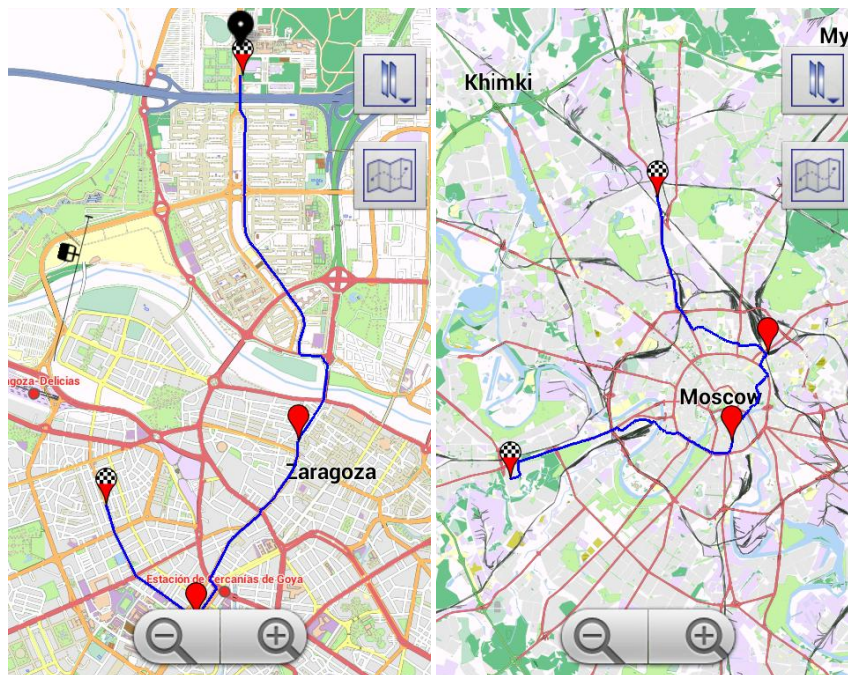


Figura 16 - Ejemplos de rutas cargadas en el teléfono sin conexión a Internet usando mapas sin conexión. La imagen de la izquierda muestra la posición actual del usuario utilizando el GPS del teléfono con el marcador negro sobre el mapa. Ambas imágenes corresponden a capturas de pantalla de la aplicación funcionando sobre un Samsung Galaxy S2 con Android Jelly Bean.

La aplicación permite elegir la capa a visualizar mediante un mensaje de selección. En ese mensaje el usuario puede elegir entre mostrar la capa de OSM oficial, la de Mapquest Streets o Mapquest Aerial. Además de estas capas, el usuario tendrá más opciones para elegir según los mapas de Mapsforge que tenga cargados en el teléfono. La aplicación busca mapas en varias ubicaciones fijas del teléfono, algunas en la propia memoria del teléfono

o bien en la tarjeta SD si hay alguna disponible. Si se encuentran mapas disponibles se muestran también en la lista para que el usuario los pueda seleccionar. En la Figura 17 se puede ver el menú mostrado cuando se dispone de los mapas para España, China, Rusia y Portugal.

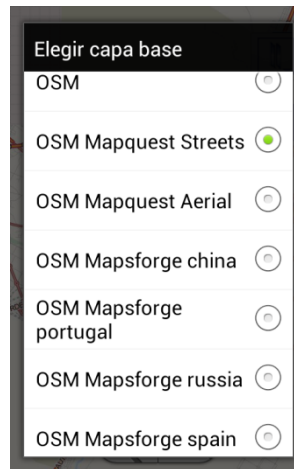


Figura 17 - Menú de selección de capa por parte del usuario

En cuanto a la carga de rutas, se hace mediante un dialog de selección de fichero que ha tenido que ser implementado. Esto ha sido necesario ya que al contrario de lo que ocurre en Java con JFileChooserDialog, Android no dispone de ningún selector de ficheros para las aplicaciones, por lo que se ha tenido que realizar la implementación a partir de un diálogo de selección normal de Android. Con el selector implementado el usuario puede navegar por el sistema de archivos de su teléfono y desplazarse hasta el directorio donde se encuentren las rutas que quiere cargar (un ejemplo de esto se encuentra en la Figura 18).

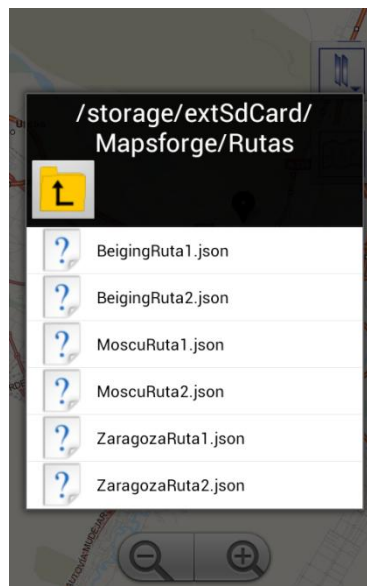


Figura 18 - Menú de selección de rutas

Para la carga de las teselas se ha creado una estructura utilizando múltiples hilos que permite descargar múltiples teselas simultáneamente. En las teselas de Mapsforge sin embargo, el renderizado se hace con único thread ya que la librería no es compatible con multi-threading.

3. Conclusiones

3.1. Cumplimiento de objetivos

Tras concluir este proyecto, es posible afirmar que se han conseguido alcanzar todos los objetivos planteados inicialmente en la propuesta. Se ha realizado el análisis de diferentes proveedores eligiendo a Yandex y a Baidu por sus singularidades. También se realizó el análisis de librerías que permitían el uso de mapas offline en Android. Finalmente se creó un prototipo de un sistema que ilustre el potencial de todo el trabajo desarrollado creando dos casos de uso para cada parte (parte Web y móvil) que trabajan en conjunto.

A lo largo del proyecto han sido varios los obstáculos que se han ido encontrando. En el caso de Baidu por ejemplo se ha presentado un paper (se puede leer en el Anexo F.) en las JIIDE 2013 (Jornadas Ibéricas de Infraestructuras de Datos Espaciales) celebradas entre el 13 y el 15 de noviembre en la Universidad de Castilla-La Mancha en Toledo. El caso de Yandex también ha presentado ciertas dificultades ya que no utilizan la misma proyección que otros servicios de mapas.

Respecto a la parte móvil se han cumplido los objetivos planteados e incluso se han ampliado con algunos más. La aplicación desarrollada en Android permite el uso de mapas y rutas sin conexión y además se le ha añadido un extra que no se había planteado inicialmente y es el soporte para teselas online de servicios estilo OpenStreetMap como Mapquest o las propias teselas de OpenStreetMap.

3.2. Líneas futuras

Todo el trabajo desarrollado en este PFC es un prototipo, no un producto final ya listo para ser usado en producción. Con este PFC se ha creado una base para seguir trabajando y crear aplicaciones que hagan uso de los elementos integrados en ambas librerías (parte web y parte móvil). Son muchas las aplicaciones que se pueden crear utilizando esto como base. Un ejemplo podrían ser aplicaciones web que permitan la planificación de rutas en diferentes lugares del mundo teniendo en cuenta factores como el tráfico, incidencias ocurridas u otros factores que puedan afectar a una ruta de tráfico. Con respecto a la parte móvil, también tiene un gran potencial de uso a partir del trabajo realizado. Una posibilidad puede ser la creación y exploración de rutas utilizando un dispositivo móvil Android con GPS bien a pie, en coche, o en móvil.

En la parte web existen ciertos aspectos que podrían ser mejorados. Tanto Yandex como Baidu disponen de otros servicios que podrían ser integrados en la librería si se deseara. También sería interesante revisar el trabajo realizado con Baidu si en algún momento el gobierno Chino decidiese eliminar la obligatoriedad del desplazamiento cosa que en principio no se ve como algo posible o si bien la fórmula o el algoritmo para el desplazamiento que aplican pasasen al dominio público.

Respecto a la parte móvil tiene un gran potencial de uso en aplicaciones de todo tipo al permitir el acceso a información cartográfica sin conexión a internet

(siempre y cuando esté cargada previamente en el dispositivo) permitiendo de esta manera mostrar al usuario un mapa para una región en cualquier situación. Algunas mejoras sobre lo desarrollado podrían consistir por ejemplo en el acceso a la información contenido en el archivo del mapa referente a calles por ciudades, comercios, cajeros automáticos y demás para de esa manera poder mostrar algunos elementos en el mapa con mayor información así como poder realizar búsquedas de esos elementos. De cara al desarrollo de una aplicación que haga uso de los elementos integrados en la librería, sería interesante que disponga de un sistema de navegación (indicaciones) para ir guiando al usuario paso a paso, por lo que sería necesario integrar también algún sistema capaz de la creación de rutas dinámicas accediendo a la estructura de nodos de las calles. También sería interesante consultar si el mapa de Mapsforge dispone de información relativa a velocidades en tramos de vías para de esa manera poder mostrar esa información al usuario. A nivel de mejoras internas, quizá la mayor mejora podría ser que se unificase el gestor de descargas de teselas de modo que esté integrado en el mapa en lugar de en cada capa por separado, lo que por ejemplo podría originar una carga mayor en caso de usar capas de overlay. Otra posible mejora a valorar sería modificar la librería de Mapsforge para permitir el renderizado de teselas desde múltiples hilos, ya que actualmente la librería realiza el renderizado desde un solo hilo, lo cual se ha tenido que respetar en la integración realizada creando ahí un cuello de botella.

3.3. Valoración personal

El desarrollo de este proyecto tiene una gran componente de investigación y desarrollo. Esto se debe a que en varios de los casos se ha estado trabajando con algunos aspectos en los que no se habían realizado trabajos previos de otros en los que se pudiese apoyar este PFC. Yandex y Baidu son dos proveedores que tienen ciertas problemáticas asociadas y que hasta ahora nadie había intentado integrar en OpenLayers. Cabe señalar que por ejemplo la problemática encontrada durante la integración de Baidu ha permitido la creación y presentación de un paper en las JIIDE 2013 (Jornadas Ibéricas de Infraestructuras de Datos Espaciales) y que se puede consultar en el Anexo F.

Además este proyecto me ha permitido explorar en más profundidad algunas tecnologías interesantes para mi formación como pueden ser los sistemas de información geográfica, el sistema GPS, las coordenadas mundiales usadas por los mapas y las proyecciones de estos, Javascript usado en la parte web del proyecto, Android utilizado en la parte móvil y el uso de GPS en Android.

Personalmente considero que este proyecto ha supuesto una aportación muy positiva a mi formación por los temas tratados y las tecnologías usadas. Todas las tecnologías utilizadas en este PFC se encuentran en auge actualmente, ya que Javascript se emplea ampliamente en casi todas las webs, Android es uno de los sistemas operativos móviles más utilizados a nivel mundial y que sigue creciendo por lo que es un objetivo interesante para el desarrollo de aplicaciones por el mercado que abarca. Además hoy en día cada vez usamos más los mapas en nuestros teléfonos para movernos por el mundo usando el GPS integrado permitiéndonos así encontrar lo que nos interesa con rapidez y sin perdernos por las calles de las ciudades. En general, estoy satisfecho con todo lo aprendido durante el desarrollo y creo que podré darle uso en el futuro.

4. Anexos

En esta sección se incluye la documentación extra creada durante el desarrollo del proyecto

Anexo A. Introducción a los sistemas de información geográfica

Los Sistemas de Información Geográfica (SIG) permiten mantener y usar datos relacionados con localizaciones geográficas. El tipo de datos que manejan estos sistemas pueden ser imágenes aéreas o de satélite siempre y cuando se pueda determinar una localización geográfica sobre ellos, planos de ciudades o mapas de carreteras digitalizados, datos relacionados con ubicaciones geográficas como pueden ser tipos de suelos, precipitaciones medias, temperaturas...

Habitualmente un SIG estructura la información en forma de capas superpuestas que aportan información adicional. Actualmente este tipo de estructura se usa en prácticamente todos los servicios de mapas Web que podemos encontrar permitiendo al usuario usar unas capas u otras según la información que desee ver.

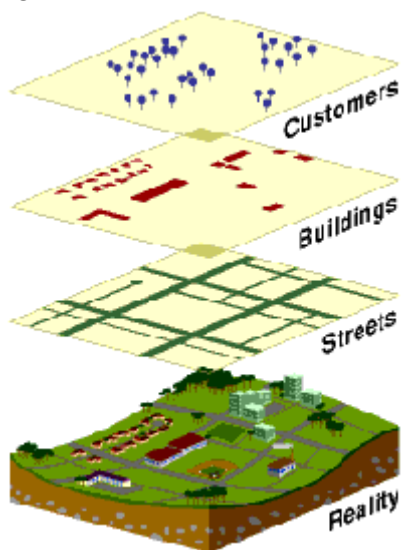


Figura A - 1 Estructura en capas de un SIG.

En los SIG estas capas reciben el nombre de coberturas y podemos encontrar dos tipos de ellas, las coberturas ráster y las vectoriales. Las coberturas ráster consisten en una matriz de celdas que representan un cierto espacio continuo. Cada elemento de esta matriz representa una zona del espacio, generalmente cada cuadrado suele representar una zona de varios metros de lado. Cada elemento de la matriz a su vez está asociado a una localización geográfica concreta por lo que a partir de la representación visual se puede obtener la localización geográfica real y viceversa. Los elementos de la matriz pueden ser por ejemplo imágenes de satélite, mapas o planos digitalizados o cualquier tipo de información representable

visualmente. En cuanto a las coberturas vectoriales, estas almacenan la información geográfica usando elementos geométricos como puntos, líneas y polígonos. Es por ello que estos reciben su correspondiente localización geográfica durante su creación. La diferencia entre ambos tipos de coberturas es básicamente que las ráster requieren tener almacenadas en el sistema las imágenes correspondientes a cada elemento de la matriz (habitualmente los elementos de esta matriz reciben el nombre de teselas o tiles en inglés) mientras que las vectoriales sólo contienen la información geométrica para el dibujado por lo que ocupan menos espacio. Sin embargo, esto a su vez tiene otro inconveniente, las capas ráster al tener ya la información dibujado sólo necesita mostrarla al usuario, sin embargo con las capas vectoriales es necesario leer la información y a continuación dibujarla para mostrarla al usuario haciendo que las capas vectoriales sean más lentas de renderizar que las ráster.

Además de los tipos de capas, existe otro elemento a tener en cuenta. Se trata de la proyección en la que se encuentra cada capa. El término proyección se refiere a la forma en la que se representa el mundo real sobre un plano. Esta representación no es una tarea sencilla, y de hecho existe una gran cantidad de proyecciones diferentes. Uno de los primeros problemas que afrontan las proyecciones es como se realiza la equivalencia entre la superficie de la Tierra y un plano. Una de las proyecciones más usadas en muchos servicios de mapas online es la conocida como proyección Mercator. La proyección Mercator proyecta los puntos de la superficie de la Tierra sobre un cilindro que la rodea. La información proyectada sobre ese cilindro es el plano al mostrar el cilindro como una superficie en dos dimensiones. Para facilitar el entendimiento se muestra a continuación una imagen al respecto extraída de la Wikipedia.

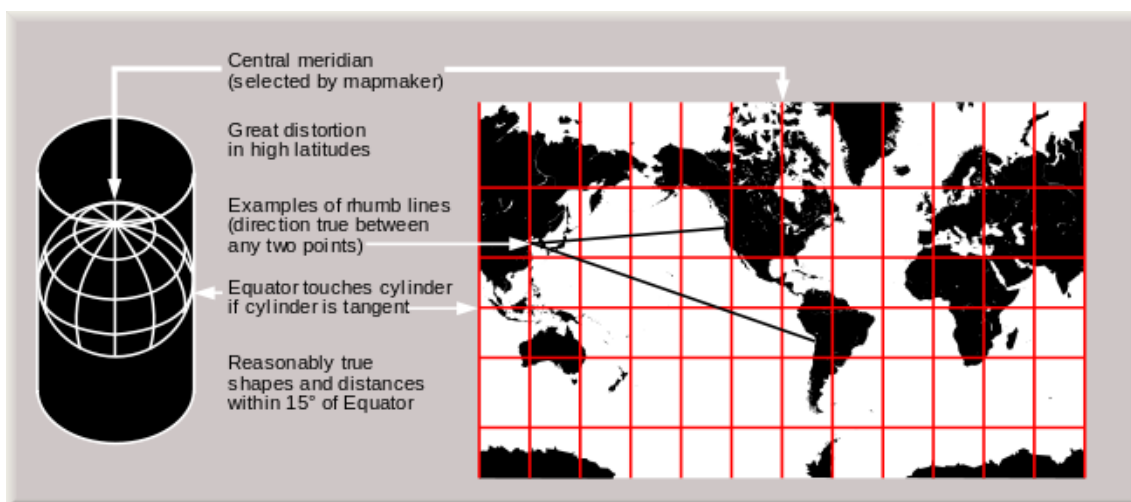


Figura A - 2 Proyección Mercator – Imagen de Wikipedia
http://commons.wikimedia.org/wiki/File:Usgs_map_mercator.svg

Sin embargo, aunque la proyección Mercator es de las más usadas hoy en día para servicios de mapas online, eso no quiere decir que sea perfecta. De hecho, esta proyección sufre una deformación considerable conforme las

coordenadas se aproximan a los polos. Esta deformación se ilustra muy claramente con la diferencia de tamaño entre Groenlandia y Australia.

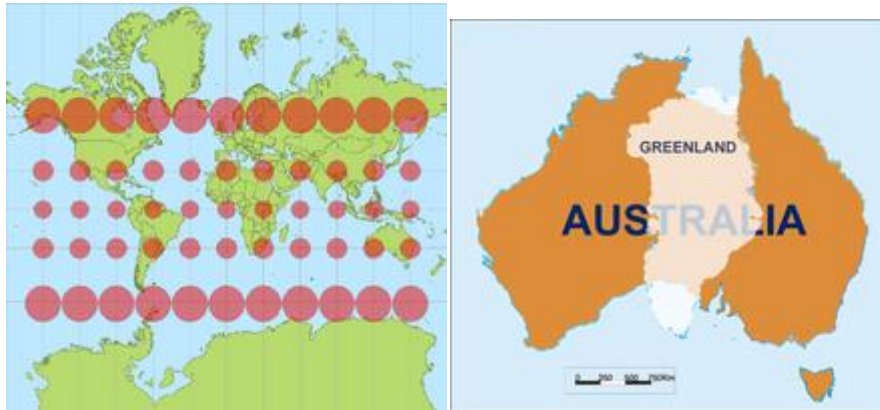


Figura A - 3Distorsión de proyección Mercator – Imágenes de Wikipedia
http://en.wikipedia.org/wiki/Mercator_projection

Existe además otro inconveniente a tener en cuenta. Aunque en muchas ocasiones se puede llegar a pensar que la Tierra es una esfera, eso no es correcto, ya que realmente la Tierra es un elipsoide. Existe una referencia usada habitualmente para la forma de la Tierra y conocida como WGS 84 Ellipsoid que es en la que se basa el sistema GPS y muchos servicios de mapas.

Anexo B. Análisis de proveedores de mapas

En este anexo se realiza un análisis de diferentes servicios de mapas, haciendo hincapié en la API y licencias que tengan para conocer las posibilidades de integración con otros servicios. La información de este documento ha sido recopilada de las páginas oficiales de cada proveedor de búsquedas. En algunos casos la información que ofrecen los proveedores sobre algún aspecto en particular puede ser escasa, por lo que si se desea ampliar la información sería necesario entrar en contacto con los respectivos proveedores para solicitar más información al respecto.

Alguno de los proveedores (principalmente Baidu) ha sido analizado usando herramientas de traducción automática por lo que la información recopilada de ellos, puede contener algún error.

Este documento ha recibido un proceso de actualización continua por lo que cada dos meses se revisaban los proveedores para comprobar cambios en sus características. Por comodidad de lectura no se ha incluido el histórico y únicamente se incluye a continuación el documento en su versión final.

Bing Maps (http://www.bing.com/maps/)

Comentario general	Mapas desarrollados por Microsoft
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://www.microsoft.com/maps/

Licencia

URL licencia	http://www.microsoft.com/maps/Licensing/licensing.aspx
Licencia de uso	Los desarrolladores de todo tipo de organizaciones pueden usar la plataforma de mapas de Bing gratuitamente siempre que cumplan unas condiciones
Limitaciones de uso	El uso será gratuito para sitios web públicos que utilicen el servicio para fines comerciales, no comercial, o gubernamentales (siempre que su uso no sea educacional o que no sea usado por parte de organizaciones sin ánimo de lucro) siempre y cuando no excedan de 125.000 transacciones en un período de 12 meses.

Posibilidades de integración

API de acceso	Sí
Información API	<p>Microsoft ofrece un SDK para las plataformas Windows Phone 7, Android, iOS y también una API REST. Para aplicaciones web y para crear aplicaciones de mapas Microsoft remite al SDK.</p> <p>La API de Bing Maps incluye:</p> <p>(Fuente: http://msdn.microsoft.com/en-us/library/dd877180.aspx)</p> <ul style="list-style-type: none">• Bing Maps para aplicaciones de Windows Store• Bing Maps AJAX Control Versión 7.0• Bing Maps AJAX Control Versión 6.3• Bing Maps iOS Control• Bing Maps Silverlight Control• Bing Maps Silverlight Control para Windows Phone• Bing Maps SOAP Services• Bing Spatial Data Services• Bing Map app SDK Beta• Bing Maps WPF Control

Cartografía

Extensión	Mundial. Excepto algunas zonas con cartografía limitada.
Calidad	Alta.
Teselado	Sí

Datos

Extensión	Mundiales, aunque servicios como StreetSide tienen una cobertura limitada. Ya que no está presente a nivel mundial.
------------------	---

Google Maps
(<http://maps.google.com/>)

Comentario general	Mapas desarrollados por Google.
Fecha última revisión	29/01/2014
URL zona desarrolladores	https://developers.google.com/maps/?hl=es

Licencia

URL licencia	https://developers.google.com/maps/licensing
Licencia de uso	<p>La API de Google Maps es un servicio gratuito que permite integrar Mapas de Google en una página que sea accesible gratuitamente y pública. No se pueden usar mapas para uso en una red interna, excepto durante el desarrollo. Ni tampoco se puede usar en una comunidad cerrada que requiera de invitaciones para acceder.</p> <p><u>Nota:</u> En aplicaciones móviles no se aplica el acceso gratuito cuando la aplicación sea vendida por un importe a través de una tienda online y es descargable a un móvil que puede acceder a esa tienda (ejemplo: aplicación vendida a través de la Play Store)</p> <p>En el caso de "Google Maps API for Business" también está permitido su uso para despliegues internos, integración en software y aplicaciones de pago, reventa de servicios con Google Maps, seguimiento de elementos privados (private asset tracking).</p>
Limitaciones de uso	<p>Google dispone de una API gratuita y una API de pago destinada a empresas.</p> <p><u>Limitaciones de Maps API (gratuita):</u></p> <ul style="list-style-type: none"> • Geocoding Web Service 2.500 solicitudes por día • Directions Web Service 2.500 solicitudes por día 10 Waypoints por solicitud • Distance Matrix Web Service 100 elementos por solicitud 100 elementos cada 10 segundos 2.500 elementos por día • Elevation Web Service 2.500 solicitudes por día 25.000 muestras por día

	<ul style="list-style-type: none"> • Static Maps API maximum resolution 640x480 • Static Maps API maximum scale 2x • Street View Image API maximum resolution 640x480 <p><u>Limitaciones de Maps API Business:</u></p> <ul style="list-style-type: none"> • Geocoding Web Service 100.000 solicitudes por día • Directions Web Service 100.000 solicitudes por día 23 Waypoints por solicitud • Distance Matrix Web Service 625 elementos por solicitud 1.000 elementos cada 10 segundos 100.000 elementos por día • Elevation Web Service 100.000 solicitudes por día 1.000.000 muestras por día • Static Maps API maximum resolution 2048x2048 • Static Maps API maximum scale 4x • Street View Image API maximum resolution 2048x2048
--	---

Posibilidades de integración

API de acceso	Sí
Información API	<p>Google ofrece diferentes API de acceso a su servicio:</p> <ul style="list-style-type: none"> • Maps JavaScript versión 3 • Maps API for Flash (permite usar mapas de Google dentro de aplicaciones en Flash) • Google Earth API (permite insertar un globo digital en 3D para moverse por el mundo) • Static Maps API (permite insertar una imagen rápida y sencilla) • Servicios web (solicitudes de URL para acceder a información, el resultado se devuelve en JSON o en XML), algunos como el servicio de Geocoding disponen de interfaz REST • Maps Data API (permite visualizar, almacenar y actualizar)

Cartografía

Extensión	Mundial. Excepto algunas zonas con cartografía limitada.
Calidad	Alta.
Teselado	Sí

Datos

Extensión	Mundiales, aunque servicios como StreetView tienen una cobertura más limitada.
------------------	--

Mappy
(<http://es.mappy.com/>)

Comentario general	<p>Mappy es subsidiaria del grupo de Páginas amarillas de Francia (PagesJaunes). En la última revisión realizada el 28/01/2014 Mappy ha retirado la disponibilidad de APIs en sus Web para el uso de sus mapas. Según el mensaje que aparece en su zona de desarrolladores, los desarrolladores que se registrasen en el servicio antes de Mayo del 2013 recibirán un correo electrónico informándoles de que en un plazo de dos meses tendrán que dejar de usar su servicio tanto de APIs, como SDKs para móviles como el Widget que ofrecían.</p> <p>Actualmente para el uso del servicio de Mappy es necesario contactar con ellos en el correo deportalisation@mappy.com facilitando un correo y un teléfono además de una presentación clara del proyecto del desarrollador (descripción y características del mapa deseadas) y ellos se pondrán en contacto con el desarrollador para hablar de su proyecto.</p>
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://corporate.mappy.com/faq/integrez-mappy/

Licencia

URL licencia	N/A
Licencia de uso	N/A
Limitaciones de uso	N/A

Posibilidades de integración

API de acceso	No pública.
Información API	Mappy ya no ofrece acceso a APIs de forma pública.

Cartografía

Extensión	Mundial. Excepto algunas zonas con cartografía limitada.
------------------	--

Calidad	Alta.
Teselado	Sí

Datos

Extensión	Mundiales, aunque la calidad de las imágenes de satélite está limitada. El último nivel de zoom en el que se dispone de imágenes de satélite es para una escala del mapa de 6 Km.
------------------	---

<h2 style="margin: 0;">MapQuest</h2> <p style="margin: 0;">(http://www.mapquest.es/)</p>
--

Comentario general	<p>Es un servicio de mapas propiedad de America Online desde el año 2.000.</p> <p>La página dispone también de un editor de mapas para los mapas de OpenStreetMap.</p> <p>(http://open.mapquest.com/)</p>
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://developer.mapquest.com/

<h2 style="margin: 0;">Licencia</h2>

URL licencia	http://developer.mapquest.com/web/tools/getting-started/terms-overview
Licencia de uso	<p>Mapquest ofrece dos tipos de licencia de uso en función de los intereses del desarrollador. Para la versión Community se pueden usar datos libre provenientes de OpenStreetMaps (Open Data) o bien datos de fuentes comerciales (licensed data).</p> <ul style="list-style-type: none"> • Enterprise Edition (Licensed Data) (comenzando con 2.500 \$ anuales) • Community Edition (Licensed Data) • Community Edition (Open Data)
Limitaciones de uso	<p>ENTERPRISE</p> <ul style="list-style-type: none"> • Mapas Sin límite preestablecido • Direcciones Sin límite preestablecido • Geocodificación Sin límite preestablecido • Búsqueda Sin límite preestablecido • Matriz de rutas Sin límite preestablecido • Tráfico Sí • Aplicaciones para móvil no comerciales o públicas Sí • Aplicaciones web privadas y/o comerciales Sí • Aplicaciones móviles de pago Sí

	<ul style="list-style-type: none"> • SSL Sí • Foro de soporte Sí • Soporte Premium (soporte 24/7 teléfono y email) Sí • Niveles de rendimiento del sistema Sí • Gestión de cuentas Sí <p>COMMUNITY</p> <p><u>Licensed Data</u></p> <ul style="list-style-type: none"> • Mapas Sin límite preestablecido • Direcciones 5.000 solicitudes/día • Geocodificación 5.000 solicitudes/día • Búsqueda 5.000 solicitudes/día • Matriz de rutas 5.000 pares de ruta/día • Tráfico Sí • Aplicaciones para móvil no comerciales o públicas Sí • Aplicaciones web privadas y/o comerciales No • Aplicaciones móviles de pago Sí • SSL Sí • Foro de soporte Sí • Soporte Premium (soporte 24/7 teléfono y email) No • Niveles de rendimiento del sistema No • Gestión de cuentas No <p><u>Open Data</u></p> <ul style="list-style-type: none"> • Mapas Sin límite preestablecido • Direcciones Sin límite preestablecido • Geocodificación Sin límite preestablecido • Búsqueda Sin límite preestablecido • Matriz de rutas
--	---

	<p>No</p> <ul style="list-style-type: none"> • Tráfico <p>No</p> <ul style="list-style-type: none"> • Aplicaciones para móvil no comerciales o públicas <p>Sí</p> <ul style="list-style-type: none"> • Aplicaciones web privadas y/o comerciales <p>Sí</p> <ul style="list-style-type: none"> • Aplicaciones móviles de pago <p>Sí</p> <ul style="list-style-type: none"> • SSL <p>No</p> <ul style="list-style-type: none"> • Foro de soporte <p>Sí</p> <ul style="list-style-type: none"> • Soporte Premium (soporte 24/7 teléfono y email) <p>No</p> <ul style="list-style-type: none"> • Niveles de rendimiento del sistema <p>No</p> <ul style="list-style-type: none"> • Gestión de cuentas <p>No</p> <p>Nota: La API de tráfico no se puede usar para navegación en tiempo real ni tampoco como objetivo principal de la página web o de la aplicación. MapQuest se reserva el derecho a limitar el acceso para evitar la degradación del servicio.</p>
--	--

Posibilidades de integración

API de acceso	Sí.
Información API	<p>Licensed Data Map APIs and Web Services</p> <ul style="list-style-type: none"> • Web Services <ul style="list-style-type: none"> ▪ Data Manager API Web Service ▪ Directions API Web Service ▪ Geocoding API Web Service ▪ Search API Web Service ▪ Static Maps API Web Service ▪ Traffic API Web Service ▪ Long URL API Web Service • JavaScript Maps API v7 • Flash Maps API • Android Maps API • iOS Maps API <p>Open Data Map APIs and Web Services</p> <ul style="list-style-type: none"> • Open JavaScript Maps API • Open Flash Maps API • Open Web Services <ul style="list-style-type: none"> ▪ Open Directions API Web Service ▪ Open Guidance API Web Service

	<ul style="list-style-type: none"> ▪ Open Geocoding API Web Service ▪ Open Static Map Web Service ▪ Open Elevation API Web Service ▪ Nominatim API Web Service ▪ Xapi API Web Service • Open Map Tiles <ul style="list-style-type: none"> ▪ MapQuest-OpenStreetMap Tiles ▪ MapQuest Open Aerial Tiles <p>La versión Community usa los mapas de OpenStreetMap y la interfaz que usa para las peticiones es similar al formato XYZ usado por OpenStreetMap.</p>
--	--

Cartografía

Extensión	Mundial. Excepto algunas zonas con cartografía limitada.
Calidad	Alta.
Teselado	Sí

Datos

Extensión	<p>Mundiales, aunque la versión americana del sitio www.mapquest.com tiene más opciones (búsqueda de comercios, alojamientos, tráfico...) que la versión en castellano www.mapquest.es carece de esas opciones. La versión .com usa mapas de NAVTEQ, mientras que la versión .es utiliza mapas de OpenStreetMap.</p> <p>Como se ha comentado, la versión americana permite la búsqueda de comercios, estado del tráfico y demás. Estos servicios son bastante completos en el territorio de Estados Unidos, fuera de él, la información está incompleta o es nula. Por ejemplo, el servicio de tráfico únicamente dispone de información para EEUU.</p> <p>Utilizando la versión de http://open.mapquest.com se puede obtener algo más de información sobre comercios, pero aún así la información mostrada parece que tampoco es del todo completa, ya que por ejemplo, al buscar en el mapa por ejemplo ATMs (Cajeros automáticos) sólo se muestran cuatro para toda Zaragoza.</p> <p>En cuanto al uso de las imágenes de satélite en conjunto con el mapa, el zoom máximo sobre la imagen de satélite en el territorio de Estados Unidos corresponde a una escala en el mapa de 50 m (100 pies), mientras que en territorio de España el zoom</p>
------------------	--

	máximo para imágenes de satélite es de 5.000 m (2 millas).
--	--

<p>NearMap (http://www.nearmap.com)</p>
--

Comentario general	NearMap es una compañía Australiana que ofrece imágenes aéreas de alta resolución, aunque se limitan únicamente a poblaciones el territorio de Australia. En el 2.008 fue comprada por Ipernica. Es servicio disponía de una API de acceso público, sin embargo a fecha 25/06/2013 la API dejó de estar disponible para el público y pasó a ser un servicio cerrado.
Fecha última revisión	28/01/2014
URL zona desarrolladores	N/A

Licencia

URL licencia	N/A
Licencia de uso	N/A
Limitaciones de uso	N/A

Posibilidades de integración

API de acceso	N/A
Información API	N/A

Cartografía

Extensión	Principalmente Australia, más concretamente poblaciones de esa región. En el resto del mundo aunque aparece cartografía, ésta no está disponibles para niveles próximos de zoom, por lo que es imposible llegar a ver calles de ciudades en otras regiones diferentes de Australia.
Calidad	Alta para poblaciones de Australia, baja para el resto del mundo.
Teselado	Sí

Datos**Extensión**

Las imágenes aéreas de calidad están limitadas a las poblaciones australianas, el resto de Australia y del mundo cuentan con imágenes aéreas, pero desde gran altura y sin mucha calidad, por lo que resultan poco útiles para ver poblaciones de otros países.

Here Maps (Nokia Maps)

(<http://here.net/>)

Comentario general	<p>Servicio ofrecido por Nokia para sus dispositivos móviles. También se les conoce como Ovi Maps. En el 2.007 Nokia compró Navteq y desde ese momento incluyó su cartografía en el servicio de mapas de Nokia.</p> <p>Actualmente Nokia ofrece su servicio de mapas con el nombre de Here Maps.</p>
Fecha última revisión	28/01/2014
URL zona desarrolladores	https://developer.here.com/

Licencia

URL licencia	<p>http://developer.here.net/faqs#l&t</p> <p>http://developer.here.net/terms_conditions</p>
Licencia de uso	<p>Nokia ofrece con sus mapas Here dos planes de licencia:</p> <p>Base Plan</p> <ul style="list-style-type: none"> Ofrece acceso a una serie de APIs sujetas a una limitación diaria de uso. Se permite el acceso a los mapas básicos, funcionalidad de direcciones y lugares tanto para uso comercial como no comercial. Sólo es aplicable para aplicaciones orientadas a consumidores no para usuarios de empresa. <p>Commercial Plan</p> <ul style="list-style-type: none"> Ofrece un Service Level Agreement (SLA) con límites de uso diarias más elevados y hasta 1.000.000 de transacciones acumuladas por mes entre todas las API. Además este plan da acceso a las teselas de tráfico que no da el plan base.
Limitaciones de uso	<p>Límites de transacciones:</p> <p>Base:</p> <ul style="list-style-type: none"> Teselas de mapas 2D, servidas vía CDN, en Normal, Móvil, Peatón, Satélite/aéreas, Gris: Sin Límite. Mapas estáticos 2D: 2.500 por día Rutas coches: 2.500 por día Rutas peatones: 2.500 por día Geocodificación: 2.500 por día

	<ul style="list-style-type: none"> • Geocodificación inversa: 2.500 por día • Búsqueda: 2.500 por día <p>Core:</p> <ul style="list-style-type: none"> • Servicio de teselas de tráfico incluyendo flujos y patrones: 100.000 por día • Teselas de mapas 2D, servidas vía CDN, en Normal, Móvil, Peatón, Satélite/aéreas, Gris: Sin Límite. • Mapas estáticos 2D: 100.000 por día • Rutas coches: 100.000 por día • Rutas peatones: 100.000 por día • Geocodificación: 100.000 por día • Geocodificación inversa: 100.000 por día • Búsqueda: 100.000 por día <p><u>Nota:</u></p> <p>Para una aplicación que use los mapas Here y que se quiera vender a través de una tienda de aplicaciones como la Nokia Store o cualquier otra, es obligatorio usar el plan Core. Una aplicación desarrollada con el Base Plan es sólo podrá ser ofrecida de forma gratuita siempre y cuando no supere el número de peticiones.</p>
--	---

Posibilidades de integración

API de acceso	Sí.
Información API	<p>Nokia ofrece varias soluciones:</p> <p>Here Maps API para JavaScript permite:</p> <ul style="list-style-type: none"> • Mostrar mapas interactivos de una posición geográfica dada con capas como marcadores, poli líneas, polígonos, mapa de datos heat. • Renderizar varios tipos de mapas • Gestionar el IU • Realizar búsquedas • Calcular y mostrar rutas <p>Nativas:</p> <ul style="list-style-type: none"> • Windows Phone SDK • Android SDK • Otros SDK nativos: <ul style="list-style-type: none"> ○ Qt: Para aplicaciones móviles y de escritorio. ○ Java ME: para dispositivos compatibles con Java Micro Edition. <p>REST:</p> <ul style="list-style-type: none"> • Permite acceder a imágenes pre-renderizadas para todas las regiones del mundo.

	<ul style="list-style-type: none"> Se puede usar el servicio para mostrar una localización dada, puntos de interés, rutas, puntos de giro y unión de vistas. Los datos necesarios como la posición, el radio, el nivel de zoom, el tipo de mapa, etc. se especifican por parámetros en la solicitud URL. <p>Mobile HTML5 Framework:</p> <ul style="list-style-type: none"> Información de uso en: http://developer.here.net/faqs#mobile_html5
--	--

Cartografía

Extensión	Mundial. Excepto algunas zonas con cartografía limitada.
Calidad	Alta
Teselado	Sí

Datos

Extensión	Mundiales. Aunque algunas zonas con cartografía limitada. Además algunos servicios por ejemplo Nokia Map Creator no está disponible para algunos países como es el caso de España.
------------------	--

<h2 style="margin: 0;">OpenStreetMap</h2> <p style="margin: 0;">(http://www.openstreetmap.es/)</p>
--

Comentario general	OpenStreetMap (OSM) es un proyecto colaborativo para crear un mapa del mundo editable gratuitamente.
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://wiki.openstreetmap.org/wiki/Main_Page

Licencia

URL licencia	http://opendatacommons.org/licenses/odbl/summary/
Licencia de uso	<p>OpenStreetMap es básicamente una base de datos, por lo que usan una licencia tipo Open Database License (oDbL) para ella.</p> <p>OpenStreetMap establece restricciones en el uso de teselas (Tiles) desde sus servidores. Los datos son gratuitos, pero el acceso a los servidores de teselas no lo es según se puede leer en la página: http://wiki.openstreetmap.org/wiki/Tile_usage_policy</p>
Limitaciones de uso	<ul style="list-style-type: none"> • Prohibido el uso intensivo (ejemplo: distribuir una aplicación que use teselas desde openstreetmap.org) sin autorización previa de los administradores. • Las llamadas a <code>/cgi-bin/export</code> sólo pueden ser desencadenadas por el usuario, ya que resultan costosas en recursos de CPU y RAM. • No se deben descargar tiles innecesarios. En concreto descargar áreas significativas de teselas a niveles de zoom de 7 o más altos para uso offline o uso posterior está prohibido por la carga que eso supone a los servidores ya que esas imágenes normalmente no están en la cache del servidor por lo que se tienen que renderizar. <p>Nota: Existen otros proveedores de teselas para los mapas de OpenStreetMap, algunos de ellos gratuitos: Servidores alternativos de teselas de OpenStreetMap</p>

Posibilidades de integración

API de acceso	Sí.
Información API	Dispone de una API basada en las ideas de una API

	<p>RESTful. El interfaz de la API es de tipo XYZ.</p> <p>Si lo que se quiere es simplemente añadir un mapa a una página web se puede recurrir a las librerías que figuran aquí. En esa página se listan varias librerías, en JavaScript y en flash y con diferentes tipos de licencias.</p>
--	---

Cartografía

Extensión	Mundial. Aunque pueden existir zonas con cartografía limitada o nula.
Calidad	Alta
Teselado	Sí

Datos

Extensión	Mundial. Sobre todo en grandes ciudades abunda información sobre comercios, hoteles, restaurantes... aunque en cambio en pequeñas poblaciones puede haber una carencia importante de información.
------------------	---

ViaMichelin
(<http://www.viamichelin.es/>)

Comentario general	ViaMichelin es una subsidiaria de la compañía Michelin Group.
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://dev.viamichelin.com/

Licencia

URL licencia	N/A
Licencia de uso	N/A
Limitaciones de uso	N/A

Posibilidades de integración

API de acceso	Sí
Información API	<p>ViaMichelin ofrece diferentes API de acceso, se puede pedir una cuenta de evaluación de 45 días para probar los servicios, pero para su uso, es necesario contactar con su equipo de ventas y solicitar un presupuesto.</p> <ul style="list-style-type: none"> • ViaMichelin REST API • ViaMichelin JavaScript API V2 • ViaMichelin SOAP API

Cartografía

Extensión	Mundial. Aunque pueden existir zonas con cartografía limitada o nula.
Calidad	Alta
Teselado	Sí

Datos

Extensión	Mundial. Aunque pueden existir zonas con datos limitados o inexistentes.
------------------	--

WikiMapia (<http://wikimapia.org>)

Comentario general	No es un proveedor de mapas como tal, sino que se trata de una iniciativa para la creación de contenido descriptivo de cualquier lugar del mundo.
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://wikimapia.org/forum

Licencia

URL licencia	http://wikimapia.org/api/
Licencia de uso	La licencia de uso es de tipo Creative Commons.
Limitaciones de uso	Los datos de Wikimapia se pueden usar gratuitamente para uso no comercial.

Posibilidades de integración

API de acceso	Dispone de una API en la que especificando determinados parámetros en la URL de la petición se pueden obtener los datos deseados.
Información API	http://wikimapia.org/api/

Cartografía

Extensión	Mundial. Aunque pueden existir zonas con cartografía limitada o nula.
Calidad	Media
Teselado	Sí

Datos

Extensión	Mundial. Aunque pueden existir zonas con datos limitados o inexistentes.
------------------	--

Yahoo! Maps
(<http://espanol.maps.yahoo.com/>)

Comentario general	Actualmente los mapas de Yahoo han desaparecido. El servicio de mapas de Yahoo muestra directamente los mapas de Nokia. Por esto no se va a evaluar este servicio, ya que sería volver a evaluar el servicio de Nokia Maps.
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://developer.yahoo.com/maps/ (En cierre) http://developer.here.com/javascript-apis

Licencia

URL licencia	N/A
Licencia de uso	N/A
Limitaciones de uso	N/A

Posibilidades de integración

API de acceso	N/A
Información API	N/A

Cartografía

Extensión	N/A
Calidad	N/A
Teselado	N/A

Datos

Extensión	N/A
------------------	-----

Yandex Maps
(<http://maps.yandex.ru/>)

Comentario general	<p>Yandex es una compañía de Internet rusa que gestiona el mayor motor de búsqueda en Rusia. En junio del 2.008 adquirió SMI, una agencia de monitorización de tráfico para unirla a su servicio de Maps.</p> <p>Nota: La información de este proveedor se ha obtenido usando un traductor automático de ruso al inglés, por lo que puede no ser del todo exacta.</p>
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://api.yandex.ru/maps/ http://api.yandex.com/maps/

Licencia

URL licencia	<p>http://legal.yandex.ru/maps_api/ http://legal.yandex.com/maps_api/ (En su sitio informan de que la versión en inglés es sólo una traducción de la versión en ruso y que en caso de conflicto entre ambas, la que prevalecerá legalmente será lo que conste en la versión en ruso).</p>
Licencia de uso	<p>El servicio sólo puede ser usado en sitios con acceso público. No se puede usar para proyectos que requiera pago o que restrinjan el acceso a terceras partes.</p> <p>El usuario puede usar los datos y funciones obtenidos sólo en la funcionalidad ofrecida por el servicio.</p> <p>Yandex se reserva el derecho de fijar límites o restricciones en el tamaño, volumen, esquemas de color y el número de imágenes de mapas o de imágenes de satélites ofrecidas por el servicio. Estas modificaciones las pueden hacer sin previo aviso escrito a los usuarios.</p> <p>Yandex se reserva el derecho de incluir material promocional a su discreción.</p> <p>Además la licencia obliga al cumplimiento de la legislación de la Federación Rusa.</p>
Limitaciones de uso	<p>La función de Geocoding sólo puede ser usada por sitios que usen el servicio. El número de llamadas a esta función está limitado a 25.000 solicitudes al día.</p> <p>El usuario no puede usar el servicio para monitorización de vehículos que muestre información</p>

	<p>en tiempo real o cualquier otro servicio relacionado con la gestión o planificación de vehículos.</p> <p>Yandex no establece ninguna limitación especial durante la fase de desarrollo. Únicamente existe algún problema en cuanto al uso de la clave de la API en un máquina de desarrollo (127.0.0.1 o localhost). Para ello Yandex, recomienda que si se quiere usar la clave de la API (no es obligatorio) se puede crear un dominio de segundo nivel del tipo http://dev.maps y pedir la clave de API para él.</p>
--	--

Posibilidades de integración

API de acceso	Sí
Información API	<p>Dispone de la librería JavaScript API 2.x preparada para su uso también desde navegadores en dispositivos móviles táctiles.</p> <p>La API es compatible con los siguientes navegadores:</p> <ul style="list-style-type: none"> • Microsoft Internet Explorer 7.x o superior • Mozilla Firefox 2.x o superior • Opera 9.x o superior • Apple Safari 3.1 o superior • Google Chrome • Яндекс.Браузер 1.0 <p>Dispone de interfaz tipo REST para el acceso a algunos servicios como por ejemplo "metrika".</p>

Cartografía

Extensión	<p>Hasta hace unos meses la cartografía del servicio de Yandex estaba claramente dividida y diferenciada en dos según el idioma. Así se tenía el caso del sitio en su versión en ruso (http://maps.yandex.ru) en el que se centraba en la cartografía de Rusia y sus países próximos. Contando con una cartografía muy limitada para el resto del mundo. En cuanto al sitio en inglés (http://maps.yandex.com) su cartografía tenía más limitaciones en zonas de Rusia, pero sí que contaba con amplia cartografía del resto del mundo.</p> <p>Sin embargo, recientemente el servicio de Yandex ha experimentado algunos cambios, entre los que cabe destacar que actualmente tanto el servicio en la versión en ruso como el servicio en la versión en inglés cuentan con una cartografía completa que abarca una gran cantidad de países del mundo en los diferentes continentes.</p> <p>Respecto a las imágenes de satélite sigue siendo un tema pendiente, ya que sólo se encuentran en calidad</p>
------------------	--

	suficiente para los niveles más cercanos de zoom en áreas de Rusia.
Calidad	Calidad alta de la cartografía con información de gran parte del mundo. Respecto a la cobertura de satélite es alta en Rusia y países próximos, pero para el resto del mundo no están disponibles en todos los niveles de zoom.
Teselado	Sí

Datos

Extensión	<p>Dispone de cartografía extensa para casi todo el mundo exceptuando algunos países que disponen de una cartografía más limitada como puede ser el caso de Japón, Corea del Sur y por supuesto Corea del Norte para el que ningún servicio dispone de información completa.</p> <p>El servicio de Yandex cuenta además con datos adicionales como el estado del tráfico, que sólo está disponible en el territorio de Rusia o de algunos países cercanos.</p>
------------------	--

Baidu Maps
(<http://map.baidu.com/>)

Comentario general	Baidu es una compañía china que ofrece servicios web como por ejemplo un buscador o un servicio de mapas. Nota: La información de este proveedor se ha obtenido usando un traductor automático de chino al inglés, por lo que puede no ser del todo exacta.
Fecha última revisión	28/01/2014
URL zona desarrolladores	http://developer.baidu.com/map/

Licencia

URL licencia	http://developer.baidu.com/map/law.htm
Licencia de uso	Si se usa Baidu Maps para una aplicación comercial (como navegadores GPS, etc.) que directa o indirectamente se benefician, se necesita un acuerdo aparte con Baidu Maps. Para usar el servicio, tu sitio debe estar accesible para el público general y se debe ofrecer sin cargo. Se exige el compromiso de cumplir con las leyes, normas y regulaciones de la República Popular de China.
Limitaciones de uso	No hay limitación de solicitudes por día para cada mapa. Sin embargo, si una aplicación genera una carga excesiva en sus servidores por exceso de peticiones, Baidu map se reserva el derecho de limitar el número de peticiones según su propio criterio. No parece haber ningún tipo de limitación especial para el periodo de desarrollo de una aplicación que use la API.

Posibilidades de integración

API de acceso	Sí.
Información API	Baidu Maps dispone de múltiples API <ul style="list-style-type: none"> • JavaScript API • Flash API • Static graph API • JavaScript open source library • Android SDK • iOS SDK

	<ul style="list-style-type: none"> • Positioning SDK • Web Services API • Car networking API • URI API <p>Baidu usa una interfaz REST para sus servicios incluida la OpenAPI 2.0.</p>
--	---

Cartografía

Extensión	China exclusivamente.
Calidad	Alta
Teselado	Sí

Datos

Extensión	<p>China exclusivamente. Baidu además dispone de un servicio de alertas que muestra incidentes ocurridos en todo el territorio de China.</p> <p>Uno de los problemas de la cartografía de Baidu es que incluye un desplazamiento de coordenadas obligado por el Gobierno Chino (GCJ-02) además de un desplazamiento propio (llamado BD-09) por lo que las coordenadas de un elemento en un mapa de Baidu no coinciden con las coordenadas de ese mismo elemento usando las coordenadas de los GPS.</p>
------------------	--

Anexo C. Análisis de OpenLayers 2.12

En este anexo se recoge el análisis de la estructura interna de la librería de OpenLayers, centrándose en la gestión que realiza de los proveedores de mapas. Se analizará la estructura, el diseño, gestión de proveedores, modelo que sigue, escalabilidad, actores que intervienen y cuál es la responsabilidad de cada uno. Para ello se hará uso de diagramas UML como el diagrama de clases, o los diagramas de secuencia.

1. ¿Qué es OpenLayers?

OpenLayers es una librería JavaScript de código abierto (utiliza una licencia derivada de BSD) usada para mostrar datos de mapas en navegadores web. La librería permite crear aplicaciones web similares a las ofrecidas por Google Maps, o Bing Maps. Fue desarrollado originalmente por Metacarta Labs, aunque desde el 2007 es un proyecto de la Open Source Geospatial Foundation.

En la siguiente imagen de la Wikipedia se puede ver un ejemplo de infraestructura de OpenLayers y su interacción con otros servicios (principalmente es compatible con servicios OGC) para mostrar la información al usuario. Hay que tener en cuenta que ésta no es la única infraestructura posible, ya que OpenLayers permite una gran libertad a la hora de integrarlo con otros servicios y aplicaciones.

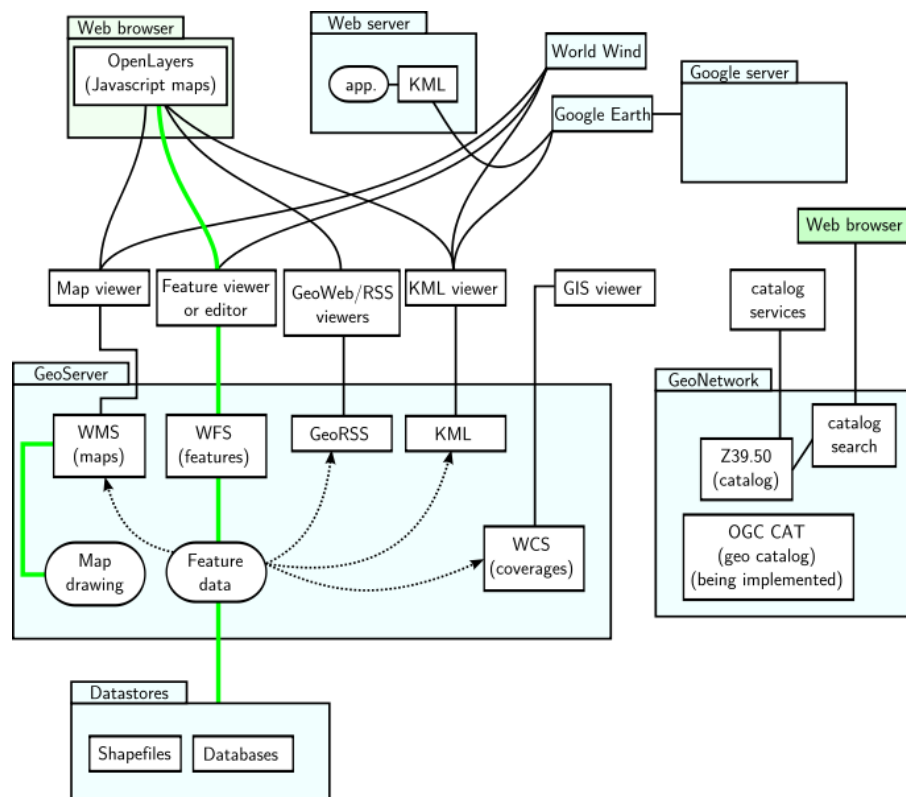


Figura C - 1Ejemplo de infraestructura de OpenLayers. (Imagen de la Wikipedia http://upload.wikimedia.org/wikipedia/commons/thumb/2/25/Geoservices_server_with_apps.png/347px-Geoservices_server_with_apps.png)

2. ¿Cómo funciona OpenLayers?

Los proveedores de mapas habitualmente disponen de capas con diferente información. Por ejemplo en Google podemos encontrar diferentes capas como Google satélite, Google terreno, Google Streets, Google híbrido... Los proveedores disponen habitualmente de varias capas mostrando diferentes tipos de información. Para gestionar esto, OpenLayers usa el concepto de capa también, pero aplicado individualmente a cada capa de cada proveedor. Es decir, toda capa que se quiera añadir al mapa, se ha de crear usando o bien una clase que gestione un tipo de servicio concreto (como WMS - Web Map Service), o bien una clase que gestione las peculiaridades de un proveedor concreto (como la clase Google, Bing...). En el caso de capas de Google o Bing usando sus respectivas clases de OpenLayers se especifica la capa del servicio usando argumentos en la construcción de la capa.

El diagrama de clases de la siguiente imagen ilustra las interrelaciones entre las diferentes clases de los proveedores. Como se puede ver, todas las clases descienden de la clase Layer, hasta llegar a las clases específicas para proveedores concretos como OSM (OpenStreetMap), Bing y Google. Y también están disponibles algunas clases concretas que pueden ser usadas directamente con múltiples proveedores como pueden ser las capas de tipo WMS (WebMapService).

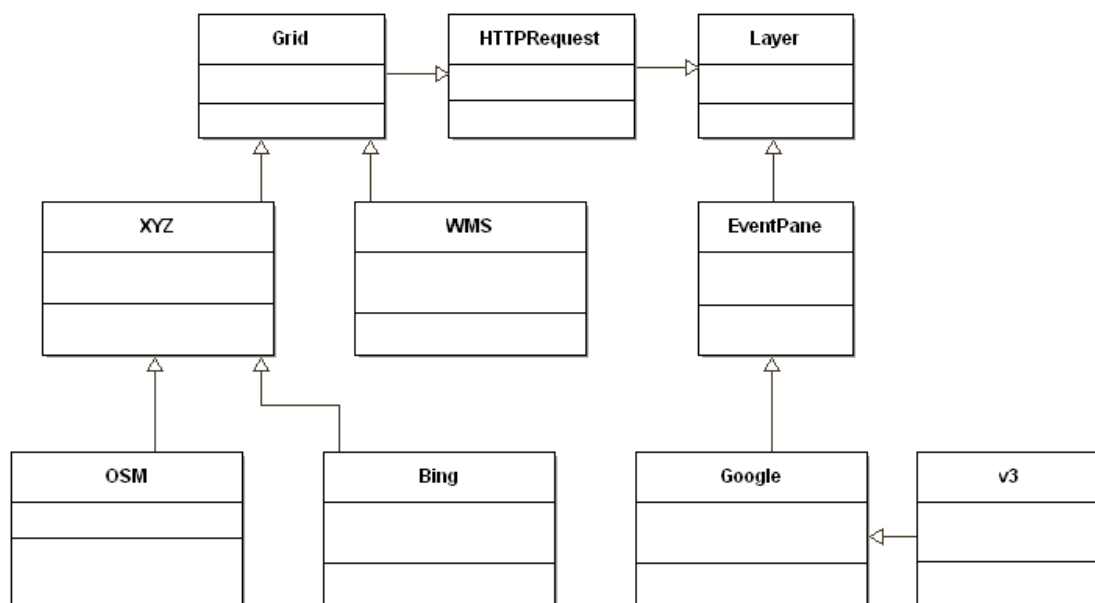


Figura C - 2 Diagrama de clases mostrando las relaciones entre las clases de OpenLayers específicas para ciertos proveedores y las clases de las que extienden.

Para añadir una capa, el primer paso es crearla con los parámetros adecuados, para posteriormente añadir la capa o capas deseadas al mapa. En la siguiente tabla se muestra un fragmento de código de ejemplo para añadir una capa de Google Maps a un mapa.

Fragmento de código para añadir capa de Google Maps a un mapa

```
map = new OpenLayers.Map('map');  
var ghyb = new OpenLayers.Layer.Google(  
    "Google Hybrid",  
    {type: google.maps.MapTypeId.HYBRID, numZoomLevels: 20}  
);  
  
map.addLayer(ghyb);  
  
//resto de código
```

A continuación se muestran los diagramas de secuencia del proceso de añadir nuevas capas a un mapa para los proveedores de mapas: Bing, Google, OSM y WMS.

Añadir capa de Bing:

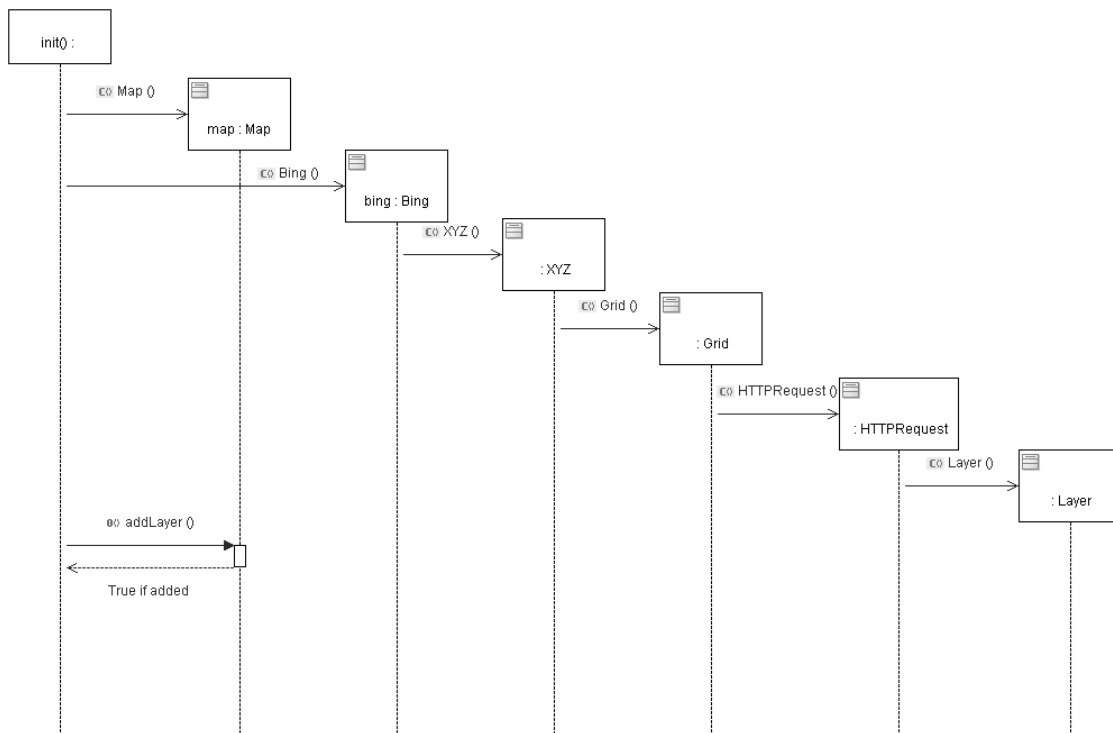


Figura C - 3 Diagrama de secuencia con el proceso de añadir una capa de tipo Bing al mapa en OpenLayers.

Añadir capa de Google:

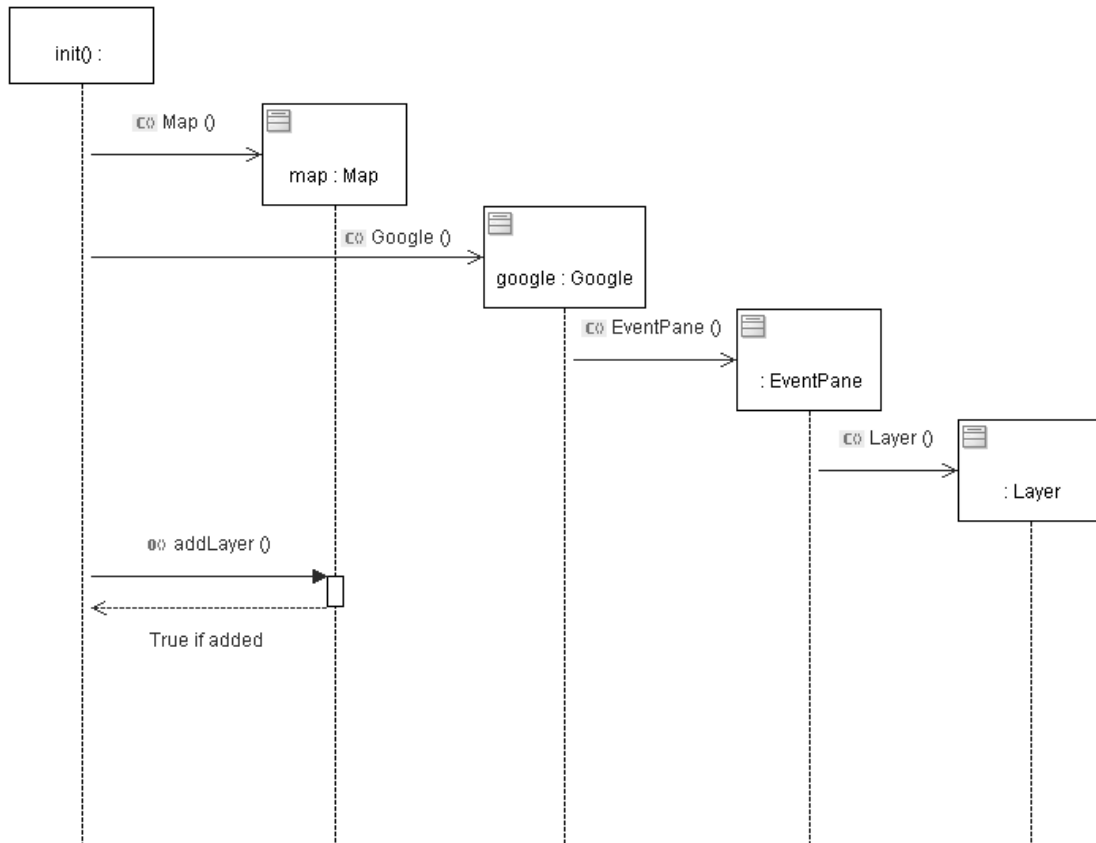


Figura C - 4 Diagrama de secuencia con el proceso de añadir una capa de tipo Google al mapa en OpenLayers.

Añadir capa de OSM:

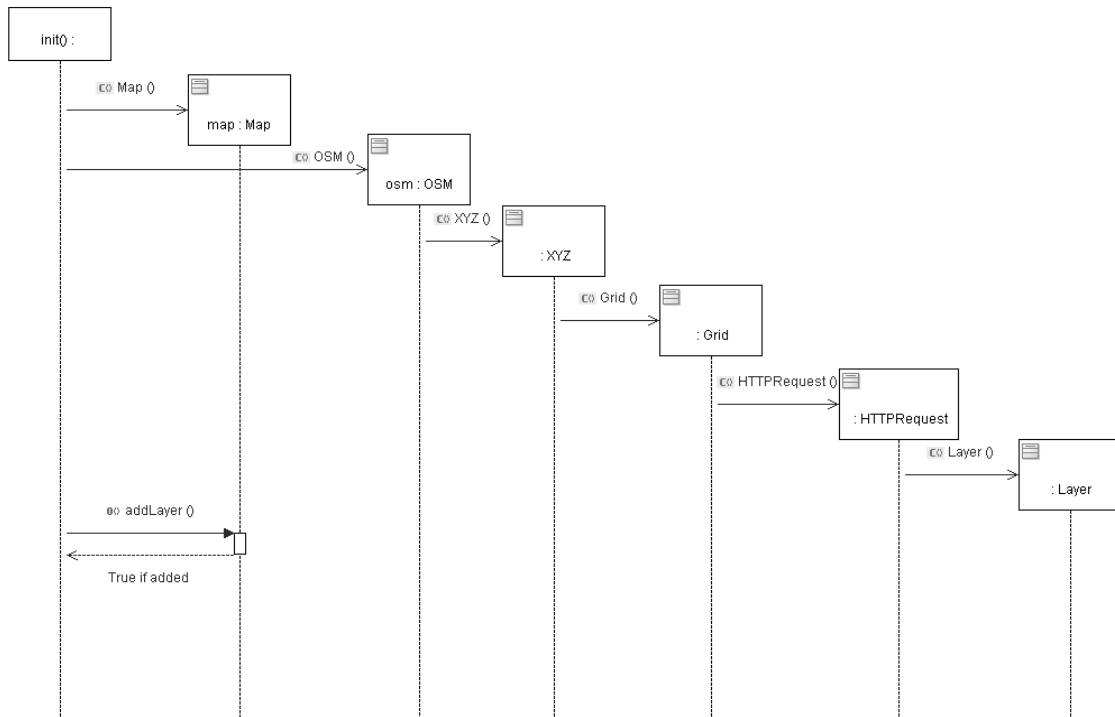


Figura C - 5 Diagrama de secuencia con el proceso de añadir una capa de tipo OSM al mapa en OpenLayers.

Añadir capa de WMS:

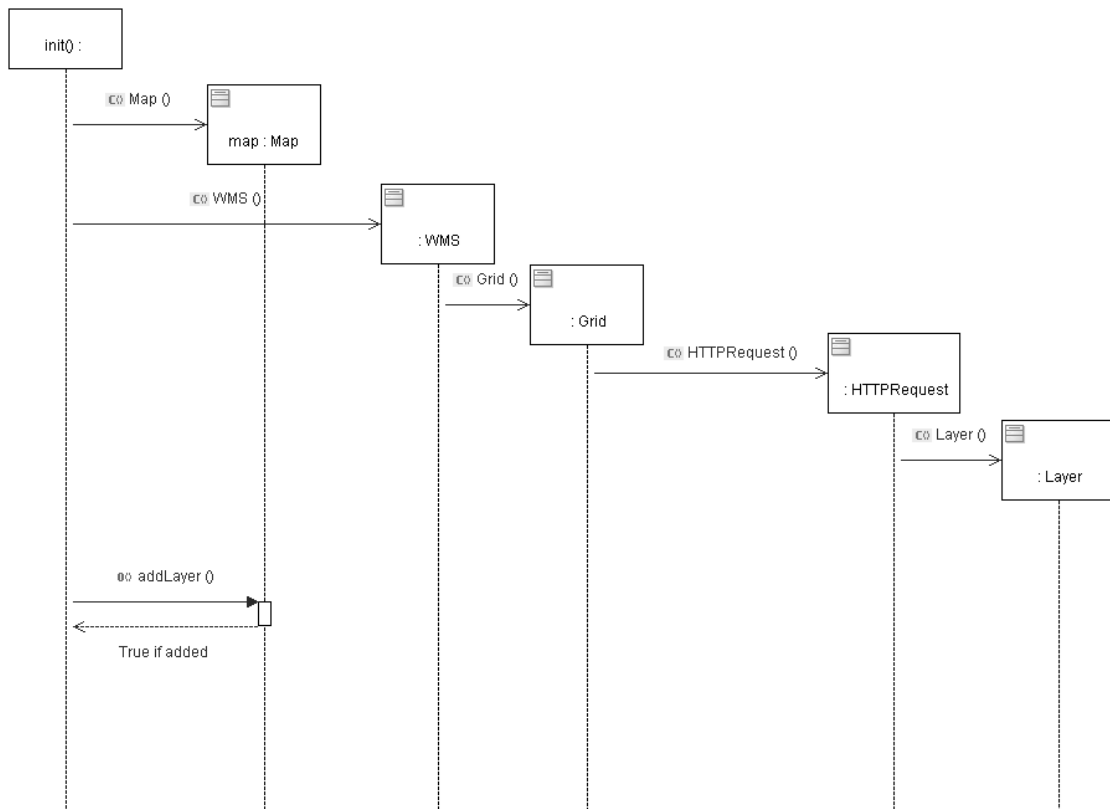


Figura C - 6 Diagrama de secuencia con el proceso de añadir una capa de tipo WMS al mapa en OpenLayers.

Normalmente hoy en día los proveedores de mapas dividen el mundo en una cuadrícula para diferentes niveles de zoom. Cada uno de esos cuadrados, se llama tesela o tile en inglés. La división en teselas permite que se manden sólo las que sean necesarias de modo que la comunicación sea más eficiente, tratando de minimizar los consumos de ancho de banda requeridos por el envío y recepción de las teselas. Gracias a esto se puede ofrecer una mejor experiencia a los usuarios, al permitir mayor fluidez en la utilización del servicio, además de poder llegar a más usuarios al minimizar el ancho de banda requerido.

A continuación se muestran las secuencias que se producen a la hora de cargar las teselas de un mapa para los proveedores Bing, OSM y para los servicios WMS. Este proceso no se muestra para Google ya que la carga de teselas se realiza de forma interna en la propia librería de Google.

Petición de teselas de Bing:

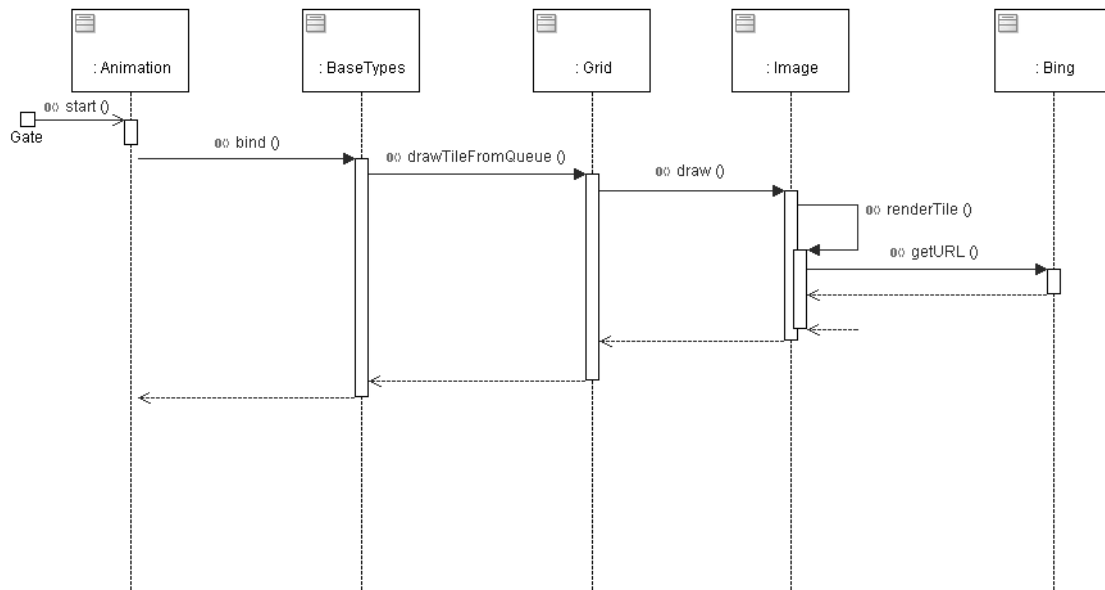


Figura C - 7 Diagrama de secuencia de petición de teselas de Bing.

Petición de teselas de OSM:

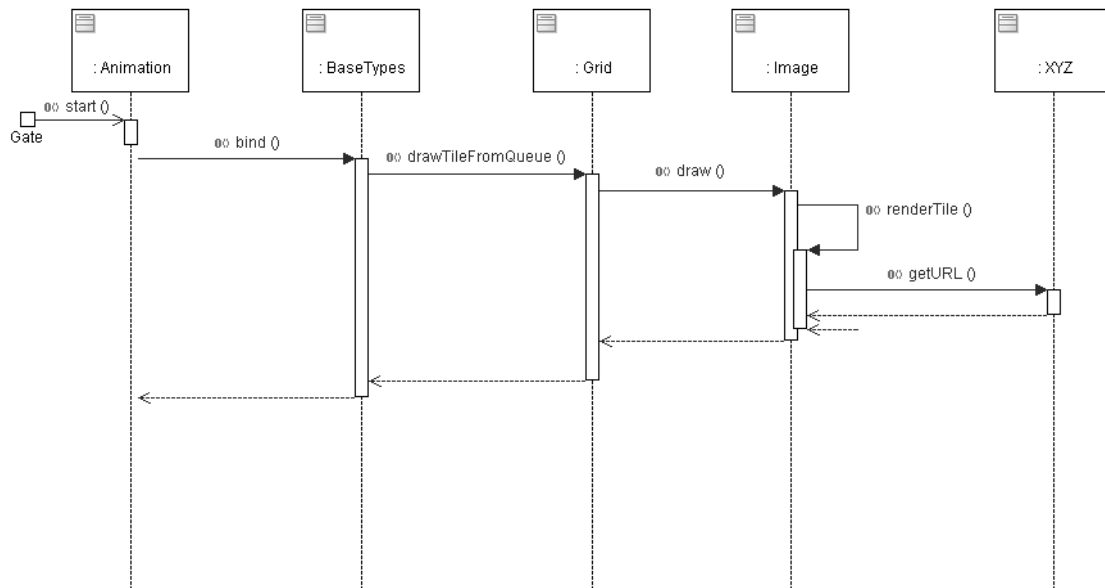


Figura C - 8 Diagrama de secuencia de petición de teselas de OSM.

Petición de teselas de WMS:

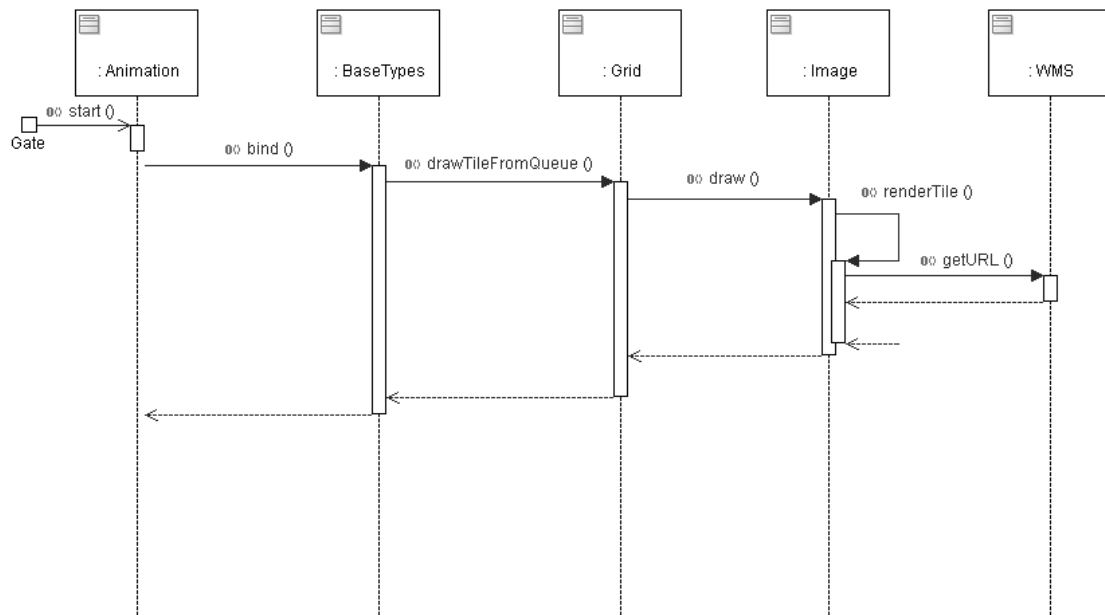


Figura C - 9 Diagrama de secuencia de petición de teselas de WMS.

El caso de Google Maps, funciona de manera diferente al resto de proveedores. Como se puede ver en el siguiente fragmento del diagrama de clases, la clase Google descinde EventPane en lugar de descnder de la clase Grid como el resto de proveedores que se han mostrado en el fragmento del diagrama del comienzo de este apartado. Esto se debe a que no es posible pedir directamente las teselas a Google, por lo que es necesario delegar en su API.

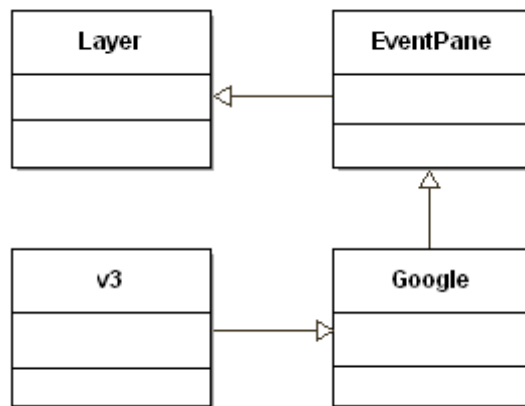


Figura C - 10 Diagrama de clases de la integración de Google en OpenLayers extendiendo de la clase EventPane.

Esto se debe, a que OpenLayers está encapsulando la librería de Google. Para conseguir hacer funcionar esto, lo que se hace es interceptar los eventos que se producen sobre el mapa y pasar la información necesaria a los métodos correspondientes de la API de la librería de Google. Cuando se captura un evento, este se pasa a la clase de Google o a la v3 según la versión de los mapas de Google que se quiera usar (la v3 es la recomendada ya que la versión 2 se encuentra "deprecated"). Para ilustrar el funcionamiento de este sistema, se muestran a continuación

los diagramas de secuencia para los eventos de cambiar la capa visible actual por otra, arrastrar el mapa y para el zoom sobre el mapa.

Google Maps – Cambio de capa visible:

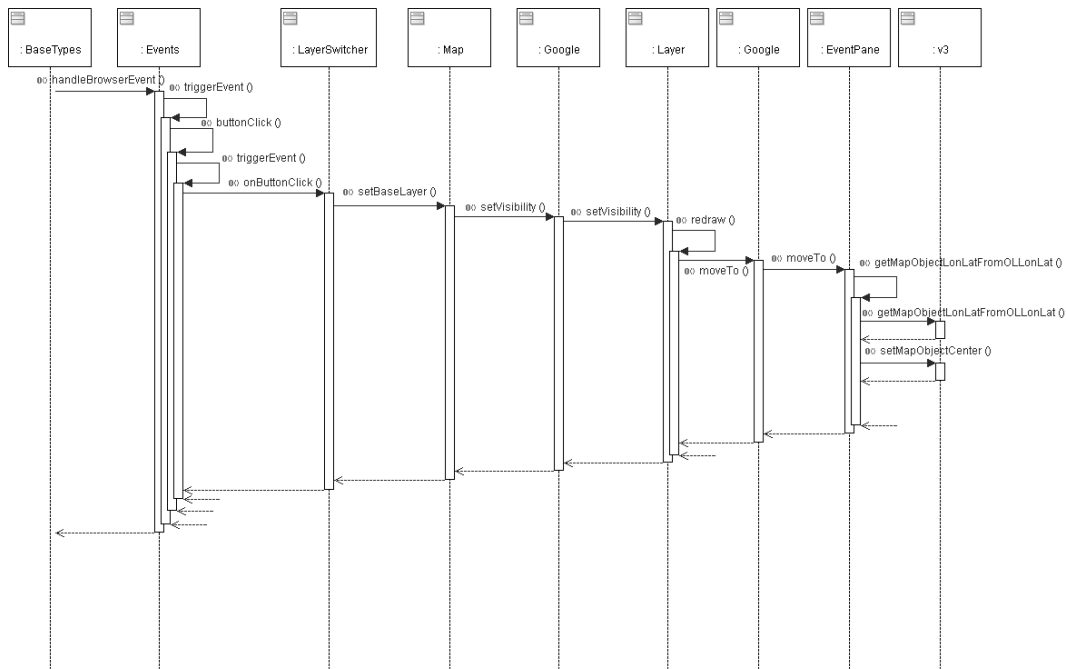


Figura C - 11 Diagrama de secuencia de cambio de capa visible con una capa de Google Maps de OpenLayers.

Google Maps – Arrastrar mapa:

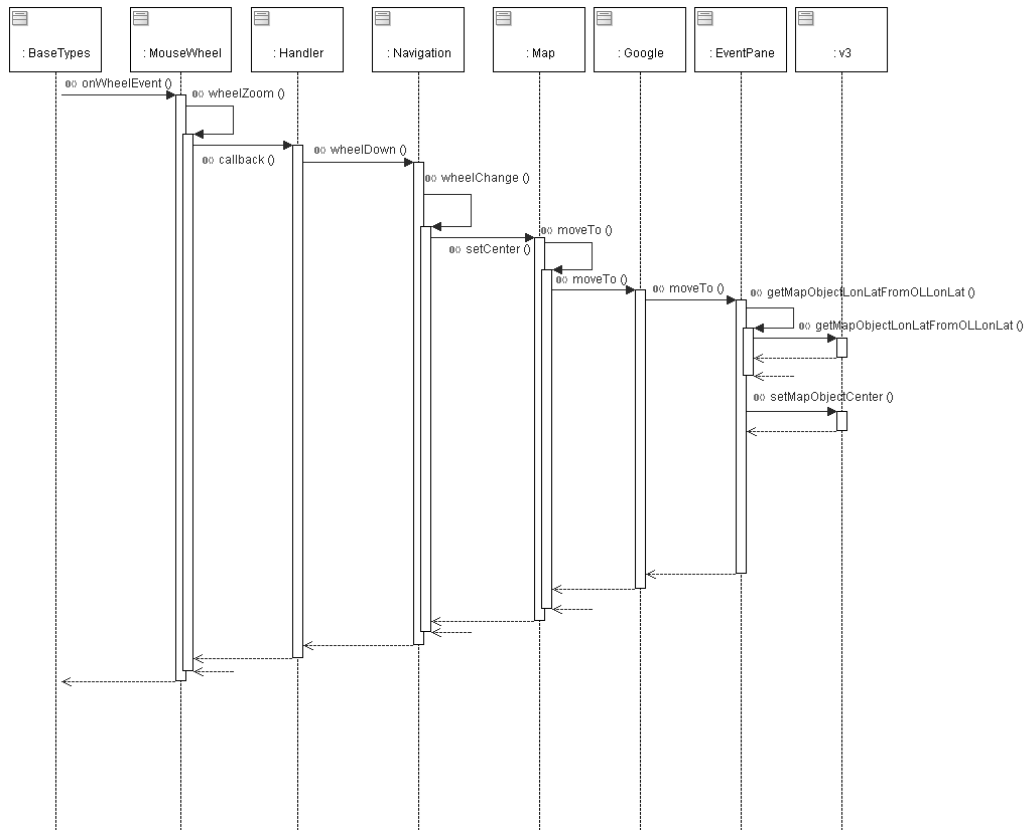


Figura C - 12 Diagrama de secuencia de arrastrar mapa con capa de Google Maps de OpenLayers.

Google Maps – Zoom del mapa:

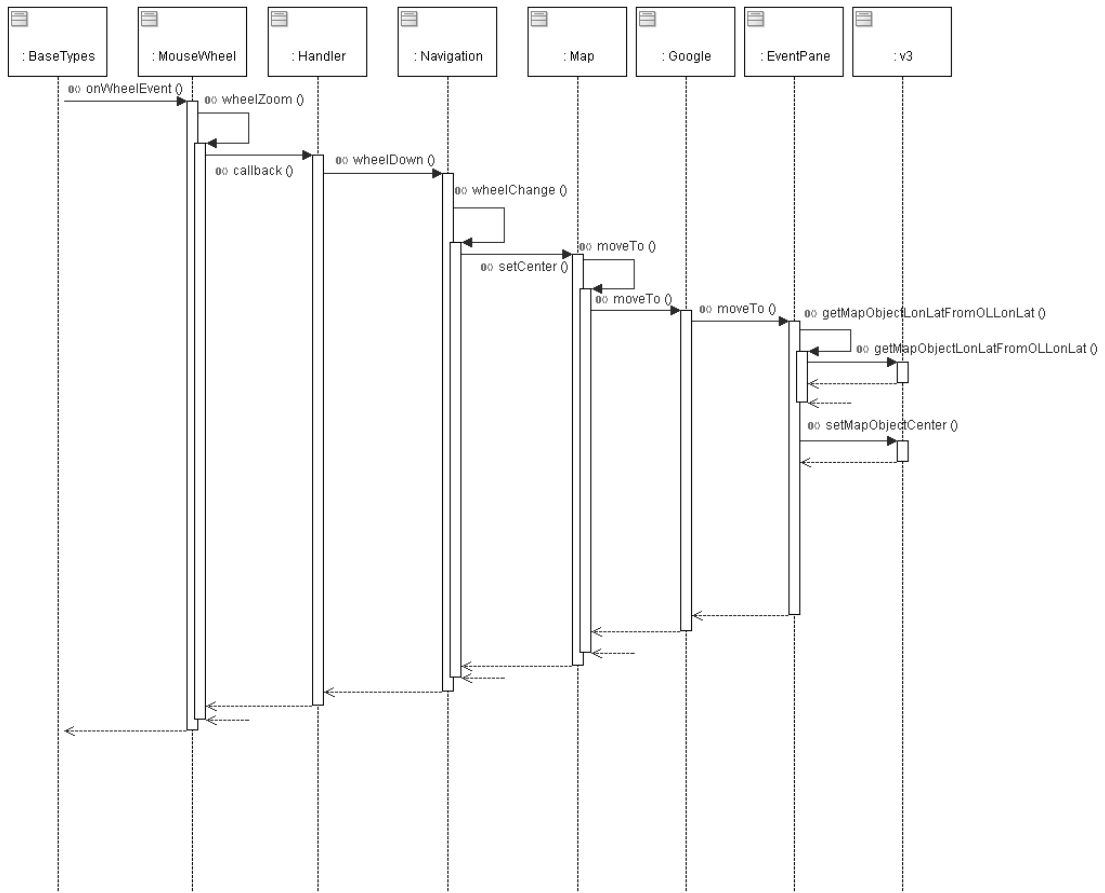


Figura C - 13 Diagrama de secuencia de zoom del mapa con una capa de Google Maps de OpenLayers.

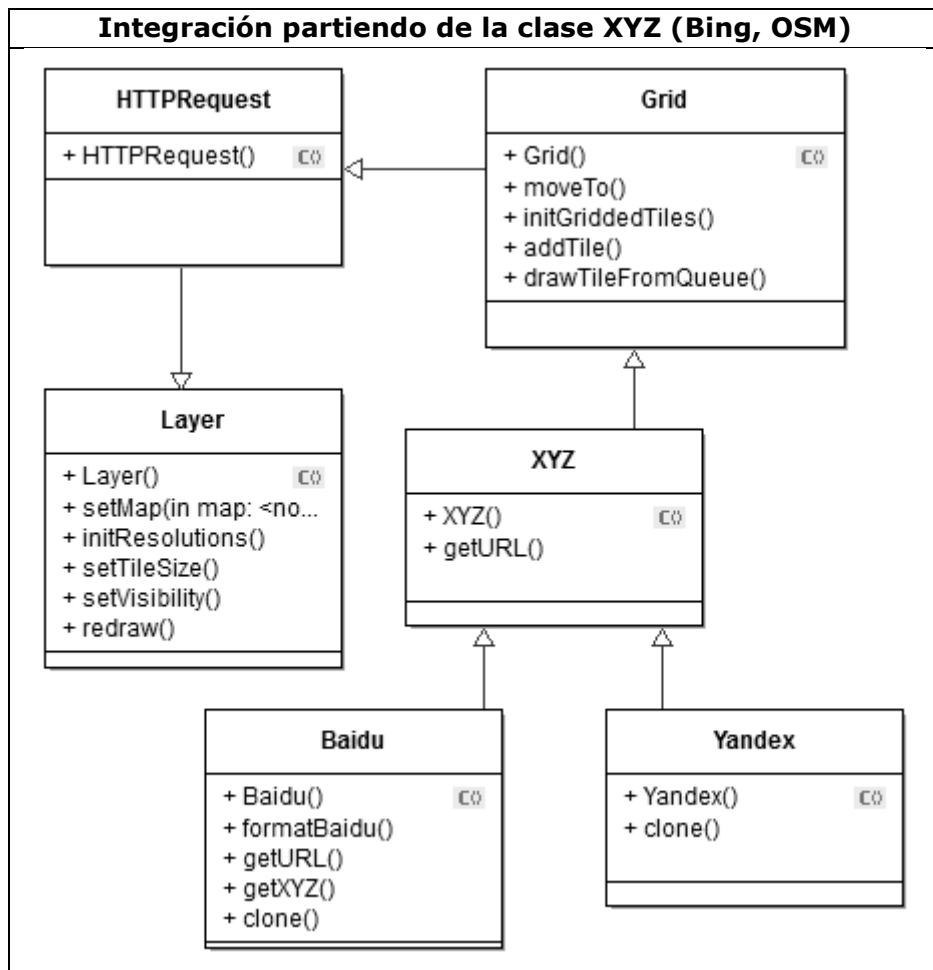
3. Vías para la integración de proveedores

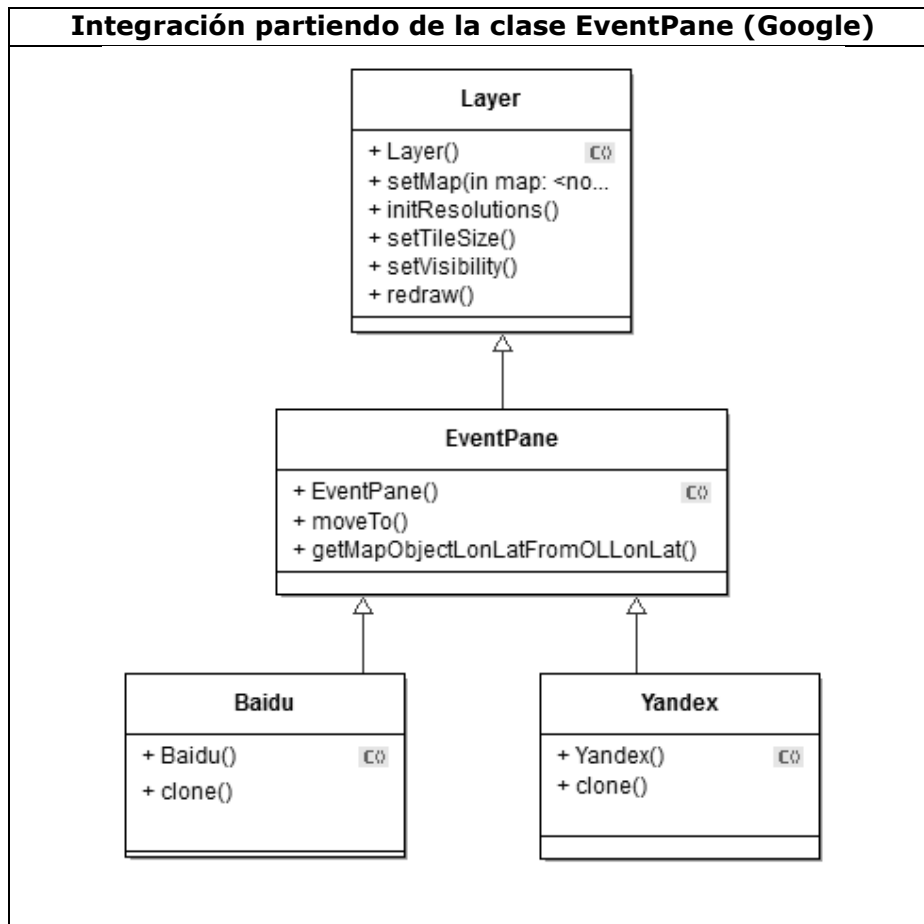
Para los proveedores de mapas que se van a usar como caso de uso, existen principalmente dos vías posibles a usar para la integración. Una es al estilo de las capas para Bing, OSM (extendiendo de XYZ) y la otra vía es usando la misma estructura que usa la librería de Google (extendiendo de EventPane).

La integración al estilo Google se basa en encapsular la librería del proveedor de modo que las llamadas de la librería de OpenLayers desencadenen llamadas a la librería propietaria al interceptar los eventos que se produzcan sobre el mapa.

La integración usando la clase XYZ como punto de partida se basa en la carga directa de las teselas desde el proveedor de mapas, de este modo la integración en la librería es completa y no requiere de librerías externas.

Los siguientes diagramas de clase ilustran ambas posibilidades.





Ambas vías tienen ventajas e inconvenientes, por lo que se trata de escoger aquella que pueda ser más adecuada para el uso que se le va a dar.

Integración a través de XYZ	Integración a través de EventPane
<p style="text-align: center;">Ventajas</p> <ul style="list-style-type: none"> - Integración completa en la librería. - No requiere de código externo. - Mayor control sobre la forma en que se muestran las teselas. - Posibilidad de uso de las capas para escribir información sobre otra capa (si permite transparencia). 	<p style="text-align: center;">Ventajas</p> <ul style="list-style-type: none"> - Los cambios en la forma de hacer las peticiones de las teselas no afectan al funcionamiento del código de OpenLayers. - Permitiría hacer uso de las funciones avanzadas que pueda tener la API del proveedor.
<p style="text-align: center;">Desventajas</p> <ul style="list-style-type: none"> - Si el proveedor cambia la URL dónde se encuentran las teselas o su formato, podría dejar de funcionar y ser necesario modificarla con la nueva información. - Es necesario que el proveedor no tenga prohibido este uso de las teselas en sus condiciones de servicio. 	<p style="text-align: center;">Desventajas</p> <ul style="list-style-type: none"> - Requiere de una librería externa (API del proveedor) para funcionar. - La encapsulación de la librería puede ir en contra de las condiciones de servicio del proveedor (Google no permite la encapsulación de su librería, aunque ha dado su consentimiento por escrito a OpenLayers para su uso de esta forma).

En conclusión, con la información obtenida, será recomendable siempre que sea posible una implementación extendiendo de la clase XYZ ya que permite una integración más completa en la librería. Sólo en los casos en que la integración a través de la clase XYZ no sea posible, se debería realizar la integración a través de la clase EventPane, o bien en aquellos casos en que se quieran usar funciones especiales de la librería del proveedor.

4. Diseño escogido para la integración

Tomando en consideración la información del punto anterior, se usará la clase XYZ como base para la realización de los casos de uso de Yandex y Baidu para así conseguir una integración más completa en la librería. Los prototipos para ambos proveedores han mostrado que la integración usando esta vía es posible por lo que en principio, se descartaría la integración usando la clase EventPane por el coste adicional de esfuerzo que supondría su uso y la poca ventaja que ofrecería el mismo.

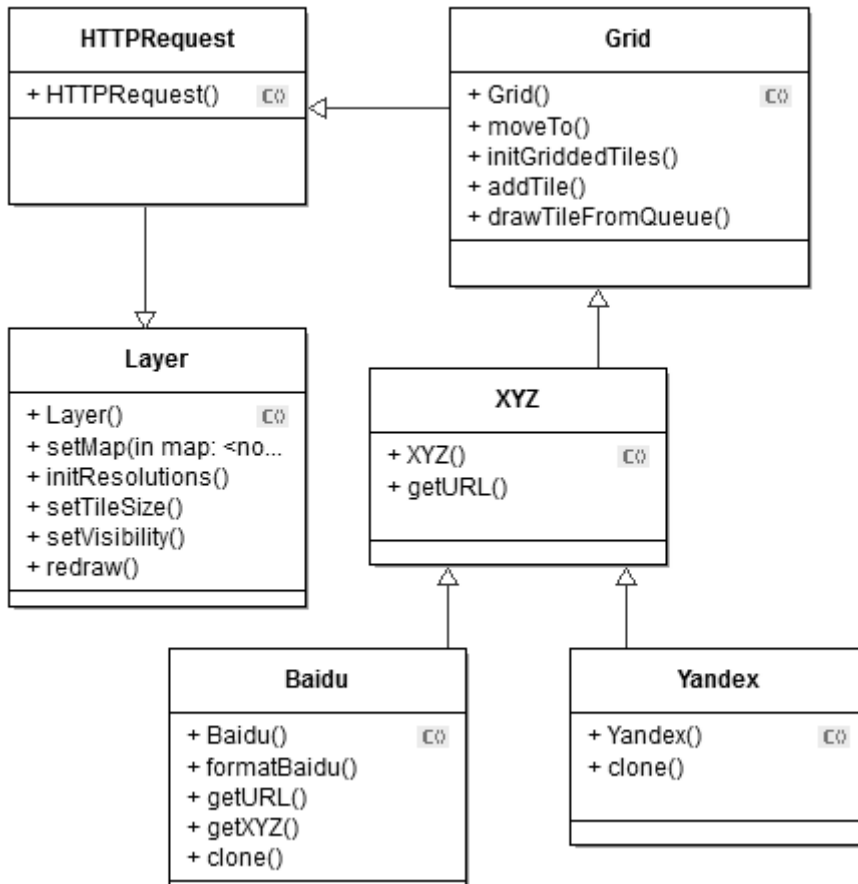


Figura C - 14 Diseño base escogido para la integración de Yandex y Baidu.

La clase XYZ permite cargar teselas desde una URL especificando un posición X e Y además un valor Z de zoom. Habitualmente los servicios de mapas como OpenStreetMap, Bing, Yandex... estructuran las teselas de modo que la tesela con X=0 e Y=0 quede en la esquina superior izquierda del mapa tal y como se ve en la siguiente imagen.

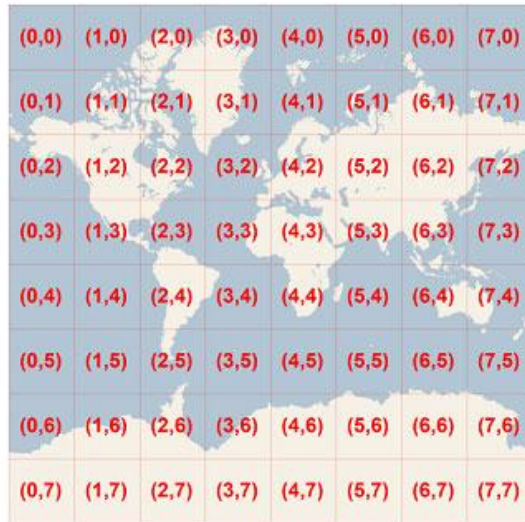


Figura C - 15 Distribución de teselas de servicios de mapas. La imagen corresponde concretamente a la distribución de teselas de Bing, pero Yandex utiliza la misma distribución al igual que OpenStreetMap.

Sin embargo, en el caso de Baidu, las teselas están organizadas de diferente manera. Baidu toma como origen de coordenadas para las teselas la intersección entre la línea del ecuador y el meridiano de Greenwich. Esto implica que las teselas quedan organizadas en 4 cuadrantes, el primer cuadrante con signos positivos y el resto tendrá X o Y negativos según corresponda. Sin embargo, en la URL Baidu no admite las coordenadas de X o Y como número negativos, por lo que en lugar de un signo '-', utilizan la letra M. Esto supone que sea necesario comprobar si los valores X o Y de la tile son negativos y si lo son, obtener el valor absoluto y formar una cadena que contenga una letra M seguida de la coordenada de la tesela. La siguiente imagen muestra el mapa del mundo de Baidu y los diferentes cuadrantes.

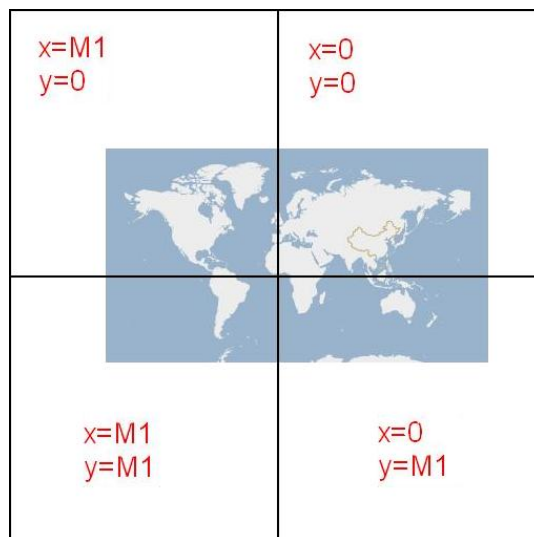


Figura C - 16 Distribución de teselas de Baidu para el nivel de zoom 0.



Figura C - 17 Distribución de teselas de Baidu para el nivel de zoom 3.

En el caso de Yandex las URL son similares a otros servicios, y para la organización de las teselas se usa un sistema muy similar al usado por OpenStreetMap u otros servicios de mapas.

5. Diagrama de clases de la integración

A continuación se va a ver el diagrama de clases de todas las clases que han sido añadidas a OpenLayers 2.12 para realizar la integración de Yandex y Baidu en la librería. El diagrama incluye las clases involucradas en las capas de tráfico de Yandex así como la capa de alertas de Baidu. También se puede ver la integración por dos vías de Baidu (Baidu que extiende XYZ y BaiduEvtPane que extiende de EventPane al igual que Google) usada para validar la integración.

El diagrama anterior muestra las clases que se han implementado. A continuación se muestra información adicional sobre cada clase:

- **BaiduEvtPane**
Clase que corresponde a la implementación de la capa de Baidu que usa su API y que actúa de puente entre la API de Baidu y la clase EventPane de OpenLayers. La implementación de esta clase se ha realizado para tener un punto de comparación con la clase de Baidu que integra directamente las teselas de Baidu.
- **Baidu**
Clase que corresponde a la implementación de la capa de Baidu que integra directamente las teselas del servicio de Baidu.
- **BaiduAlertsVec**
Clase que integra el servicio de alertas de Baidu como una capa de overlay que se puede superponer al mapa.
- **GeoJSONBaidu**
Clase encargada de realizar el parseo de la petición de Baidu en formato JSON creando las Features de OpenLayers que correspondan.
- **Yandex**
Clase que integra las teselas de Yandex tanto del mapa normal, como las de satélite, como las de transparencias de nombres en inglés o en ruso.
- **YandexTrafficLoadVec**
Clase que implementa la capa vectorial de carga de tráfico de Yandex. En esta capa se muestran los semáforos con el nivel de carga, así como los popup con la información del tráfico para una ciudad.
- **YandexUtils**
Clase con varios elementos comunes a las capas de Yandex.
- **YandexTmEvtStrategy**
Estrategia que actualiza las Features de OpenLayers que contienen la información del tráfico cada vez que se produce un cambio en el timestamp usado por el servicio de tráfico de Yandex.
- **TileStrategy**
Clase que se encarga de manejar la estrategia de actualización de teselas de tráfico de Yandex.
- **YandexTrafficVecInfo**
Clase que contiene las Features de OpenLayers relativas a la información extra del tráfico de Yandex con las velocidades en las diferentes calles de la zona visible.
- **GeoJSONYandex**
Clase encargada de realizar el parseo de la petición de tráfico de Yandex en formato JSON creando las Features de OpenLayers que correspondan.

Anexo D. Problemática integración Yandex

1. Introducción

Yandex es una compañía con sede en Moscú, Rusia, y que posee el mayor motor de búsquedas en Internet de Rusia. A fecha 27 de junio de 2013 durante los últimos 3 meses Yandex ha controlado de promedio el 61.7% del tráfico de búsquedas por Internet, seguido por Google con el 26.1%. Información extraída de:

<http://www.liveinternet.ru/stat/ru/searches.html?slice=ru;id=13;id=4;id=5;id=9;id=8;period=month>

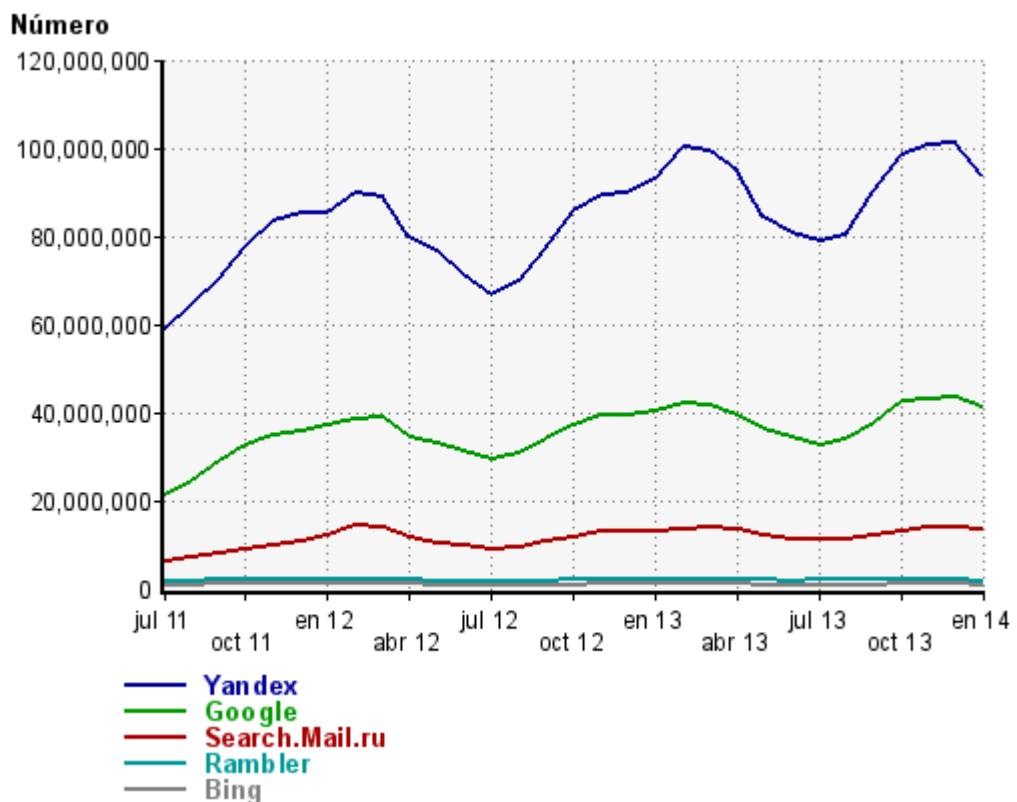


Figura D - 1 Gráfica comparativa de tráfico de varios motores de búsqueda en Rusia.

La popularidad del servicio de Yandex en Rusia lo convierte en un objetivo interesante para su integración en la librería por el posible potencial de mercado que puede tener.

2. Problemática para la integración de Yandex

Yandex dispone también de un servicio de mapas. Existen dos versiones, <http://maps.yandex.ru> y <http://maps.yandex.com>. La versión ".ru" contiene cartografía utilizando el idioma ruso, esto quiere decir que los nombres de países o calles emplean en muchos casos caracteres cirílicos para la nomenclatura. En la versión ".com" en cambio se utiliza el idioma local para los nombres, haciendo que en España las ciudades y las calles aparezcan en

español, en Francia en francés y así sucesivamente. Hasta hace unos meses, la cartografía de ambos servicios estaba claramente diferenciada centrándose cada una en ciertas regiones. El servicio “.ru” tenía una cartografía centrada en Rusia y países próximos a ella, mientras que el servicio “.com” contenía cartografía de países de varias regiones principalmente del hemisferio Norte como Europa, Norte América, y algunas regiones del Norte de Asia. Esto provocaba que por ejemplo Zaragoza apareciera con escasa información en el servicio en ruso (“.ru”) mostrando sólo la región de la ciudad y las carreteras principales sin mostrar ninguna calle, mientras en el servicio en inglés (“.com”) Zaragoza aparecía con la información completa incluyendo las calles de la ciudad.

Estas diferencias entre ambos servicios fueron subsanadas recientemente haciendo que ambos servicios dispongan actualmente de la misma cartografía haciendo que en el caso de estar en la versión en ruso muestre teselas en el idioma de la región cuando no están disponibles en ruso. En este PFC se intentó solucionar esta problemática usando lógica en el cliente que determinase el error al cargar una tesela en inglés o en ruso y probar entonces a cargarla en el otro idioma.

El servicio de Yandex mapas está construido usando la proyección Mercator (http://en.wikipedia.org/wiki/Mercator_projection), al igual que otros servicios como Google. Sin embargo, servicios como Google o Bing usan la proyección Mercator esférica en la que se considera que la Tierra es una esfera. Yandex en cambio, utiliza la proyección Mercator elíptica en la que la Tierra se considera que es un elipsoide. En cuanto al sistema de coordenadas Yandex usa WGS-84 al igual que otros servicios como Google o Bing. Yandex considera un radio ecuatorial de 6378.137 km y un radio polar de 6356.752 km. La proyección Mercator tiene el problema de que tiende a deformar objetos conforme estos están más cerca de los polos.

Las dos imágenes siguientes extraídas de la página de la Wikipedia (http://en.wikipedia.org/wiki/Mercator_projection) ilustran la problemática de la deformación de esta proyección. La imagen de la izquierda muestra como la deformación se incrementa conforme el elemento está más cerca de los polos. Como ejemplo característico se puede ver la imagen de la derecha que muestra el tamaño real de Groenlandia comparado con el de Australia, mientras que en la proyección Mercator aparece que Groenlandia es mucho mayor que Australia (Imagen de la izquierda).

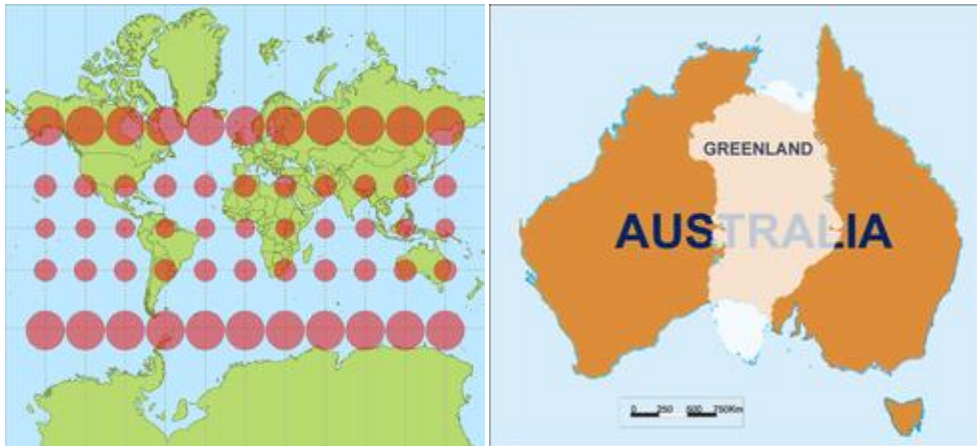


Figura D - 2Distorsión de la proyección Mercator. Las dos imágenes pertenecen a la Wikipedia.

3. Integración de Yandex en OpenLayers

Se encontraron y valoraron dos posibles opciones de integración de Yandex en la librería. La primera posibilidad era integrar directamente las teselas que usa el servicio dentro de la capa de Yandex y la otra opción era integrar Yandex usando su propia API (así está integrado Google en OpenLayers). La mejor opción es la de integrar las teselas del servicio directamente ya que da un mayor control sobre el funcionamiento de la capa. Dado que Yandex no ofrece públicamente la URL de las teselas que usa en el servicio, esta información tuvo que ser averiguada mediante el análisis de las peticiones en segundo plano que realiza su API. Para realizar este análisis se utilizó el plugin para Firefox Tamper Data 11.0.1. Entre todo el tráfico analizado, se encontraron URL s correspondientes a las teselas de varios servicios así como las URLs necesarias para implementar la capa de tráfico. La siguiente tabla muestra las URL recopiladas en los diferentes análisis del tráfico para varios servicios. En las URL el parámetro "l" representa la capa, "v" la versión, "lang" el lenguaje de la capa (algunas capas ignoran este parámetro como puede ser la capa de satélite), [x,y,z] representa las coordenadas de las teselas en las URLs que corresponde a teselas. Algunas URL relacionadas con el servicio de tráfico tienen un parámetro adicional "tm" que es un timestamp y corresponde a un momento de tiempo que se obtiene mediante otra petición. Una de las URLs corresponde a un servicio en el que da el timestamp actual para ser usado con otras peticiones de tráfico.

Lista de URLs obtenidas de diferentes servicios que han sido integrados:

Teselas del mapa inglés:

http://vec00.maps.yandex.net/tiles?l=map&v=2.44.0&lang=en_US&x=0&y=0&z=0

http://vec01.maps.yandex.net/tiles?l=map&v=2.44.0&lang=en_US&x=0&y=0&z=0

http://vec02.maps.yandex.net/tiles?l=map&v=2.44.0&lang=en_US&x=0&y=0&z=0

http://vec03.maps.yandex.net/tiles?l=map&v=2.44.0&lang=en_US&x=0&y=0&z=0

Teselas del mapa ruso:

http://vec00.maps.yandex.net/tiles?l=map&v=2.44.0&lang=ru_RU&x=0&y=0&z=0

http://vec01.maps.yandex.net/tiles?l=map&v=2.44.0&lang=ru_RU&x=0&y=0&z=0

http://vec02.maps.yandex.net/tiles?l=map&v=2.44.0&lang=ru_RU&x=0&y=0&z=0

http://vec03.maps.yandex.net/tiles?l=map&v=2.44.0&lang=ru_RU&x=0&y=0&z=0

Teselas capa satélite:

http://sat00.maps.yandex.net/tiles?l=sat&v=3.101.0&lang=ru_RU&x=0&y=0&z=0
http://sat01.maps.yandex.net/tiles?l=sat&v=3.101.0&lang=ru_RU&x=0&y=0&z=0
http://sat02.maps.yandex.net/tiles?l=sat&v=3.101.0&lang=ru_RU&x=0&y=0&z=0
http://sat03.maps.yandex.net/tiles?l=sat&v=3.101.0&lang=ru_RU&x=0&y=0&z=0

Teselas capa nombres ruso:

http://vec00.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=ru_RU&x=0&y=0&z=1
http://vec01.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=ru_RU&x=0&y=0&z=1
http://vec02.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=ru_RU&x=0&y=0&z=1
http://vec03.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=ru_RU&x=0&y=0&z=1

Teselas capa nombres inglés:

http://vec00.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=en_US&x=15&y=11&z=5
http://vec01.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=en_US&x=15&y=11&z=5
http://vec02.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=en_US&x=15&y=11&z=5
http://vec03.maps.yandex.net/tiles?l=skl&v=2.2.3&lang=en_US&x=15&y=11&z=5

Teselas servicio de tráfico:

http://jgo.maps.yandex.net/1.1/tiles?l=trf&lang=ru_RU&tm=<valorTimestamp>&x=309&y=160&z=9

Timestamp para los servicios de tráfico:

http://api-maps.yandex.ru/services/traffic-info/1.0/?format=json&lang=ru_RU
http://api-maps.yandex.ru/services/traffic-info/1.0/?format=json&lang=en_US

Features para capa de tráfico:

http://jgo.maps.yandex.net/1.1/tiles?l=trj&lang=en_US&x309&y=160&z=9&tm=<valorTimestamp>

Una vez obtenidas las URLs, el siguiente paso consistió en ajustar las coordenadas para que un punto del mapa correspondiese con sus coordenadas reales en el mundo. OpenLayers necesita que una capa especifique el valor de su máxima extensión y la máxima resolución para poder mostrarse correctamente sus coordenadas. Además es necesario que la capa especifique la proyección en la que está para poder convertir coordenadas de un sistema a otro. A continuación se van a explicar cada uno de esos valores y como se han calculado.

maxExtent (máxima extensión): Este parámetro representa los límites de la extensión de la capa por la izquierda, arriba, a la derecha y abajo. En el caso de Yandex la capa cubre todo el mundo, por lo que para calcular su extensión total podemos usar la fórmula que sirve para calcular el perímetro de un círculo $W = 2\pi R$ siendo W el ancho del mundo y R el radio del ecuador usado por Yandex. El resultado de este cálculo será:

$$W = 2\pi R = 2 * \pi * 6378137 = 40075016.685578$$

Si dividimos por dos el valor de la fórmula anterior tenemos el valor 20037508.342789244 que corresponde con los límites del mundo en las cuatro direcciones desde un punto central por lo que el valor del maxExtent de la librería OpenLayers quedaría como:

maxExtent: *new OpenLayers.Bounds(-20037508.342789244, -20037508.342789244, 20037508.342789244, 20037508.342789244)*

maxResolution (Resolución máxima): Este parámetro representa la equivalencia entre pixeles y metros en el mundo real. En el caso de Yandex su valor máximo, se dará para el nivel de zoom 0 que es el que dispone de una única tesela de 256 pixeles que engloba todo el mundo.

$$\text{maxResolution} = \frac{40075016.685578}{256} = 156543.033928041$$

projection (Proyección en la que está la capa): El valor de este parámetro indica la proyección en la que está la capa. Este valor es necesario, sobre todo para trabajar con capas de otros proveedores en distintas proyecciones. Consultando la página de Yandex Maps, lo único que se puede encontrar es que usan la proyección Elliptical Mercator, pero no dan más información. Al no tener más información no era posible establecer un valor para este parámetro. OpenLayers no soporta por defecto la proyección usada por Yandex, por lo que era necesario buscar más información. Tras realizar múltiples análisis y búsquedas de información, hubo dos documentos que aportaron datos para añadir la proyección Mercator elíptica a OpenLayers. Uno de ellos es <http://mercator.myzen.co.uk/mercator.pdf> referenciado en la página de la Wikipedia sobre la proyección Mercator. El otro documento que resultó el más útil para llevar a cabo la integración ya que incluía fórmulas detalladas y ejemplos de uso que permitieron ajustar correctamente los valores de las fórmulas para que los cálculos fueran correctos, es el libro SNYDER, JOHN P., *Map Projections – A Working Manual*, U.S. Geological Survey Professional Paper 1395, pp 38-47 que se puede consultar en el siguiente enlace de publicaciones del USGS:

<http://pubs.er.usgs.gov/publication/pp1395>

Se trata de un libro publicado por el USGS (United States Survey Group) que data del año 1987. El libro se encuentra a disposición del público por parte del propio USGS en el enlace anterior. En ese enlace se puede descargar el libro en versión PDF que como se puede ver el USGS lo ha sacado de un libro con los sellos de la librería del U.S. Bureau of Mines. En concreto la información usada se encuentra en la página 44 y siguientes del libro en cuestión. Durante la búsqueda de información para la implementación del código para convertir del sistema de coordenadas usado por Yandex a otros, la proyección Mercator elíptico apareció en alguna ocasión referenciada como EPSG:3395 (<http://spatialreference.org/ref/epsg/3395/>). Con un sistema de transformación ya en funcionamiento se realizaron pruebas para contrastar la hipótesis de que el código EPSG:3395 realmente correspondía con el sistema usado por Yandex. Tras realizar pruebas con la librería proj4js de conversión de proyecciones, se pudo comprobar que efectivamente las coordenadas convertidas usando las fórmulas extraídas del libro de USGS coincidían

perfectamente con las calculadas usando la librería proj4js especificando la proyección EPSG:3395. Por lo que en el proyecto finalmente se dejó integrada la librería proj4js para su uso. Además las funciones extraídas del libro de USGS se dejaron en el código para que sólo sean usadas en caso de que la librería de proj4js no estuviese disponible, así el código seguiría funcionando aunque no estuviese.

Como ya se ha mencionado antes, Yandex dispone de teselas que contienen únicamente nombres y carreteras siendo el resto transparente. Estas teselas permiten superponer la información de las ubicaciones sobre imágenes de satélite. En el análisis inicial que se hizo del servicio, se pudo comprobar que esas teselas están disponibles para el parámetro lang con valor "ru_RU" y "en_US". Ahora bien, como ya se ha comentado las teselas en ambos lenguaje contaban con limitaciones. La versión para inglés no disponía de teselas para niveles de zoom entre 0 y 4, además esta versión sólo disponía de información geográfica de países en el hemisferio norte y al oeste de Rusia. Las teselas se cargaban en "ru_RU" en el servicio en ruso de Yandex mientras que las de "en_US" en el servicio ".com". Dado que ninguno de los servicios ofrecía una solución completa, se realizó un intento de unificar teselas de ambos servicios en la integración de Yandex en la librería. Para ello, se realizaron las modificaciones necesarias en la tesela de OpenLayers que usa la capa de Yandex de modo que en caso de que se produjese error al cargar una tesela se intentase cargar en el otro idioma. Esta solución funcionaba razonablemente bien por ejemplo al hacer zoom sobre Zaragoza desde el mapa en ruso de modo que cambiaba automáticamente a cargar las teselas de la versión en inglés. Sin embargo, esta solución no era suficientemente buena en las zonas colindantes entre ambos servicios. La siguiente imagen ilustra el problema que ocurría en zonas entre Europa y Rusia.

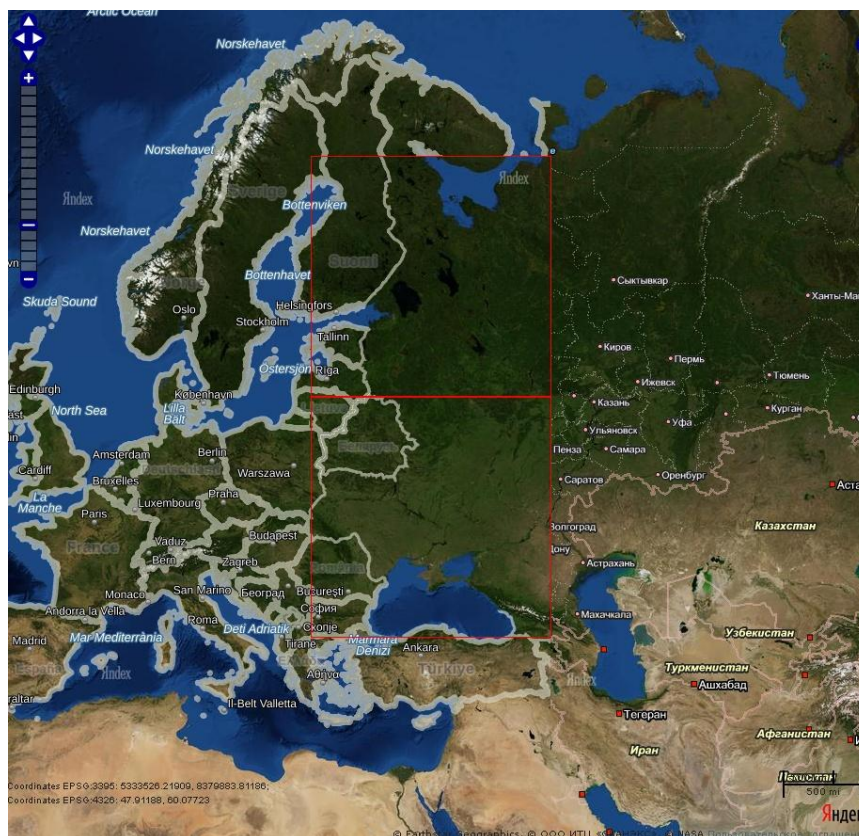


Figura D - 3 Problemática de combinación de teselas de Yandex en ruso e inglés para la capa de overlay.

El problema mostrado en la imagen anterior consistía en que las teselas del servicio en inglés no contenían la misma información que las del servicio en ruso. El servicio en inglés tenía omitida parte de la información dejando un hueco vacío. Dado que la tesela si existía, pero estaba parcialmente vacía no existía una forma limpia de realizar este ajuste. Como se puede ver en la imagen anterior, por ejemplo Moscú no aparecía dibujado en las teselas en inglés por lo que al juntar las teselas de la capa de inglés y la de ruso quedaba la zona vacía que se comentaba. Desde la actualización que hizo Yandex de su servicio y que ya se ha comentado, esto dejó de ser necesario ya que ambos servicios compartían su cartografía.

En cuanto a las teselas de nombres en ruso cabe señalar que tampoco tenían información completa, ya que no disponían de información detallada para algunos niveles de zoom en ciertas zonas. Por ejemplo, en el caso de Zaragoza, estas teselas sólo estaban disponibles para los niveles de zoom 0 al 13. A partir de este punto era necesario cambiar a la capa de nombres en inglés que si disponía de esa información. Dada esta situación se realizó una capa híbrida que intentaba unir ambos grupos de teselas. El funcionamiento de esta capa se detalla a continuación. En primer lugar es necesario especificar el idioma de preferencia para la capa (inglés o ruso). Este idioma de preferencia será el que se use para cargar las teselas, sin embargo, si al intentar cargar la tesela en ese idioma, se encuentra que no está disponible, se intenta cargar en el otro idioma. Esto ofrece una solución, funcional

aunque con algunos problemas en la zona limítrofe con Rusia por el motivo ya mencionado.

Como ya se ha comentado esta solución era funcional, aunque presentaba ciertas deficiencias por las características del servicio de Yandex en ambos idiomas, sin embargo desde la actualización del servicio de Yandex mencionada, ha dejado esas capas híbridas sin ningún sentido ya que ahora es el servicio de Yandex el que está realizando esa funcionalidad al haber unificado la cartografía.

Anexo E. Problemática integración Baidu

1. Introducción

Baidu es una compañía de servicios de China, entre ellos, ofrece un servicio de búsquedas y uno de mapas. Esta compañía ocupa el quinto lugar en tráfico a nivel mundial y el primero en China. El siguiente gráfico compara Baidu.com con el servicio de Google en China (Google.cn) entre enero y junio del 2013. Como se puede ver, Baidu tiene mucho más tráfico dentro de China que el servicio equivalente de Google (Fuente: <http://www.alexapro.com/siteinfo/>).



Figura E - 1 Comparativa de tráfico de Baidu y Google China.

Si se realiza la comparación entre Baidu con Google a nivel mundial, Google si tiene mucho más tráfico. Sin embargo, en el ranking de tráfico a nivel mundial, Baidu se encuentra en el 5º puesto.

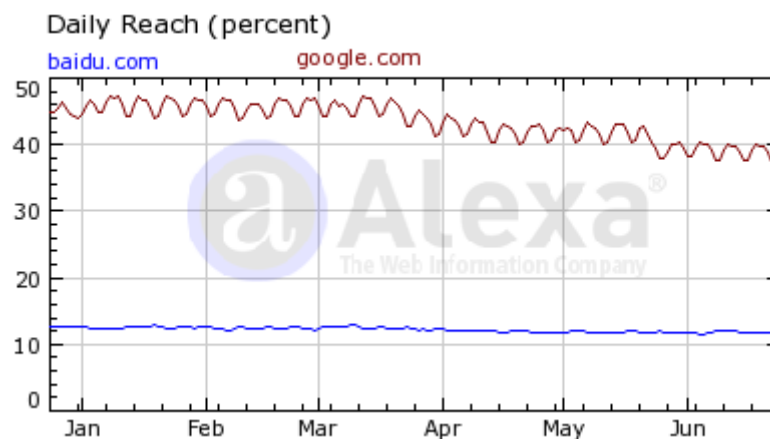


Figura E - 2 Comparativa de tráfico de Baidu y Google.com.

Esto convierte a Baidu en un objetivo interesante para su integración en la librería, debido al potencial de mercado que tiene y el público potencial al que puede llegar. Sin embargo existe una problemática asociada a su integración que se analizará en el siguiente punto.

2. Problemática para la integración de Baidu

Baidu es uno de los mayores proveedores de mapas de China, y como ya se ha mencionado se encuentra relativamente próximo a los niveles de tráfico de Google a nivel mundial. Sin embargo, Baidu está ubicado en China y por tanto cumple con las normativas del gobierno chino al igual que cualquier otro proveedor que quiera operar en China. Esto tiene una consecuencia importante de cara a la integración de Baidu en la librería OpenLayers.

El problema consiste en que el gobierno chino obliga a todos los proveedores de mapas a añadir un desplazamiento no lineal y no constante (GCJ-02 Fuente: <http://developer.baidu.com/map/question.htm#qa004>) en los mapas de forma que la ubicación real de las cosas queda enmascarada. Este proceso se conoce como encriptación geográfica. Además, para que funcionen correctamente las aplicaciones de hoy en día que usan mapas para posicionamiento y navegación, en China los dispositivos móviles añaden este desplazamiento, de esta manera sus usuarios pueden hacer uso de los mapas, sin embargo las coordenadas de las ubicaciones no coinciden con las coordenadas del sistema internacional usado por los GPS.

Esta situación genera algunos casos particulares que se exploran más a fondo en el paper "*Restricciones al trabajo con información geográfica online en China*" redactado para las JIIDE 2013 que se incluye en el Anexo F. así como la normativa que afecta principalmente al proceso de creación de mapas. En el caso de Baidu el desplazamiento es todavía más notable, ya que además del desplazamiento obligado por el gobierno, añaden un segundo desplazamiento propio llamado BD-09 (Fuente: <http://developer.baidu.com/map/question.htm#qa004>) que altera todavía más las coordenadas del mapa.

En China el ejército dispone de mapas que siguen las coordenadas internacionales, pero para que cualquier otro use el sistema de coordenadas internacionales en mapas, es necesario que se solicite una autorización al gobierno Chino justificando la necesidad de su uso.

En la información de su API indican que disponen de una API especial de conversión de coordenadas. Se consultó con ellos y facilitaron el documento que se puede ver en el 0 Se realizaron pruebas sobre ese interfaz que se van a mostrar a continuación. Para probar el interfaz se realizó la conversión de coordenadas desde Google con su capa de satélite, OpenStreetMap y Bing. Para los tres servicios se cogió la misma ubicación geográfica que se trata de una esquina de la zona de la Ciudad Prohibida en Pekín.

Coordenadas en Google satélite



Figura E - 3 Coordenadas en el servicio de Google 116.39561 | 39.912331

URL Solicitud:
http://api.map.baidu.com/ag/coord/convert?from=2&to=4&x=116.39561&y=39.912331
Respuesta:
<code>{"error":0,"x":"MTE2LjQwODIyNzM1MzQ=","y":"MzkuOTIwMDgzNzgzMjA1"}</code>

El interfaz devuelve las coordenadas codificadas en Base 64, por lo que al decodificar se obtiene:

"x":"MTE2LjQwODIyNzM1MzQ" → 116.4082273534
"y":"MzkuOTIwMDgzNzgzMjA1" → 39.920083783205

Si pasamos esas coordenadas al servicio de Baidu coincide la posición en ambas ubicaciones

<http://map.baidu.com/?latlng=39.920083783205,116.4082273534&title=A&content=A&autoOpen=false&l=>



Figura E - 4 Posición de las coordenadas en el servicio de Baidu.

Coordenadas en Bing satélite



Figura E - 5Coordenadas en el servicio de Bing 116.395675 | 39.912335

URL Solicitud:
http://api.map.baidu.com/ag/coord/convert?from=0&to=4&x=116.39562&y=39.91232
Respuesta:
<pre>{"error":0,"x":"MTE2LjQwODIzNzQwNDU0","y":"MzkuOTIwMDcyOTE3NDMy"}</pre>

Al decodificar se obtiene:

"x":"MTE2LjQwODIzNzQwNDU0" →116.40829249488

"y":"MzkuOTIwMDcyOTE3NDMy" →39.920087733668

Si pasamos esas coordenadas al servicio de Baidu coincide la posición en ambas ubicaciones

<http://map.baidu.com/?latlng=39.920087733668,116.40829249488&title=A&content=A&autoOpen=false&l=>



Figura E - 6 Posición de las coordenadas en el servicio de Baidu.

Coordenadas en OpenStreetMap

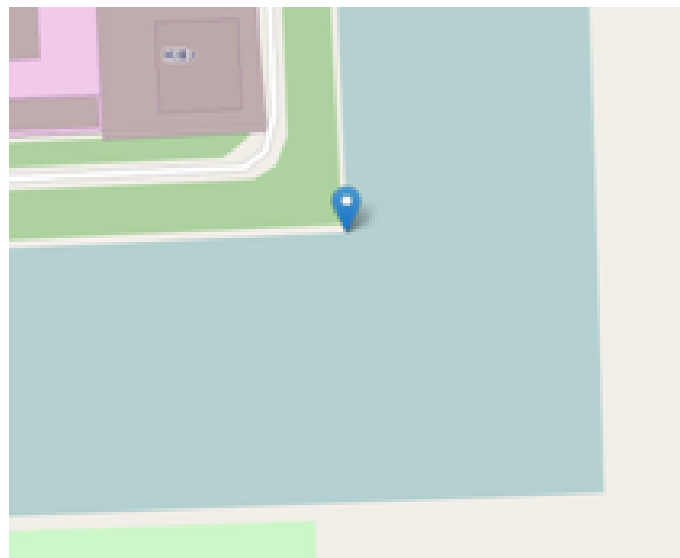


Figura E - 7 Coordenadas en el servicio de OpenStreetMap 116.39562 | 39.91232

URL Solicitud:
http://api.map.baidu.com/ag/coord/convert?from=0&to=4&x=116.39562&y=39.91232
Respuesta:
<pre>{"error":0,"x":"MTE2LjQwODIzNzQwNDU0","y":"MzkuOTIwMDcyOTE3NDMy"}</pre>

Al decodificar se obtiene:

"x":"MTE2LjQwODIzNzQwNDU0" → 116.40823740454

"y":"MzkuOTIwMDcyOTE3NDMy" → 39.920072917432

Si pasamos esas coordenadas al servicio de Baidu coincide la posición en ambas ubicaciones

<http://map.baidu.com/?latlng=39.920072917432,116.40823740454&title=A&content=A&autoOpen=false&l=>



Figura E - 8 Posición de las coordenadas en el servicio de Baidu.

Sin embargo, aunque la conversión de coordenadas EPSG:4326 (GPS) a coordenadas de Baidu funciona en su API de conversión, no se hace mención alguna a la conversión de coordenadas de Baidu a EPSG:4326, se consultó nuevamente con Baidu al respecto, pero su respuesta fue que lamentablemente no disponían de la función inversa de conversión de coordenadas, por lo que no es posible la conversión de coordenadas de Baidu a EPSG:4326. Dada esta información no se pudo implementar el uso de esta interfaz para el cálculo del centro del mapa al cambiar la capa de un servicio a otro, ya que el paso sólo podría ser unidireccional. Además, aunque se decidiese integrar esta conversión de coordenadas en sentido unidireccional, el número de peticiones que se le pueden hacer a esa API está limitado, por lo que tampoco podría ser una solución para el uso planteado por el número de peticiones que podrían llegar a hacerse.

3. Integración de Baidu en OpenLayers

A la hora de integrar Baidu en OpenLayers, el primer paso es echar un vistazo al servicio de Baidu. Usando el plugin Tamper Data para Firefox al igual que en el caso de Yandex, examinaron las peticiones que hace su API para poder obtener las URLs de las teselas. Baidu dispone en su API, de tres grupos de teselas: Las teselas de la capa que contienen la información de calles sin transparencia, las de la capa de satélite que contiene fotografías aéreas y por último una capa que se superpone a la vista de satélite para aportar información de las calles sobre esa vista. Las URLs que han sido recogidas mediante el análisis del tráfico se detallan en la tabla que se muestra a continuación:

Lista de URLs obtenidas de diferentes servicios que han sido integrados:
--

Teselas del mapa:

http://shangetu0.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44

http://shangetu1.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44

http://shangetu2.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44

<http://shangetu3.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>
<http://shangetu4.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>
<http://shangetu5.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>
<http://shangetu6.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>
<http://shangetu7.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>
<http://shangetu8.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>
<http://shangetu9.map.bdimg.com/it/u=x=5;y=2;z=5;v=016;type=web&fm=44>

Teselas capa satélite:

<http://shangetu0.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu1.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu2.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu3.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu4.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu5.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu6.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu7.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu8.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>
<http://shangetu9.map.bdimg.com/it/u=x=1;y=0;z=3;v=009;type=sate&fm=46>

Teselas capa nombres chino (existen dos juegos de URLs funcionales):

<http://shangetu0.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu1.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu2.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu3.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu4.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu5.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu6.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu7.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu8.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>
<http://shangetu9.map.bdimg.com/it/u=x=1;y=0;z=3;v=015;type=trans&fm=47>

<http://online0.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online1.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online2.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online3.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online4.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online5.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online6.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online7.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online8.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>
<http://online9.map.bdimg.com/tile/?qt=tile&x=1&y=0&z=3&styles=sl&v=015>

Una vez obtenidas las URLs, el siguiente paso sería ajustar las coordenadas para que un punto del mapa correspondiese con sus coordenadas reales en el mundo, sin embargo, dado que en China se aplica encriptación geográfica a los mapas, no es posible cumplir esto. En lugar de tratar de ajustar a las coordenadas reales, se ha realizado el ajuste con respecto a las coordenadas que proporciona la API de Baidu. OpenLayers necesita que una capa especifique el valor de su máxima extensión y la máxima resolución para poder mostrarse correctamente sus coordenadas. Además es necesario que la capa especifique la proyección en la que está para poder convertir coordenadas de un sistema a otro. A continuación se van a explicar cada uno de esos valores y como se han calculado.

maxExtent (máxima extensión): Este parámetro representa los límites de la extensión de la capa por la izquierda, arriba, a la derecha y abajo. En el caso de Baidu este valor no se ha calculado si no que ha sido extraído directamente de su API mediante ingeniería inversa.

maxExtent: new OpenLayers.Bounds(-20446584.83, -12474104.17, 20969201.02, 12474104.17)

maxResolution (Resolución máxima): Este parámetro representa la equivalencia entre pixeles y metros en el mundo real. En el caso de Baidu su valor máximo, se dará para el nivel de zoom 0 aunque no está disponible una tesela para ese nivel ya que devuelve un tesela de color gris. La resolución para cualquier nivel de zoom corresponde con la siguiente fórmula:

$$resolucion = 2^{18-nivelZoom}$$

Por lo que la resolución máxima sería la correspondiente al nivel de zoom 0 como se ha mencionado antes tal y como se muestra en la siguiente fórmula:

$$maxResolution = 2^{18-0} = 262.144$$

Sin embargo, dado que en Baidu los primeros niveles de zoom no tienen teselas (el primer nivel de zoom con teselas es el 3), se ha realizado un ajuste haciendo que el nivel de zoom 3 de Baidu corresponda al nivel 0 dentro de OpenLayers. Por ello, el valor que toma maxResolution con el ajuste se muestra en la siguiente fórmula:

$$maxResolution = 2^{18-3} = 32.768$$

projection (Proyección en la que está la capa):El valor de este parámetro indica la proyección en la que está la capa. Este valor es necesario, sobre todo para trabajar con capas de otros proveedores en distintas proyecciones. Sin embargo, dado que Baidu no sigue los sistemas de coordenadas internacionales, si no que usa uno propio que incluye la encriptación geográfica, no es posible especificar una proyección directamente. Para resolver este problema, se realizó ingeniería inversa a la API de Baidu mediante diferentes herramientas para conseguir extraer el código usado por Baidu para la conversión desde sus coordenadas en metros a sus coordenadas en grados (ninguna de las dos coincide con coordenadas como las correspondientes a EPSG:4326 ni EPSG:900913). Dado que este código extraído no corresponde a ninguna proyección estándar, se creó un nombre específica para la proyección "BAI_COORD" y se añadió el código extraído a la librería de conversión de coordenadas de OpenLayers. De cara a la compatibilidad con otras capas usadas, se ha establecido que las coordenadas de Baidu en grados se equiparan a ESPG:4326 y las que corresponden a metros se

equiparan a EPSG:900913 y EPSG:3395, debido a la imposibilidad de hacer una conversión de unos sistemas a otros. A continuación, se incluyen las funciones extraídas de la API de Baidu por ingeniería inversa para pasar de grados a metros y de metros a grados en sus sistemas de coordenadas.

Código de Baidu para convertir de metros a grados:

```
function calculateBaiduA4326(xy) {
  var Eo = [1.289059486E7, 8362377.87, 5591021, 3481989.83, 1678043.12,
0];

  var zs = [
    [1.410526172116255E-8, 8.98305509648872E-6, -1.9939833816331,
200.9824383106796, -187.2403703815547, 91.6087516669843, -
23.38765649603339, 2.57121317296198, -0.03801003308653, 1.73379812E7],
    [-7.435856389565537E-9, 8.983055097726239E-6, -0.78625201886289,
96.32687599759846, -1.85204757529826, -59.36935905485877,
47.40033549296737, -16.50741931063887,
2.28786674699375, 1.026014486E7
],
    [-3.030883460898826E-8, 8.98305509983578E-6,
0.30071316287616, 59.74293618442277,
7.357984074871, -25.38371002664745,
13.45380521110908, -3.29883767235584,
0.32710905363475, 6856817.37
],
    [-1.981981304930552E-8, 8.983055099779535E-6,
0.03278182852591, 40.31678527705744,
0.65659298677277, -4.44255534477492,
0.85341911805263, 0.12923347998204, -0.04625736007561,
4482777.06
],
    [3.09191371068437E-9, 8.983055096812155E-6,
6.995724062E-5, 23.10934304144901, -2.3663490511E-4, -
0.6321817810242, -0.00663494467273, 0.03430082397953, -
0.00466043876332, 2555164.4
],
    [2.890871144776878E-9, 8.983055095805407E-6, -3.068298E-8,
7.47137025468032, -3.53937994E-6, -0.02145144861037, -1.234426596E-5,
1.0322952773E-4, -3.23890364E-6, 826088.5]
];

  var arraySel = null;

  for (var i = 0; i < Eo.length; i++) {
    if (Math.abs(xy.y) >= Eo[i]) {
      arraySel = zs[i];
      break;
    }
  }
  if (arraySel != null) {
    var c = arraySel[0] + arraySel[1] * Math.abs(xy.x);
    var d = Math.abs(xy.y) / arraySel[9];
    d = arraySel[2] + arraySel[3] * d + arraySel[4] * d * d + arraySel[5] * d *
d * d + arraySel[6] * d * d * d * d + arraySel[7] * d * d * d * d * d + arraySel[8]
* d * d * d * d * d * d * d * d * d;
  }
}
```

```

    c = c * (0 > xy.x ? -1 : 1);
    d = d * (0 > xy.y ? -1 : 1);

    xy.x = c;
    xy.y = d;
  } else {
    xy.x = 0;
    xy.y = 0;
  }
  // console.log("B->4326 = (" + xy.x + ", " + xy.y + ")");
  return xy;
}

```

Código de Baidu para convertir de grados a metros:

```

function calculate4326ABaidu(xy) {
  console.log("4326 -> BAI_COORD");
  var lonlat;
  var Zi = [75, 60, 45, 30, 15, 0];

  var Co = [
    [-0.0015702102444, 111320.7020616939,
      1704480524535203, -10338987376042340,
      26112667856603880, -35149669176653700,
      26595700718403920, -10725012454188240,
      1800819912950474, 82.5
    ],
    [8.277824516172526E-4, 111320.7020463578,
      6.477955746671607E8, -4.082003173641316E9,
      1.077490566351142E10, -1.517187553151559E10,
      1.205306533862167E10, -5.124939663577472E9,
      9.133119359512032E8, 67.5
    ],
    [0.00337398766765, 111320.7020202162,
      4481351.045890365, -2.339375119931662E7,
      7.968221547186455E7, -1.159649932797253E8,
      9.723671115602145E7, -4.366194633752821E7,
      8477230.501135234, 52.5
    ],
    [0.00220636496208, 111320.7020209128,
      51751.86112841131, 3796837.749470245,
      992013.7397791013, -1221952.21711287,
      1340652.697009075, -620943.6990984312,
      144416.9293806241, 37.5
    ],
    [-3.441963504368392E-4, 111320.7020576856,
      278.2353980772752, 2485758.690035394,
      6070.750963243378, 54821.18345352118,
      9540.606633304236, -2710.55326746645,
      1405.483844121726, 22.5
    ],
    [-3.218135878613132E-4, 111320.7020701615,
      0.00369383431289, 823725.6402795718,
      0.46104986909093, 2351.343141331292,
      1.58060784298199, 8.77738589078284,
      0.37238884252424, 7.45
    ]
  ];
}

```

```

    ]
  ];

  var arraySel = null;

  var limLonRight = 180;
  var limLonLeft = -180;
  while (xy.x > limLonRight) {
    xy.x -= limLonRight - limLonLeft;
  }

  while (xy.x < limLonLeft) {
    xy.x += limLonRight - limLonLeft;
  }

  var limLatUp = 74;
  var limLatDown = -74;
  xy.y = Math.max(xy.y, limLatDown);
  xy.y = Math.min(xy.y, limLatUp);

  /** ***** */

  for (var i = 0; i < Zi.length; i++) {
    if (xy.y >= Zi[i]) {
      arraySel = Co[i];
      break;
    }
  }

  if (arraySel == null) {
    for (var i = Zi.length - 1; 0 <= i; i--) {
      if (xy.y <= -Zi[i]) {
        arraySel = Co[i];
        break;
      }
    }
  }

  if (arraySel != null) {
    var c = arraySel[0] + arraySel[1] * Math.abs(xy.x);
    var d = Math.abs(xy.y) / arraySel[9];
    d = arraySel[2] + arraySel[3] * d + arraySel[4] * d * d + arraySel[5] * d *
d * d + arraySel[6] * d * d * d * d + arraySel[7] * d * d * d * d * d + arraySel[8]
* d * d * d * d * d * d;

    c = c * (0 > xy.x ? -1 : 1);
    d = d * (0 > xy.y ? -1 : 1);

    xy.x = c;
    xy.y = d;
  } else {
    xy.x = 0;
    xy.y = 0;
  }
  console.log("x:" + xy.x + ", y:" + xy.y);
  return xy;
}

```

Una vez completada la integración de Baidu en OpenLayers, dado que no se puede comprobar la correcta integración con otras capas, se procedió a integrar directamente la API de Baidu en OpenLayers para validar la integración de la capa que usa las teselas. Para realizar esta integración se tomó como referencia la integración de la capa de Google que hace OpenLayers, ya que está integrada la API de Google. Para realizar esta integración se usa la clase EventPane de OpenLayers que permite interceptar los eventos que van a la API y posteriormente realizar llamadas a las funciones de la API que desencadenan la respuesta a la acción realizada, de esta manera se puede interactuar con la API pero usando como intermediario a OpenLayers. Para hacer esto mismo con Baidu, el primer paso consistió en realizar pruebas con su API directamente para poder averiguar los nombres de las funciones necesarias y así posteriormente proceder a su integración.

Una vez se completó el proceso de integración de la API se pudo comprobar que tanto la capa integrada usando directamente las teselas del servicio de Baidu como la capa que usa directamente su API coinciden, por lo que a pesar de que las coordenadas no coincidan con las del sistema internacional, es posible validar la integración correcta de ambas.

Anexo F. Restricciones al trabajo con información geográfica online en China (Paper JIIDE 2013)

A continuación se reproduce el paper "Restricciones al trabajo con información geográfica online en China" presentado en las JIIDE 2013 a raíz de este proyecto. Este paper se puede consultar también en el enlace

http://www.ideo.es/resources/presentaciones/JIIDE13/miercoles/15_Restricciones_China.pdf

Rabaza Bergua, Carlos S.; López-de-Larrínzar-Galdámez, Juan; Salvador Suárez, Iván; Usón Montesinos, Miguel; Muro Medrano, Pedro R.

En China existe una legislación bastante restrictiva en algunos aspectos en lo relativo a los mapas y a su creación. Esas normativas son las responsables de algunos problemas existentes a la hora de integrar diferentes proveedores de información geográfica de Internet en un único servicio. En este artículo se busca analizar esa problemática y mostrar los problemas encontrados entre la información de diferentes servicios.

PALABRAS-CLAVE

Mapas en China, encriptación geográfica en China, Baidu, Google China, Bing China, OpenStreetMap China.

1. Introducción

Aunque en los últimos años la República Popular de China ha experimentado una notable apertura al exterior, en otras áreas, sigue manteniendo importantes restricciones. Una de éstas, es la topografía y cartografía. Estas restricciones son limitaciones artificiales impuestas por el gobierno chino para salvaguardar su seguridad nacional.

No es, sin embargo, un caso único en el mundo. Otros gobiernos han mostrado cierta preocupación respecto a los servicios de mapas online. Sobre todo en lo referente a instalaciones consideradas sensibles que puedan aparecer en la cartografía o imágenes de satélite de servicios como Google Maps. De hecho en ocasiones, algunos gobiernos solicitan a este tipo de servicios la ocultación de algunas zonas que puedan contener alguna de estas instalaciones [1].

El gobierno chino va un paso más allá, en lugar de recurrir únicamente a la ocultación de ciertas zonas, obligan a que todos los proveedores de mapas apliquen una transformación a sus mapas (también llamada encriptación geográfica). De esta manera, las coordenadas usadas por proveedores de mapas chinos en Internet sólo utilizan sistemas de coordenadas autorizados por las autoridades chinas. En la actualidad, estas autoridades no autorizan el uso de sistemas de coordenadas como el EPSG:4326 o el EPSG:3857 que son los utilizados por los proveedores de mapas mas populares. Esto genera toda una problemática de por sí, sobre todo a la hora de relacionar información de unos proveedores con otros, ya que incluso hay algunos que aplican una segunda transformación como es el caso de Baidu uno de los mayores proveedores de mapas online en China y competencia directa de Google en ese territorio.

Este documento presenta un breve análisis de las limitaciones técnicas impuestas por el gobierno chino a los servicios de mapas por Internet al amparo de su legislación altamente restrictiva.

2. Restricciones legales en China

En esta sección se presenta un análisis de las leyes chinas [2] que puedan afectar directamente a la creación de información topográfica o cartográfica.

Toda actividad relacionada con la topografía o cartografía en China entra en el ámbito de un organismo especial llamado National Administration of Surveying, Mapping and Geoinformation también conocido como State Bureau of Survey and Mapping (SBSM) [3]. Además existen ciertos aspectos especificados en la legislación China que implican también la participación del Ejército Popular de Liberación (también conocido como People's Liberation Army o PLA [4]), ya que los mapas son considerados un asunto de seguridad nacional, dado que la divulgación secretos en los mapas de cualquier clase que pongan en peligro la soberanía o la seguridad del estado puede constituir un crimen y derivarse responsabilidades criminales dependiendo de la gravedad de la información. Si las circunstancias no son serias podría quedar en un multa entre 10.000 y 100.000 yuanes (entre 1.239,38309 y 12.393,8309 €), de lo contrario la multa podría ser de 100.000 a 500.000 yuanes (entre 12.393,8309 y 61.969,1546€) y además la expulsión del país.

Toda organización, o persona de origen extranjero, que desee llevar cabo análisis geológicos o actividades relacionadas con los mapas en cualquier parte del territorio chino bajo la jurisdicción de la República Popular de China, debe en primer lugar solicitar una autorización al SBSM además de al departamento correspondiente del PLA. Además una vez obtenida la autorización, se debe realizar el proyecto en conjunto con los departamentos relevantes o unidades de la República Popular de China trabajando en conjunto con igualdad de condiciones o bien se les puede subcontratar el trabajo. La legislación China establece la restricción de que la información topográfica o cartográfica no puede contener secretos de estado o comprometer la seguridad del mismo.

El gobierno chino define un sistema de referencia geodésico propio para todo su territorio que cuenta con la aprobación de seguridad del PLA. En caso de que sea realmente necesario usar un sistema de coordenadas no definido por las autoridades Chinas se debe autorizar su uso por parte del SBSM y por parte del departamento competente de topografía y cartografía del PLA.

Una vez se ha completado un proyecto de topografía y cartografía se debe remitir la información recogida al SBSM. Este organismo en conjunto con el departamento correspondiente del PLA procederá a examinar la información referente a posiciones, elevaciones, profundidades, áreas y longitudes. Una vez se le dé el visto bueno a la información esta pasa al SBSM para su aprobación final. Adoptar un sistema de coordenadas sin autorización cuando se realizan las actividades de topografía y cartografía así como publicar información geográfica y datos relacionados con cualquier área bajo la jurisdicción de la República Popular de China puede suponer una multa de 100.000 yuan (12.393'8309 €).

3. Restricciones de los mapas en China

Como ya se ha mencionado, el gobierno chino obliga a los proveedores de mapas a que apliquen una transformación a la información cartográfica de modo que no coincidan las coordenadas de sus mapas con los sistemas de coordenadas usados

por otros servicios. Esta transformación o encriptación geográfica que están obligados a aplicar los proveedores de mapas recibe el nombre de GCJ-02. Se da el caso también de que algunos proveedores de mapas aplican un segundo desplazamiento como el de Baidu que, según informa en su página web [5], aplica otro nivel más de desplazamiento llamado BD-09 para incrementar la privacidad de sus usuarios. Esta encriptación geográfica, no es reversible, es decir, existen por ejemplo formas para transformar coordenadas desde EPSG:4326 a BD-09 o a GCJ-02 pero no existe función inversa que permita recuperar las coordenadas originales. A este respecto se preguntó directamente vía correo electrónico al servicio de atención al cliente de Baidu y su respuesta fue que no es posible aplicar una transformación inversa.

Esta situación en China genera problemas a los usuarios de servicios online que desean usar las coordenadas de un punto en un servicio e intentan pasarlas a otro servicio o en general al intentar combinar información de diferentes servicios. Aunque este problema existe también en proveedores de mapas online internacionales por el uso de diferentes sistemas coordenadas, en el caso de China, la situación se agrava debido a que como ya se ha mencionado en el apartado "RESTRICCIONES LEGALES EN CHINA" el gobierno chino tiene su propio sistema de referencio geodésico que cuenta con la aprobación del PLA. Ese sistema de referencia propio, es el que se tiene que usar allí a menos que se justifique la necesidad de usar un sistema de coordenadas distinto a los habitualmente autorizados y se apruebe su uso, tal y como consta en la legislación china.

A continuación, para ilustrar la problemática de uso de información entre diferentes proveedores de mapas por Internet, se van a analizar varios de ellos y las diferentes coordenadas que se muestran en cada uno de ellos para una misma posición geográfica. Los servicios que se van a comparar se listan a continuación, así como el sistema de coordenadas usado en las coordenadas obtenidas en cada uno:

Proveedor:	Sistema de referencia:	Unidad de medida:
Google Maps	EPSG:4326	Grados
Bing Maps	EPSG:4326	Grados
OpenStreetMap	EPSG:4326	Grados
Baidu Maps	GCJ-02 y BD-09	Grados

Tabla 1: Resumen de proveedores analizados

En primer lugar se van a mostrar las diferentes coordenadas obtenidas para un mismo punto geográfico ubicado en China para los servicios de la Tabla 1. Para comparar la información mostrada se va a usar como punto geográfico de referencia la esquina inferior derecha de la ciudad prohibida ya que resulta fácil de localizar visualmente en un mapa (Ver Figura 1).



Figura 1: El punto rojo muestra la esquina usada como referencia geográfica.

Comenzamos el análisis con Google Maps (<https://maps.google.es/>), que habitualmente es considerado como un referente en información cartográfica en Internet. La figura 2 muestra las coordenadas que tiene el punto usado como referencia en la capa satélite de Google, sus coordenadas son (39.912331°, 116.395608°) y la figura 3 muestra esas coordenadas sobre la capa del mapa. Como se puede ver el punto no está en la misma zona en ambas imágenes.

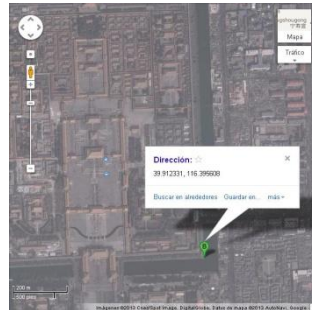


Figura 2: Punto de referencia marcado en la capa de satélite de Google.



Figura 3: Coordenadas del punto de referencia en la capa de satélite sobre la capa del mapa de Google.

Cómo se puede ver en las figuras 2 y 3 aunque las coordenadas son las mismas en ambas imágenes, el marcador no está situado en la misma posición visual en ambas capas. En la figura 4 se muestra la posición del punto de referencia en la capa mapa de Google, las coordenadas son: (39.913767°, 116.401876°). La diferencia entre las coordenadas de la capa satélite y la capa mapa es de: ($\Delta 0.001436^\circ$, $\Delta 0.006268^\circ$)

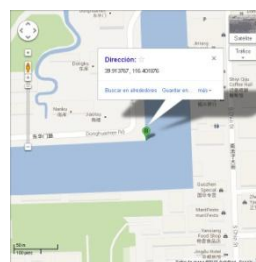


Figura 4: Coordenadas del punto de referencia en la capa de mapa de Google.

Sin embargo, Google ha resuelto esta diferencia para su servicio de mapas en China (<http://ditu.google.cn/>). Al contrario de lo que ocurre en los servicios de mapas de Google usados en el resto del mundo, la versión de mapas de Google en China hace coincidir el desplazamiento de la capa del mapa con la capa de satélite, haciendo que ambas capas trabajen en el mismo sistema de coordenadas. Este

comportamiento se puede ver en las figuras 5 y 6 (las coordenadas usadas en ambas son 39.913767° , 116.401876°) en las que el error entre la posición en la capa mapa y la satélite es mucho menor al error presente en el servicio normal de Google, aunque sigue existiendo una desviación entre ambas capas.

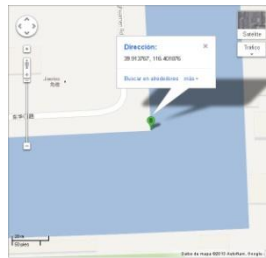


Figura 5: Coordenadas del punto de referencia en la capa de mapa de Google China.

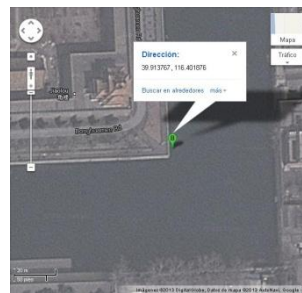


Figura 6: Coordenadas del punto de referencia en la capa de mapa de Google China sobre la capa de Satélite de Google China.

A continuación se va a ver el caso de Bing Maps. Bing Maps no dibuja en la capa mapa la ciudad prohibida, por lo que no se puede usar para comparar con el punto de referencia que estábamos usando. Por ello, se muestran a continuación las coordenadas del punto de referencia de la ciudad prohibida en la capa satélite de Bing (figura 7) para seguidamente realizar la comparación en Bing con otro punto de referencia geográfico.

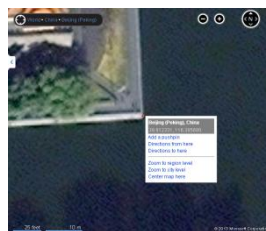


Figura 7: Coordenadas del punto de referencia (39.912331° , 116.395680°) en la capa de satélite de Bing.

El nuevo punto de referencia elegido para Bing se encuentra en la avenida justo delante de la ciudad prohibida. La figura 8 muestra el nuevo punto de referencia abajo a la izquierda en color verde y el punto de referencia del resto de proveedores en rojo en la parte superior derecha.

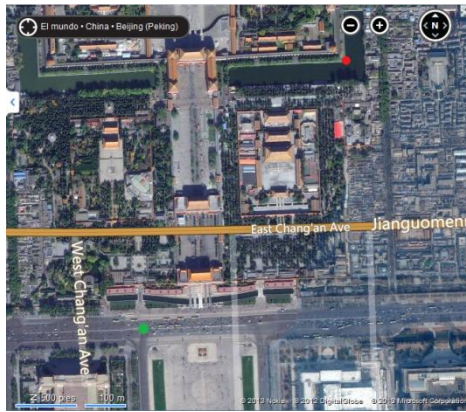


Figura 8: Nuevo punto de referencia para Bing Maps.

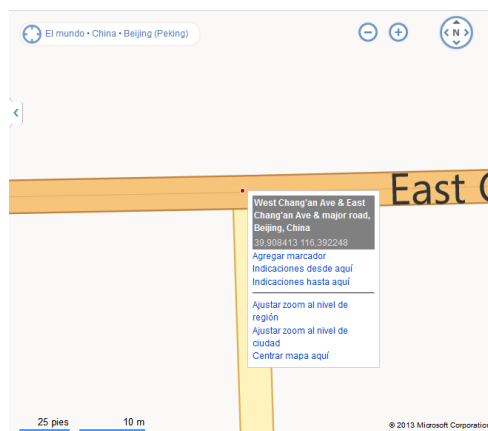


Figura 9: Coordenadas del punto de referencia de Bing (39.908413° , 116.392248°) en la capa de calles de Bing.

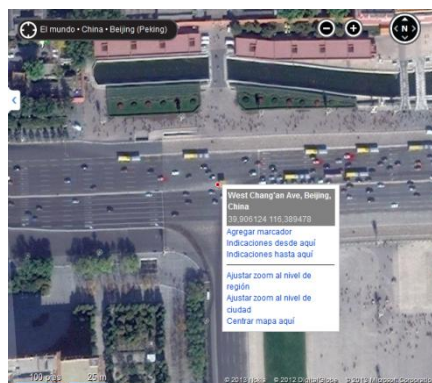


Figura 10: Coordenadas del punto de referencia de Bing (39.906124° , 116.389478°) en la capa de satélite de Bing.

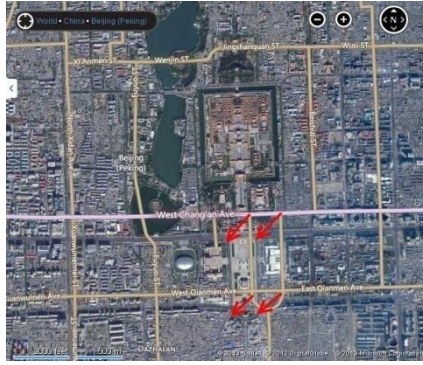


Figura 11: Desplazamiento entre capa de calles y satélite de Bing marcado con las flechas rojas.

El siguiente proveedor a valorar es OpenStreetMap. OSM es diferente de otros proveedores dado que son los usuarios los que construyen el mapa de forma colaborativa. Esta forma de proceder les da cierto grado de independencia, en este caso, del gobierno chino, por lo que sus mapas no contienen el desplazamiento obligado por la legislación de China. La figura 12 muestra la posición del punto de referencia en los mapas de OpenStreetMap. Las coordenadas de ese punto en OpenStreetMap son (39.912331°, 116.395608°).

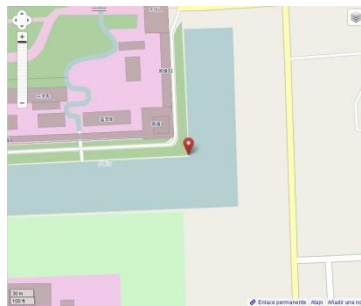


Figura 12: El marcador muestra la posición de 39.912331°, 116.395608° en OSM (<http://www.openstreetmap.org/?mlat=39.912331&mlon=116.395608&zoom=18>).

Por último se va a analizar la posición del punto de referencia en Baidu. Este caso tiene el añadido de que además de la encriptación geográfica GCJ-02 se aplica un segundo nivel propio de Baidu llamado BD-09 por lo que los desplazamientos pueden variar más. En este caso sólo se van a mostrar las coordenadas en la capa de satélite de Baidu (Figura 10) ya que coincide perfectamente con la capa de calles de Baidu.

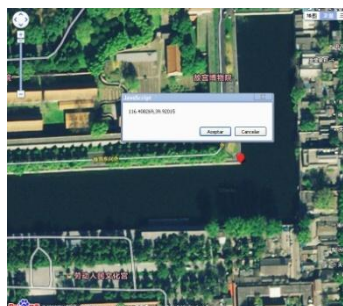


Figura 10: El marcador 1 muestra la posición del punto en Baidu 39.920150°, 116.408269°.

A modo de resumen, se incluyen a continuación tres tablas que recogen la información mostrada hasta este punto. La Tabla 2 contiene un resumen de las coordenadas correspondientes a la posición usada como referencia para cada uno de los proveedores y capas analizadas. Con la información de la Tabla 2 se han construido las Tablas 3 y 4 que muestran las desviaciones existentes entre cada uno de los casos analizados medidas en grados y expresadas en valores absolutos. La Tabla 3 muestra las desviaciones para los valores de latitud, y la Tabla 4 para longitud. Todos los datos recogidos a continuación son datos obtenidos en los interfaces web de cada proveedor, de forma manual, de modo que se puede dar un pequeño margen de error por error humano al hacer los clics en cada uno de ellos (margen de pocos pixels).

Proveedor	Capa	Abreviación	Latitud	Longitud	Sistema ref:
Google	Calles	GCal	39.913767°	116.401876°	EPSG:4326
Google	Satélite	GSat	39.912331°	116.395608°	EPSG:4326
Google China	Calles	GCnC	39.913767°	116.401876°	EPSG:4326
Google China	Satélite	GCnS	39.913738°	116.401855°	EPSG:4326
Bing	Satélite	BingS	39.912331°	116.395680°	EPSG:4326
Bing	Calles punto ref2	BingC2	39.908413°	116.392248°	EPSG:4326
Bing	Satélite punto ref2	BingS2	39.906124°	116.389478°	EPSG:4326
OSM	Calles	OSM	39.912331°	116.395608°	EPSG:4326
Baidu	Calles y Satélite	BaiH	39.92015°	116.408269°	GCJ-02 y BD-09

Tabla 2: Resumen de coordenadas del punto de referencia para los diferentes proveedores analizados.

	GCal	GSat	GCnC	GCnS	BingS	BingC2	BingS2	OSM	BaiH
GCal	0	0,001436	0	0,000029	0,001436	-	-	0,001436	0,006383
GSat		0	0,001436	0,001407	0	-	-	0	0,007819
GCnC			0	0,000029	0,001436	-	-	0,001436	0,006383
GCnS				0	0,001407	-	-	0,001407	0,006412
BingS					0	-	-	0	0,007819
BingC						0	0,00228	-	-

2							9		
BingS2							0	-	-
OSM								0	0,007819
BaiH									0

Tabla 3: Diferencias de latitud entre los diferentes servicios (valores absolutos y expresados en grados), las celdas marcadas con "-" no se comparan por haberse usado otro punto de referencia para las coordenadas.

	GCal	GSat	GCnC	GCnS	BingS	BingC2	BingS2	OSM	BaiH
GCal	0	0,006268	0	0,000021	0,006196	-	-	0,006268	0,006393
GSat		0	0,006268	0,006247	0,000072	-	-	0	0,012661
GCnC			0	0,000021	0,006196	-	-	0,006268	0,006393
GCnS				0	0,006175	-	-	0,006247	0,006414
BingS					0	-	-	0,000072	0,012589
BingC2						0	0,002770	-	-
BingS2							0	-	-
OSM								0	0,012661
BaiH									0

Tabla 4: Diferencias de longitud entre los diferentes servicios (valores absolutos y expresados en grados), las celdas marcadas con "-" no se comparan por haberse usado otro punto de referencia para las coordenadas.

Para concluir, se van a analizar los desplazamientos que ocurren en los diferentes proveedores para 4 ubicaciones diferentes de China: Beijing, Lhasa, Xi'an y Guangzhou ubicadas donde se puede ver en la Figura 11.



Figura 11: Muestra las ubicaciones de las ciudades analizadas.

La tabla 5 muestra la información recogida para la 4 ubicaciones. Para cada una de ellas se han tomado las coordenadas de una misma posición geográfica tomada como referencia en los diferentes servicios (columna proveedor) y sus respectivas capas (columna capa). Las columnas "Latitud", "Longitud" muestran las coordenadas obtenidas para esa ubicación utilizando la API del proveedor indicado. La columna "Sistema ref" Indicia el sistema de referencia en el que están las coordenadas de latitud y longitud. Para realizar la valoración de los posibles desplazamientos, se comparan las coordenadas de cada capa de cada proveedor con la capa satélite de Google. Esta comparación se realiza obteniendo el valor absoluto de la diferencia entre las coordenadas de ambas capas y se muestra en las columnas "OffsetLat" (diferencia de latitud) y "OffsetLong" (diferencia de longitud).

	Proveedor	Capa	Latitud	Longitud	Sistema ref	OffsetLat	OffsetLong
B e i g i n g	Google	Calles	39,90759	116,395728	EPSG:4326	0,001506	0,006281
	Google	Satélite	39,906084	116,389447	EPSG:4326	0,000000	0,000000
	Google China	Calles	39,907586	116,395726	EPSG:4326	0,001502	0,006279
	Google China	Satélite	39,907586	116,395726	EPSG:4326	0,001502	0,006279
	Bing	Calles	39,908412	116,392246	EPSG:4326	0,002328	0,002799
	Bing	Satélite	39,906102	116,389536	EPSG:4326	0,000018	0,000089
	OSM	Calles	39,906084	116,389447	EPSG:4326	0,000000	0,000000
	Baidu	Calles y Satélite	39,913809	116,402091	GCJ-02 y BD-09	0,007725	0,012644
L h a s a	Google	Calles	29,652491	91,121445	EPSG:4326	0,002657	0,001588
	Google	Satélite	29,655148	91,119857	EPSG:4326	0,000000	0,000000
	Google China	Calles	29,652447	91,121445	EPSG:4326	0,002701	0,001588
	Google China	Satélite	29,652447	91,121445	EPSG:4326	0,002701	0,001588
	Bing	Calles	29,65249	91,121445	EPSG:4326	0,002658	0,001588
	Bing	Satélite	29,655158	91,119862	EPSG:4326	0,000010	0,000005
	OSM	Calles	29,655148	91,119857	EPSG:4326	0,000000	0,000000
	Baidu	Calles y Satélite	29,65832	91,128054	GCJ-02 y BD-09	0,003172	0,008197
	Google	Calles	34,259427	108,947033	EPSG:4326	0,001529	0,004651

X i , a n	Google	Satélite	34,260956	108,942382	EPSG:4326	0,000000	0,000000
	Google China	Calles	34,259418	108,94703	EPSG:4326	0,001538	0,004648
	Google China	Satélite	34,259418	108,94703	EPSG:4326	0,001538	0,004648
	Bing	Calles	34,259459	108,94703	EPSG:4326	0,001497	0,004648
	Bing	Satélite	34,260945	108,942434	EPSG:4326	0,000011	0,000052
	OSM	Calles	34,260956	108,942382	EPSG:4326	0,000000	0,000000
	Baidu	Calles y Satélite	34,265687	108,953517	GCJ-02 y BD- 09	0,004731	0,011135
	G u a n g z h o u	Google	Calles	23,098866	113,252909	EPSG:4326	0,002692
Google		Satélite	23,101558	113,247561	EPSG:4326	0,000000	0,000000
Google China		Calles	23,098851	113,252914	EPSG:4326	0,002707	0,005353
Google China		Satélite	23,098851	113,252914	EPSG:4326	0,002707	0,005353
Bing		Calles	23,09885	113,252911	EPSG:4326	0,002708	0,005350
Bing		Satélite	23,101609	113,24758	EPSG:4326	0,000051	0,000019
OSM		Calles	23,101558	113,247561	EPSG:4326	0,000000	0,000000
Baidu		Calles y Satélite	23,104904	113,25941	GCJ-02 y BD- 09	0,003346	0,011849

Tabla 5: Coordenadas y desplazamientos respecto a la capa de satélite de Google para las 4 ubicaciones analizadas.

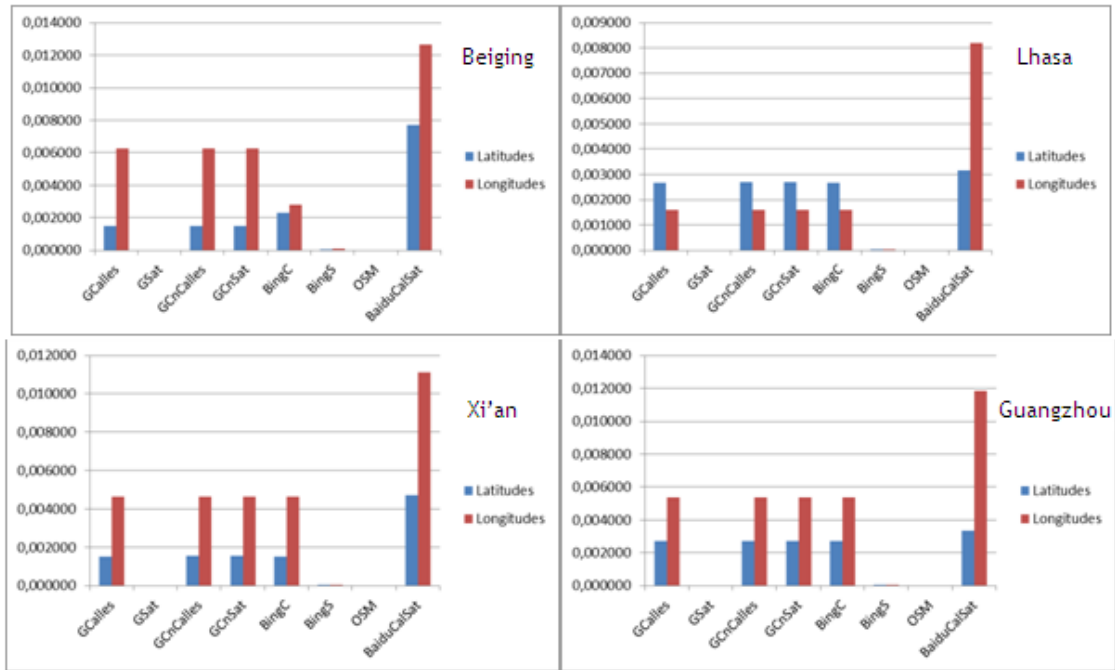


Figura 12: Gráficas de los desplazamientos en las cuatro ubicaciones respecto a la capa de satélite de Google.

4. Conclusiones

Después del análisis realizado de diferentes proveedores de mapas en China, se puede afirmar que en aquellos que tienen presencia internacional las capas que contienen información de las calles se encuentran desplazadas en mayor o menor medida en relación a sus capas de satélite (dado que las imágenes de satélite no dependen del gobierno chino y la información de calles sí). Se ha apreciado que los servicios proveedores de mapas con cobertura internacional y que dan servicio en China suelen adoptar la estrategia de ofrecer un servicio paralelo destinado al mercado chino, como puede ser el caso de Google o Microsoft, en el que corrigen este desplazamiento para que ambas capas coincidan (calles y satélite). El citado desplazamiento se debe a la encriptación geográfica que los proveedores de mapas chinos, como es el caso de Baidu, están obligados a aplicar. Además, como se puede apreciar en la Figura 15 el desplazamiento que se aplica con la encriptación geográfica GCJ-02 no es constante para todo el territorio chino, ya que varía para cada una de las regiones analizadas.

Del estudio se concluye que, según la legislación china y salvo permiso expreso, en su territorio los proveedores de mapas deben utilizar el sistema de referencia autorizado. Lo cuál obliga a adoptar la encriptación geográfica citada anteriormente. Esta encriptación no dispone de función inversa, de modo que la información geográfica producida allí no puede compararse con información geográfica en sistemas de referencia utilizados internacionalmente. Sólo aquellos mapas que se sometan a las anteriores restricciones podrán compararse entre sí, con la limitación que ello supone (por ejemplo a la hora de superponer distintas capas en un visualizador de mapas).

Por último, cabe destacar la excepción de OpenStreetMap que, al tratarse de un proyecto colaborativo en el que cualquier ciudadano puede participar, no aplica a sus mapas la transformación a la que obliga el gobierno chino, como se puede ver en la Figura 15. Iniciativas como las de OpenStreetMap garantizan la generación y

la utilización de la información geográfica con carácter abierto.

5. Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno de España a través del proyecto TIN2012-37826-C02-01. El trabajo de Juan López ha sido cofinanciado por el Gobierno de España a través de la Ayuda INNCORPORA INC-TU-2011-1528.

6. Referencias

- [1] Zonas con imágenes de satélite alteradas recogidas en la Wikipedia, enlace: http://en.wikipedia.org/wiki/Satellite_map_images_with_missing_or_unclear_data
- [2] Versión en Inglés de la legislación China referente a los mapas: <http://en.sbsm.gov.cn/article/LawsandRules/Laws/200710/20071000003241.shtml>
- [3] Página Web del National Administration of Surveying, Mapping and Geoinformation, también conocido como State Bureau of Survey and Mapping (SBSM): <http://en.sbsm.gov.cn/>
- [4] Página Web del People's Liberation Army: <http://eng.mod.gov.cn/>
- [5] Información referente a coordenadas y offset usado por el proveedor de mapas chino Baidu que es competencia directa de Google en China: <http://developer.baidu.com/map/question.htm#qa004>

AUTORES

Carlos S. Rabaza Bergua
carlosrb@geoslab.com

GeospatiumLab S.L.

Juan López-de-Larrínzar Galdámez
juanlg@geoslab.com

GeospatiumLab S.L.

Iván Salvador Suárez
ivans@unizar.es

Universidad de Zaragoza

Departamento de
Informática e Ingeniería de
sistemas

Miguel Usón Montesino
muson@geoslab.com

GeospatiumLab S.L.

Pedro R. Muro Medrano
prmuro@unizar.es

Universidad de Zaragoza

Departamento de
Informática e Ingeniería de
sistemas

Anexo G. Baidu Maps API interfaz conversión coordenadas

Este anexo contiene el documento facilitado por Baidu sobre su API de conversión de coordenadas al contactar por ellos vía correo electrónico. El documento estaba originalmente en chino, por lo que se utilizó Google Translate para traducirlo al inglés que es el idioma en el que se ha incluido aquí.

Coordinate Transformation Services

Interface Description of the coordinate transformation

Other coordinate system into the coordinate system of Baidu. Interface sample as follows:

<http://api.map.baidu.com/ag/coord/convert?from=2&to=4&x=116.254615&y=29.814476>

Parameter Description:

x/y : Latitude and longitude coordinates

from/to : Decided to transition effects, specific parameters days filled as follows:

Coordinate system	Parameter
WGS-84Coordinate system (GPS)	from=0,to=4
GCJ-02Coordinate system (Google, soso, Sogou)	from=2,to=4
51Map coordinate system	(First 51 map coordinates in addition to 10,000)from=2,to=4
Mapbar Coordinate system	You need to call back the conversion algorithm. Get WGS84 Coordinate. Re-use : from=0,to=4

Back to Results :

json Array, errorShow that the error number, x、 y Is the result of base64Encrypted string, Need Reflexive Solutions

Attach :mapbarCoordinate transformation , Input mapbar coordinate, Output gps coordinate (js Code Description)

```
function mapBar2WGS84(x,y){
x = parseFloat(x)*100000%36000000;
y = parseFloat(y)*100000%36000000;

x1 = parseInt(-
(((Math.cos(y/100000))*(x/18000))+((Math.sin(x/100000))*(y/9000))))+x);
```

```

y1 = parseInt(-
(((Math.sin(y/100000))*(x/18000))+((Math.cos(x/100000))*(y/9000)))+y);

x2 = parseInt(-
(((Math.cos(y1/100000))*(x1/18000))+((Math.sin(x1/100000))*(y1/9000)))+x+((
x>0)?1:-1));
y2 = parseInt(-
(((Math.sin(y1/100000))*(x1/18000))+((Math.cos(x1/100000))*(y1/9000)))+y+((
y>0)?1:-1));

return [x2/100000.0,y2/100000.0];
}

```

Limit

Less than 50 times per second visit

Example

Use JSONPWay, cross-domain access. Sample Link :

<http://dev.baidu.com/wiki/static/map/API/examples/?v=1.2&06#0&6§>

The batch coordinate transformation Interface (Advanced)

Interface parameters:

- 1.A single set of coordinate transformation with the previous request URL format unchanged
- 2.Batch coordinate conversion request URL more a parameter 'mode', mode = 1 request the bulk coordinate conversion interface to request a single set of coordinate conversion interface mode to another value.

Parameters each separated by a comma in the x-coordinate, y coordinates in each group separated by commas. Such as:

<http://api.map.baidu.com/ag/coord/convert?from=2&to=4&x=40,50&y=116,117&mode=1>

- 3.The batch coordinate transformation coordinate set the maximum number of 20 over 20 only return 20 results

Back to the results:

1.Data is returned in json format

2.A single set of coordinate transformation back to the same format before

3.Batch coordinate conversion return to the format:

```
[{"error":0,"x":"NDAuMDA2NjY4MzUzNTg3","y":"MTE2LjAwNTkyMzYyNTU1"}, {"error":0,"x":"NTAuMDA2Njc1NDk5OTY3","y":"MTE3LjAwNTkyNDk5OTg1"}]
```

Batch conversion limit

Per visit is less than 50 times, once 20.

Bulk sample

Take advantage of the the JSONP way, cross-domain access. Example link:

http://dev.baidu.com/wiki/static/map/API/examples/?v=1.2&0_7#0&7§

Anexo H. Análisis de librerías de mapas offline

En este anexo se realiza un análisis de diferentes librerías que permitan el uso de mapas sin conexión. La información de este documento ha sido recopilada de las páginas oficiales de cada librería. Uno de los requisitos para la librería es que ha de ser de código abierto, por lo que sólo se han considerado librerías cuyo código fuente esté disponible. También se han considerado para el análisis algunas aplicaciones siempre y cuando su código fuente esté disponible. Dado que la librería escogida se integraría con la librería de GeosLab la licencia es otro aspecto a tener en cuenta y a evaluar.

Big Planet Tracks	
Descripción	Big Planet Tracks es una aplicación de mapas online/offline que puede importar, exportar y grabar las rutas. También muestra datos como el tiempo, velocidad, distancia y elevación.
URL	https://code.google.com/p/big-planet-tracks/
Licencia del código	GNU GPL v3
Cartografía	Puede usar mapas online y offline (Base de datos SQLite creada usando Mobile Atlas Creator 1.7 o 1.8)
Uso offline	Sí. Para usar mapas offline es necesario crear un a base de datos SQLite usando Mobile Atlas Creator 1.7 o 1.8. Permite cambiar las bases de datos SQLite de los mapas de forma dinámica.
Fecha análisis	27/06/2013

AndNav2	
Descripción	AndNav2 es una aplicación que permite la navegación giro a giro. Soporta búsqueda de ubicación, POIs, texto a voz, fuentes de tráfico, búsqueda de contactos, predicción meteorológica para 4 días, Via-Points, zonas a evitar... La última actualización del proyecto parece ser del año 2009, por lo que parece estar abandonado.
URL	https://code.google.com/p/andnav/
Licencia del código	GNU GPL v3
Cartografía	Usa la información de OpenStreetMap.

Uso offline	Sí. Pero no ofrece navegación offline a menos que se prepare previamente la ruta cuando se disponga de conexión a Internet. Al crear la ruta se puede seleccionar que esté disponible de forma desconectada, haciendo que se descarguen las teselas correspondientes para su uso en la ruta cuando se esté en modo offline.
Fecha análisis	27/06/2013

Osmdroid

Descripción	Osmdroid proporciona herramientas y vistas para interactuar con los datos de OpenStreetMap.
URL	https://code.google.com/p/osmdroid/
Licencia del código	Apache License 2.0
Cartografía	Usa la información de OpenStreetMap.
Uso offline	Sí. Permite el uso de mapas offline habiendo descargado previamente las teselas de algún servicio de mapas como OpenStreetMap usando algún programa como el Mobile Atlas Creator. Los archivos creados con este programa se han de copiar al teléfono para ser usados por la aplicación. No permite el uso de información vectorial descargada que pueda ser renderizada en tiempo real.
Fecha análisis	27/06/2013

Mapforge

Descripción	<p>El proyecto de Mapforge proporciona software gratuito y abierto para aplicaciones basadas en OpenStreetMap.</p> <p>En la página de OpenStreetMap se detallan los pros y contras del uso de esta librería, se reproducen a continuación :</p> <ul style="list-style-type: none"> • + Fast on-device rendering of OpenStreetMap data • + Powerful and flexible overlay API • + Compact file format for map data storage • + Support for multi-touch gestures (e.g. pinch-to-zoom) • + Very high details (support for ~ 200 OSM tags) • + Runs on all devices with Android 1.5 and
--------------------	--

	<p>higher</p> <ul style="list-style-type: none"> + Small library (~ 400 KB, including sources and images) + Easy to use API (similar to Google Maps for Mobile API) + 100% free and open source (LGPL license) <p>- Sub-optimal rendering of water areas</p>
URL	<p>https://code.google.com/p/mapsforge/ http://wiki.openstreetmap.org/wiki/Mapsforge</p>
Licencia del código	GNU Lesser GPL
Cartografía	Usa la información de OpenStreetMap. Permite usar información vectorial para renderizarla en el dispositivo móvil en la visualización.
Uso offline	<p>Sí. Mapsforge dispone de mapas descargables para todo el mundo en la URL http://download.mapsforge.org/maps/ . En el caso del mapa para España su última actualización es del 9 de mayo del 2013.</p> <p>Si se quiere, es posible crear mapas para Mapsforge usando un plugin especial para la aplicación Osmosis que permita generar un archivo con información vectorial compatible.</p> <p>Estos mapas, son mapas vectoriales, y ocupan relativamente poco espacio, el fichero con la información para el mapa de España tiene un tamaño de 205 MB.</p>
Fecha análisis	27/06/2013

OsmAnd (OSM Automated Navigation Directions)

Descripción	Este proyecto está destinado a ofrecer visionado de mapas y navegación en dispositivos móviles.
URL	https://code.google.com/p/osmand/
Licencia del código	GNU GPL v3
Cartografía	Usa la información de OpenStreetMap.
Uso offline	Sí. OsmAnd es una aplicación de código abierto (GPLv3) que permite el uso de mapas de OpenStreetMap tanto de forma online como de forma offline descargando los mapas desde la propia aplicación. Los mapas offline pueden ser descargados desde la propia aplicación por países. Los mapas de esta aplicación son mapas vectoriales, ya que son de tamaño reducido y se puede apreciar cómo se

	renderizan en la visualización.
Fecha análisis	27/06/2013

Conclusiones

Después del análisis realizado, se ha escogido la librería de Mapsforge por ser la que se adapta mejor a las necesidades y al tipo de licencia. Una de las razones por las que también se ha escogido Mapsforge es porque disponen de un servidor con mapas de OpenStreetMap para las diferentes regiones del mundo, por lo que simplifica mucho el uso de diferentes mapas. Además tampoco se está limitado al uso de sus mapas, si no que existen las herramientas necesarias para crear mapas para cualquier región de la que OpenStreetMap disponga de información.

Anexo I. Análisis de Mapsforge

En este anexo se realiza un análisis de la librería escogida para la integración en la librería de GeosLab para Android.

1. ¿Qué es Mapsforge?

Mapsforge es una librería para Android de código abierto (bajo licencia GNU Lesser GPL). Se trata de una librería que permite el uso de mapas vectoriales de OpenStreetMap. Los mapas vectoriales tienen la ventaja de que ocupan muy poco espacio en el dispositivo, ya que la información se renderiza en el propio dispositivo. La desventaja que tienen este tipo de mapas, es que son más costosos de visualizar al tener que dibujar la tesela para mostrarla en pantalla, en Mapsforge, este problema se intenta aliviar usando una cache en la que se guardan las teselas ya pintadas mientras sea posible (existen restricciones de tamaño máximo ocupado). El proyecto de Mapsforge se inició en el año 2008 por el "Computer science institute of Freie Universität Berlin".

Mapsforge soporta el renderizado de mapas offline y online de OSM, overlays, planificación de rutas. Soporta también gestos multi-touch como el uso de "pinch" para hacer zoom. La API sigue una estructura similar a la API de Google Maps para móviles y su tamaño es bastante reducido unos 300 KB.

2. ¿Cómo funciona Mapsforge?

Mapsforge tiene dos modos de funcionamiento, el modo online y el offline. El modo online muestra las teselas descargándolas del servicio de mapas de Mapnik conforme son necesarias. Este modo evidentemente requiere de conexión a internet para poder llevar a cabo la descarga. El modo offline en cambio, no requiere de conexión a internet, ya que en lugar de descargar teselas, lo que hace es acceder a un archivo que tiene que estar en el almacenamiento del teléfono. Ese archivo contiene la información de la región que se quiere visualizar en formato vectorial. Al ser información vectorial lo que se almacena en el archivo ocupa mucho menos espacio que si se estuvieran almacenando las teselas directamente, pero tiene la desventaja de que visualizar una tesela en el mapa a partir de la información vectorial del archivo es más costoso en tiempo ya que tiene que accederse a la información y renderizarla antes de poder mostrarla. La alternativa offline, es una buena opción para dispositivos que por la razón que sea no disponen o no desean estar conectados a internet, permitiendo almacenar en el dispositivo el archivo para la región que se va a usar (para el mapa de España de fecha 25 de julio de 2013 ocupa 208 MB).

De cara a agilizar el dibujado de las teselas en modo offline, Mapsforge utiliza una cache tanto en memoria como en el almacenamiento del dispositivo móvil para guardar las que han sido generadas y así ahorrar tiempo en los dibujados. Ahora bien, esta caché no es infinita y por tanto tiene un tamaño limitado en el que ir almacenándolas.

En la siguiente página se muestra la estructura general de Mapsforge de las clases más básicas. La clase ubicada en "test.example.offlineosm.MainActivity"

sería un ejemplo de uso tanto para el caso online como el offline. Es una clase que tiene que extender de la clase "org.mapsforge.android.maps.MapActivity" y tiene que implementar el método onCreate. En ese método es donde va el código necesario para inicializar la librería bien en el modo online u offline según se requiera el desarrollador.

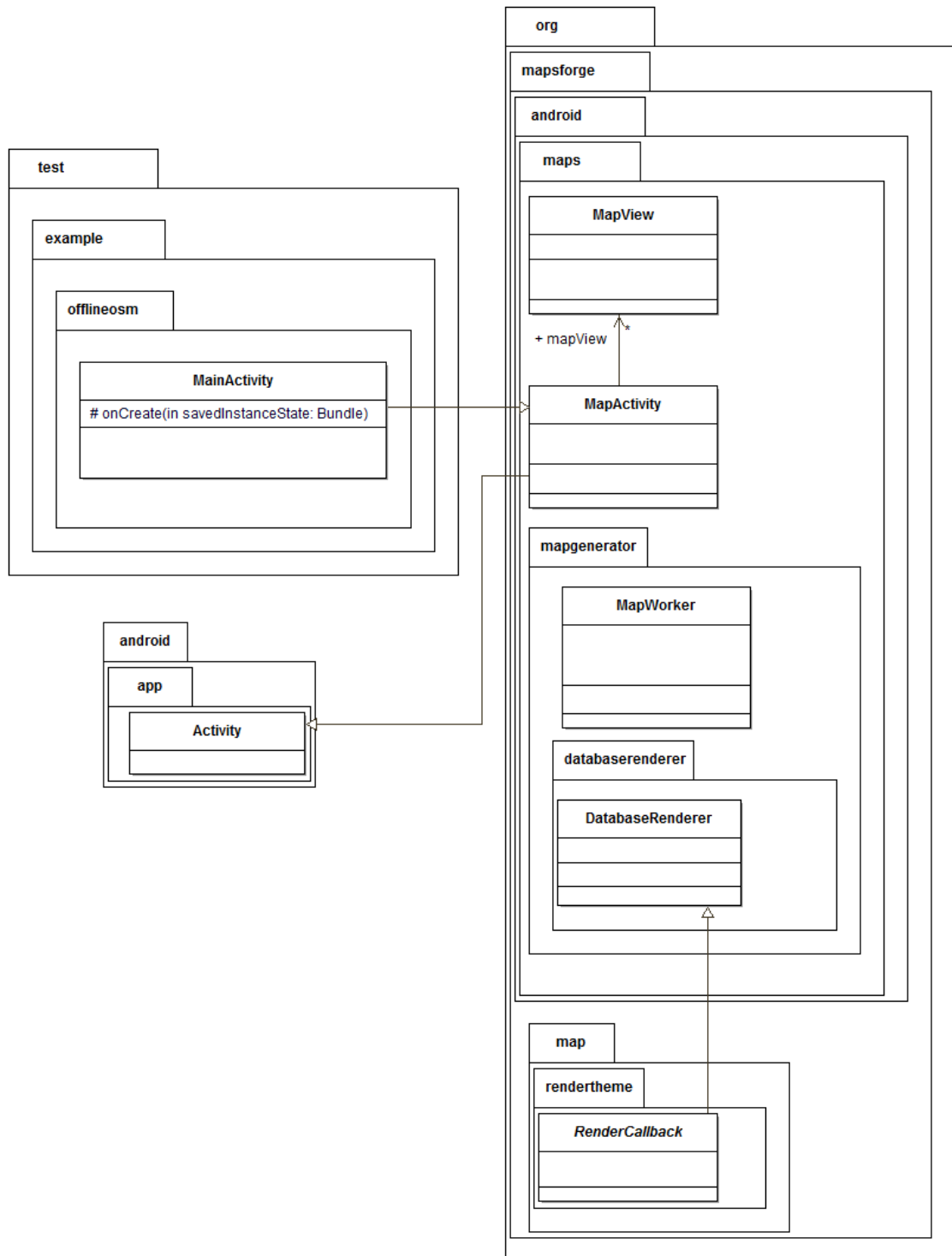


Figura I - 1 Diagrama de clases con las relaciones entre las clases básicas de Mapsforge.

A continuación se muestran dos diagramas para el uso de la librería en modo online y offline.

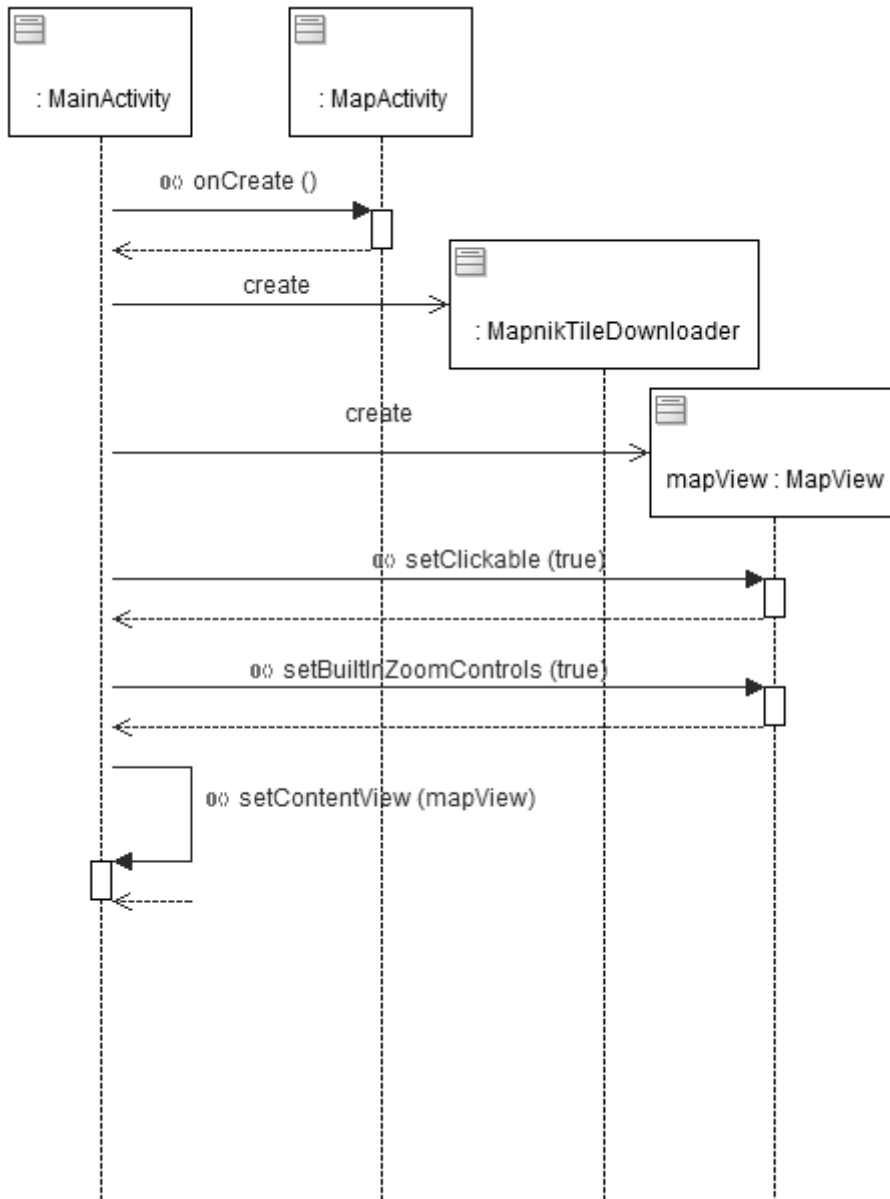


Figura I - 2 Diagrama de secuencia del uso de Mapsforge en modo online.

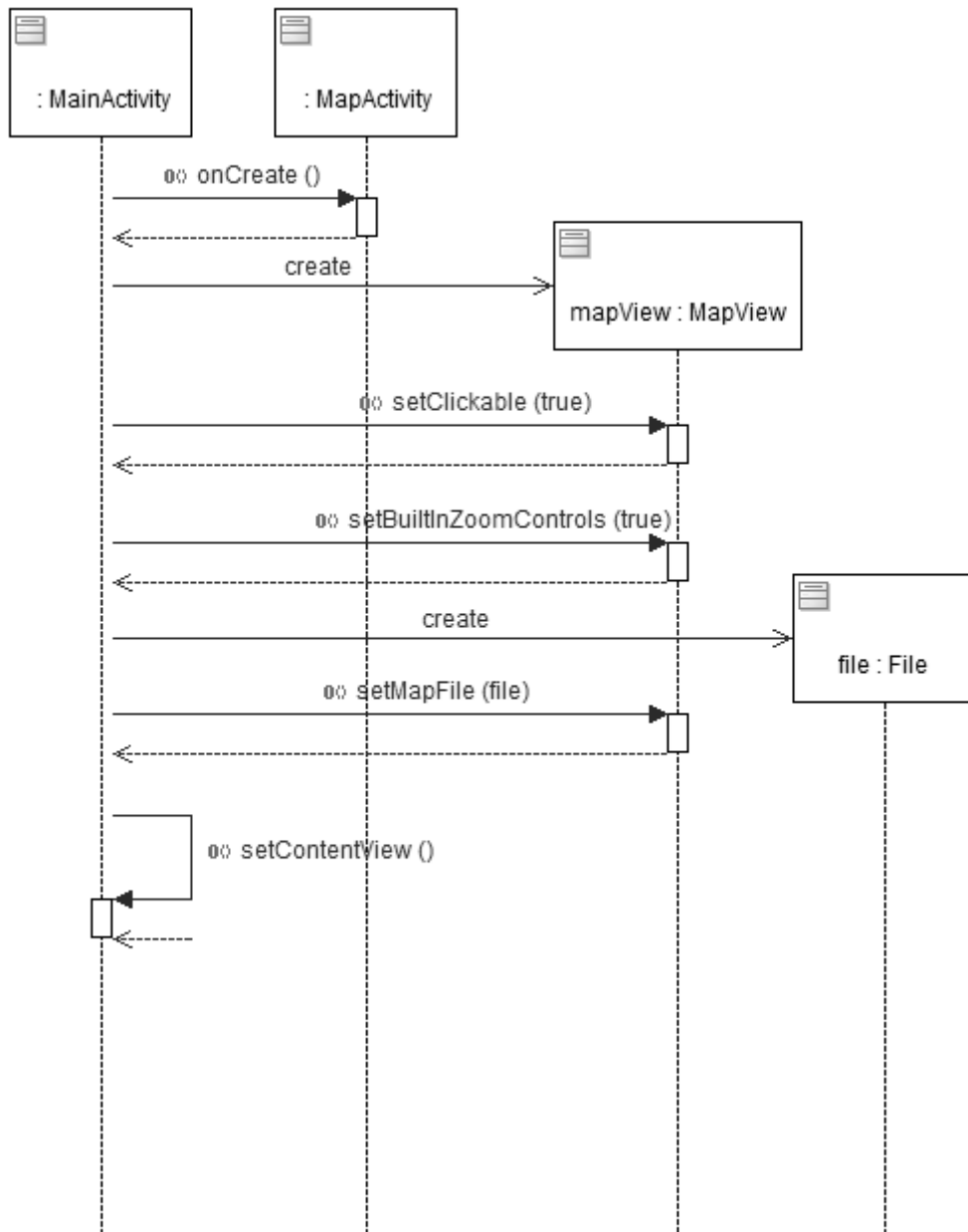


Figura I - 3 Diagrama de secuencia del uso de Mapsforge en modo offline.

Durante la creación de MapView se inicializan todos los elementos necesarios para el funcionamiento de la librería. En el siguiente diagrama de secuencia, se muestra la creación de los elementos que se consideran más importantes durante la creación de MapView.

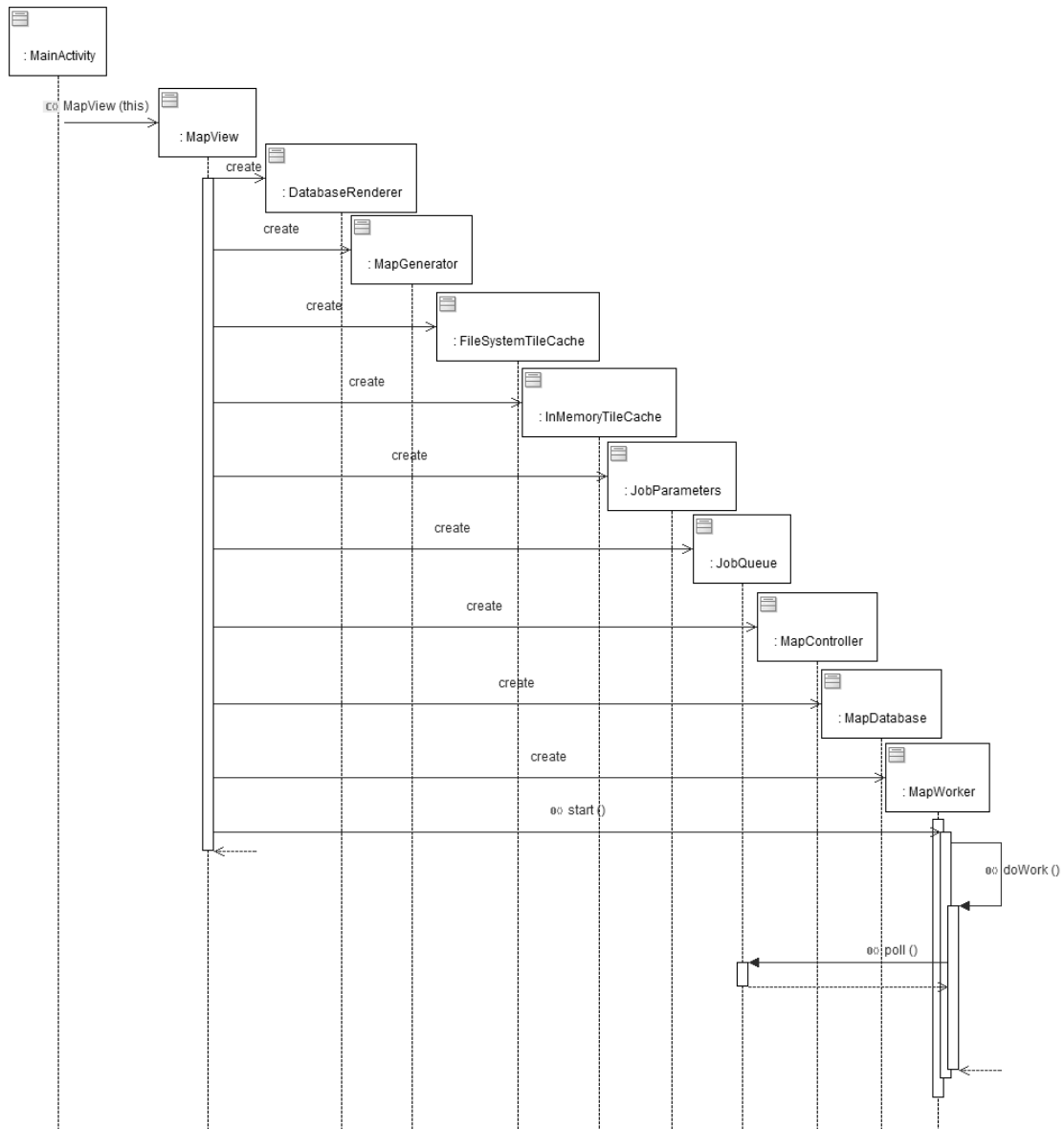


Figura I - 4 Diagrama de secuencia de la inicialización de los elementos de Mapsforge necesarios por parte de su clase MapView.

Finalmente, el dibujado del mapa en la pantalla del dispositivo se muestra en el siguiente diagrama. En la parte inferior del diagrama se puede ver que al poner el contenido de la vista, se dispara el evento de cambio de tamaño. Cuando esto ocurre, se llama al método "redraw", que en este acaba añadiendo trabajos para el dibujado de las teselas correspondientes a la zona visible en la pantalla.

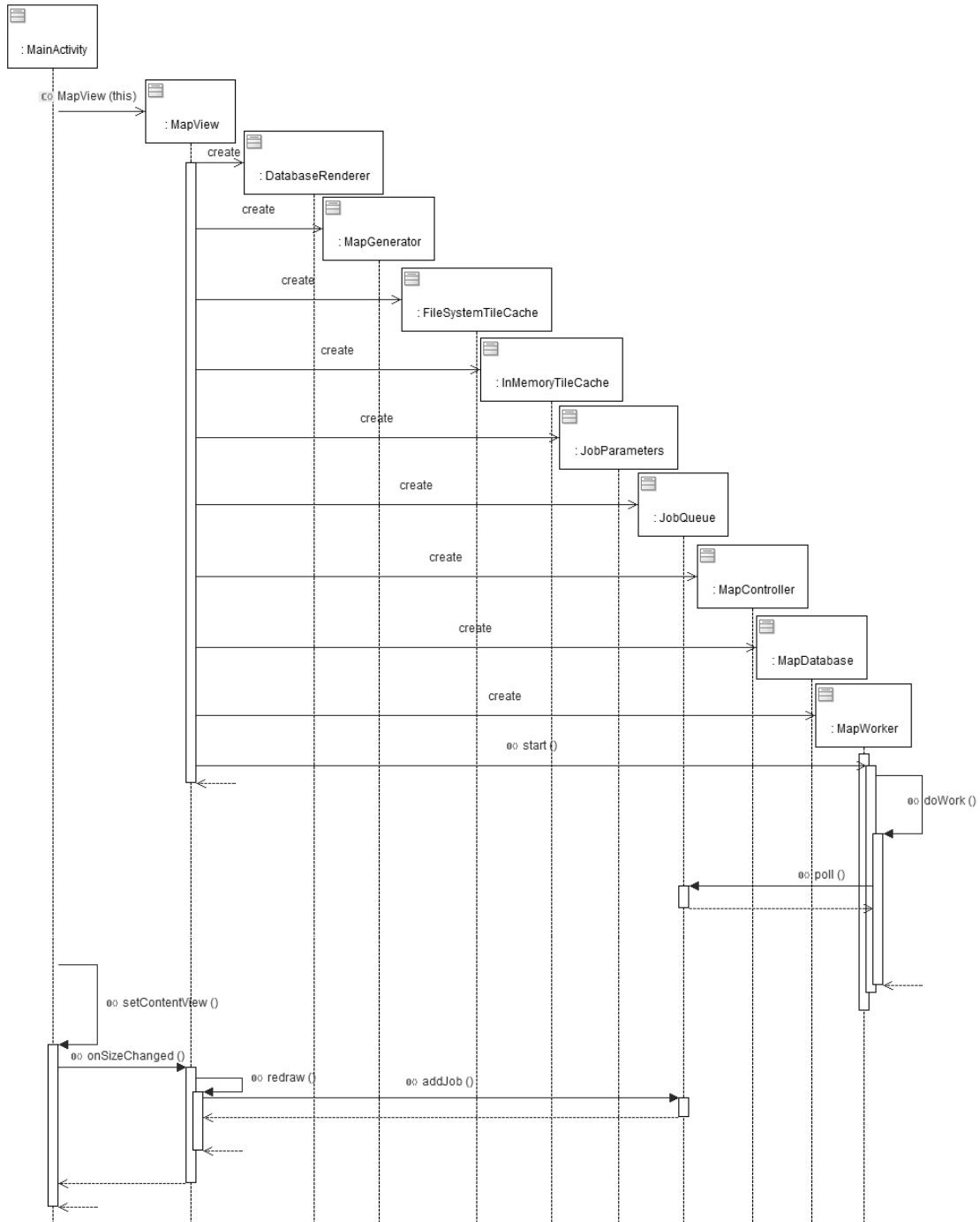


Figura I - 5 Diagrama de secuencia de la inicialización de los elementos de Mapsforge necesarios por parte de su clase MapView además del inicio del dibujado de las teselas.

3. Vías para la integración en la librería Android de GeosLab

La siguiente imagen, muestra la estructura de la librería de GeosLab en la que se va a integrar Mapsforge.

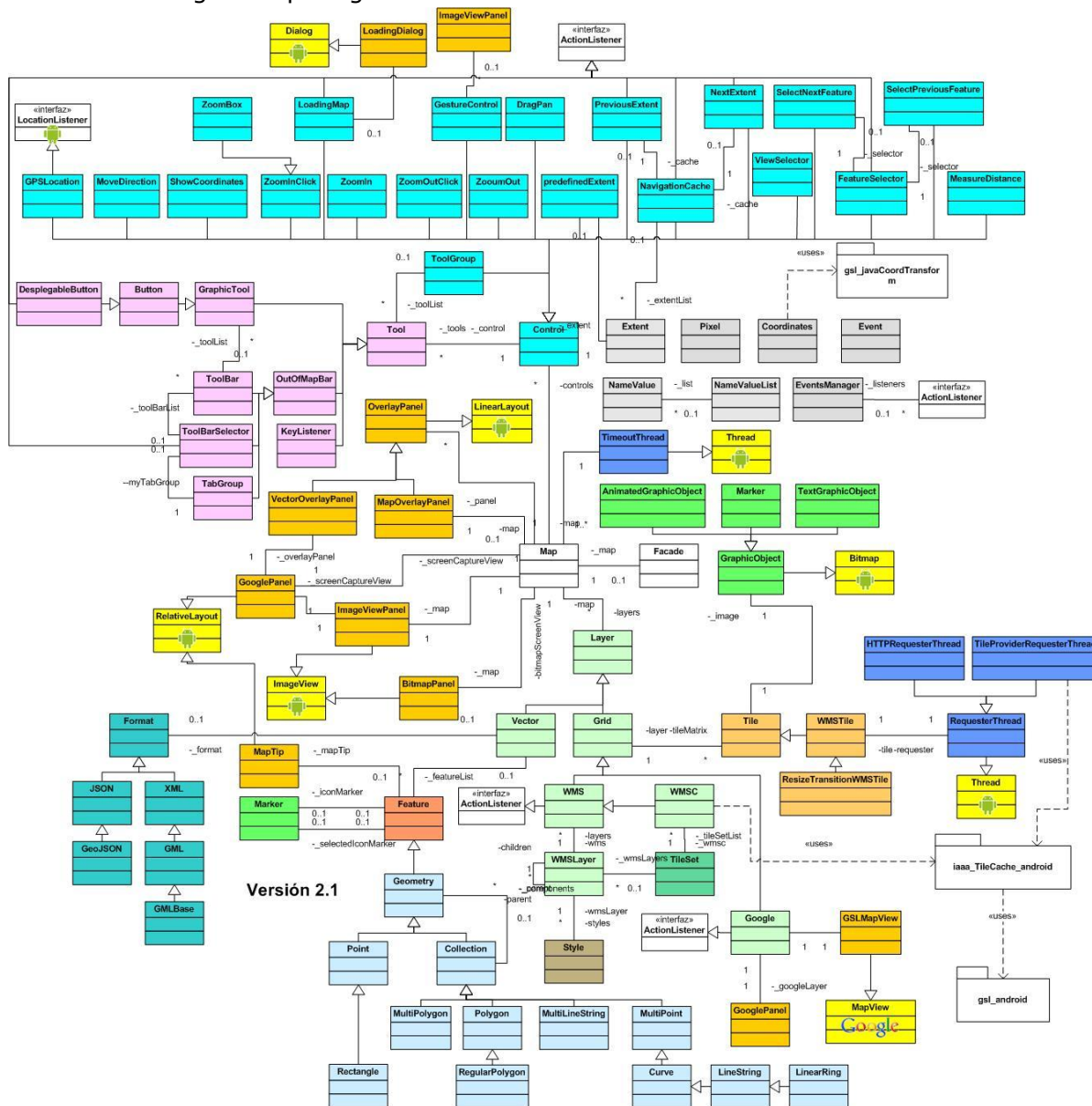


Figura I - 6 Diagrama de clases de la librería de GeosLab para Android.

Teniendo en cuenta esta estructura, y después de la información obtenida en el análisis, se han encontrado dos posibles vías de integración. La opción 1 sería crear una clase que extienda de la clase `Layer` o `Grid` de la librería de GeosLab y que haga las llamadas necesarias a la clase `MapView` de Mapsforge. La opción 2 sería crear una clase que extienda de `Layer` o `Grid` de GeosLab y que esa clase

asuma las funciones necesarias realizadas por la clase "MapView" de Mapsforge, de forma que esta ya no sea necesaria.

Opción 1 (Integración MapView)	Opción 2 (Sustitución MapView)
Ventajas	Ventajas
- Mayor facilidad ante cambios en la librería de Mapsforge (cambios de versiones).	- Mayor integración con la librería de GeosLab. - Mayor control sobre la librería Mapsforge.
Desventajas	Desventajas
- Menor control sobre la librería Mapsforge. - Mayor dificultad de integración.	- Acceso a clases internas de la librería de Mapsforge, lo que puede dificultar las actualizaciones de una versión a otra.

4. Diseño escogido para la integración

La opción escogida ha sido la opción 2 de sustitución de la clase MapView de Mapsforge, ya que de esa manera se podrá adaptar con más facilidad la librería de Mapsforge para su integración con la de GeosLab. Para llevar a cabo la integración, será necesario replicar parcialmente el código de la clase MapView de Mapsforge en la clase que se integrará en la librería. Esa clase se va a llamar Mapsforge y será la encargada de hacer las llamadas necesarias para el renderizado del mapa.

Dado que la librería de GeosLab tampoco dispone de soporte para mostrar mapas de OpenStreetMap online, se decidió también realizar la integración de estos. Los mapas de Mapsforge usan mapas de OpenStreetMap por lo que al integrar los mapas online directamente, la diferencia con Mapsforge sería únicamente el origen de las imágenes que componen las teselas. Para llevar esto a cabo se ha creado una clase llamada XYZ que manejará tanto los mapas online de OpenStreetMap como los mapas offline de Mapsforge. En la implementación realizada, la visualización de mapas de Mapsforge se ha considerado como un caso particular de los mapas de tipo XYZ en el que únicamente cambia la fuente de las teselas. El diagrama de clases que se muestra a continuación ilustra la estructura de las clases implementadas. El paquete XYZ contiene todas las clases involucradas con las capas de este tipo. Las clases dentro del paquete xyz.osm están destinadas a ser usadas por mapas de OpenStreetMap. Por ello se ha creado el paquete Mapsforge se encuentra dentro del xyz.osm, ya que como se puede ver se extienden algunas clases usadas en los mapas de OpenStreetMap.

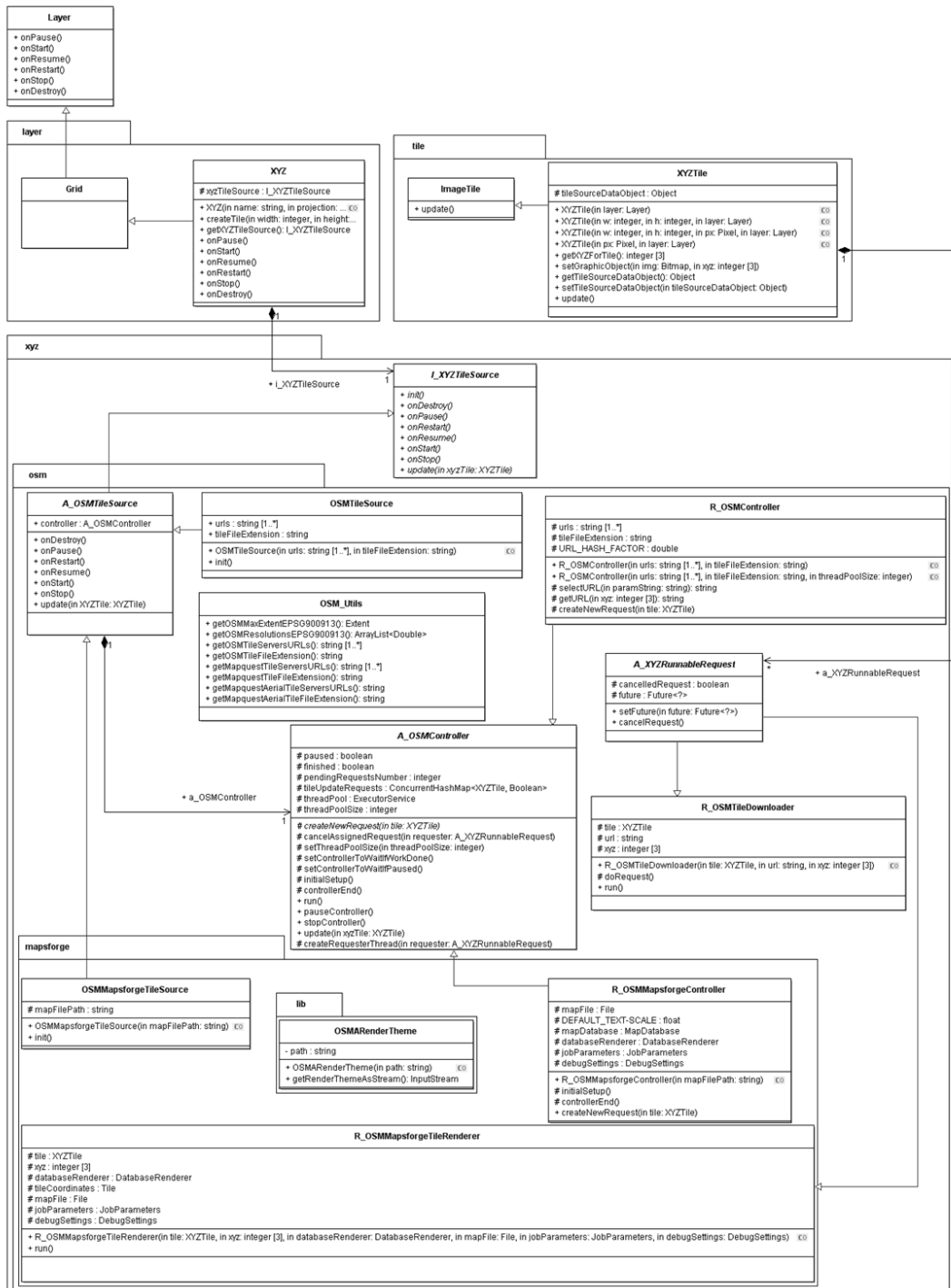


Figura I - 8 Diagrama de clases de la integración realizada de Mapsforge en la librería de GeosLab, así como las clases usadas para cargar las teselas de servicios tipo OpenStreetMap.

El diagrama anterior muestra las clases que se han implementado en la librería de GeosLab. A continuación se muestra información adicional sobre cada clase:

- **XYZ**
Esta clase corresponde a una capa pensada para su uso con diferentes servicios proveedores de teselas que usan una estructura tipo XYZ, como la usada por OpenStreetMap.
- **XYZTile**
Clase para las teselas usadas en la capa XYZ.
- **I_XYZTileSource**
Interfaz a implementar por las clases que necesiten crear un nuevo origen de datos de las teselas.
- **OSMUtils**
Clase auxiliar con métodos estáticos para facilitar la creación de capas basadas en OSM.
- **A_OSMTileSource**
Clase que agrupa funcionalidad común de la comunicación con la capa en la que está añadida.
- **OSMTileSource**
Clase que contiene la funcionalidad de comunicación con la capa de OSM en la que está añadida.
- **A_OSMController**
Clase abstracta que agrupa funcionalidad del controlador común a clases correspondientes a capas estilo OSM.
- **R_OSMController**
Clase correspondiente al controlador específico de carga de teselas de OSM.
- **A_XYZRunnableRequest**
Clase abstracta que agrupa funcionalidad de las clases encargadas de obtener las imágenes de las teselas solicitadas.
- **R_OSMTileDownloader**
Clase encargada de la descarga de una imagen de una tesela que se pasa a un pool de hilos para su ejecución.
- **OSMMapsforgTileSource**
Clase que crea el origen de datos de tiles para Mapsforg.
- **OSMAREnderTheme**
Tema para el renderizado de teselas.
- **R_OSMMapsforgController**
Clase correspondiente al controlador específico de carga de teselas de OSM.
- **R_OSMMapsforgTileRenderer**
Clase encargada de realizar las llamadas necesarias a la librería de Mapsforg para realizar el renderizado de las teselas. La petición se pasa se pasa a un pool de hilos de uno sólo hilo dado que Mapsforg actualmente no permite peticiones desde múltiples hilos. Se usa un pool aunque sea de tamaño 1, porque ese hilo se reutiliza entre las diferentes peticiones ahorrando costes de creación y destrucción de hilos.

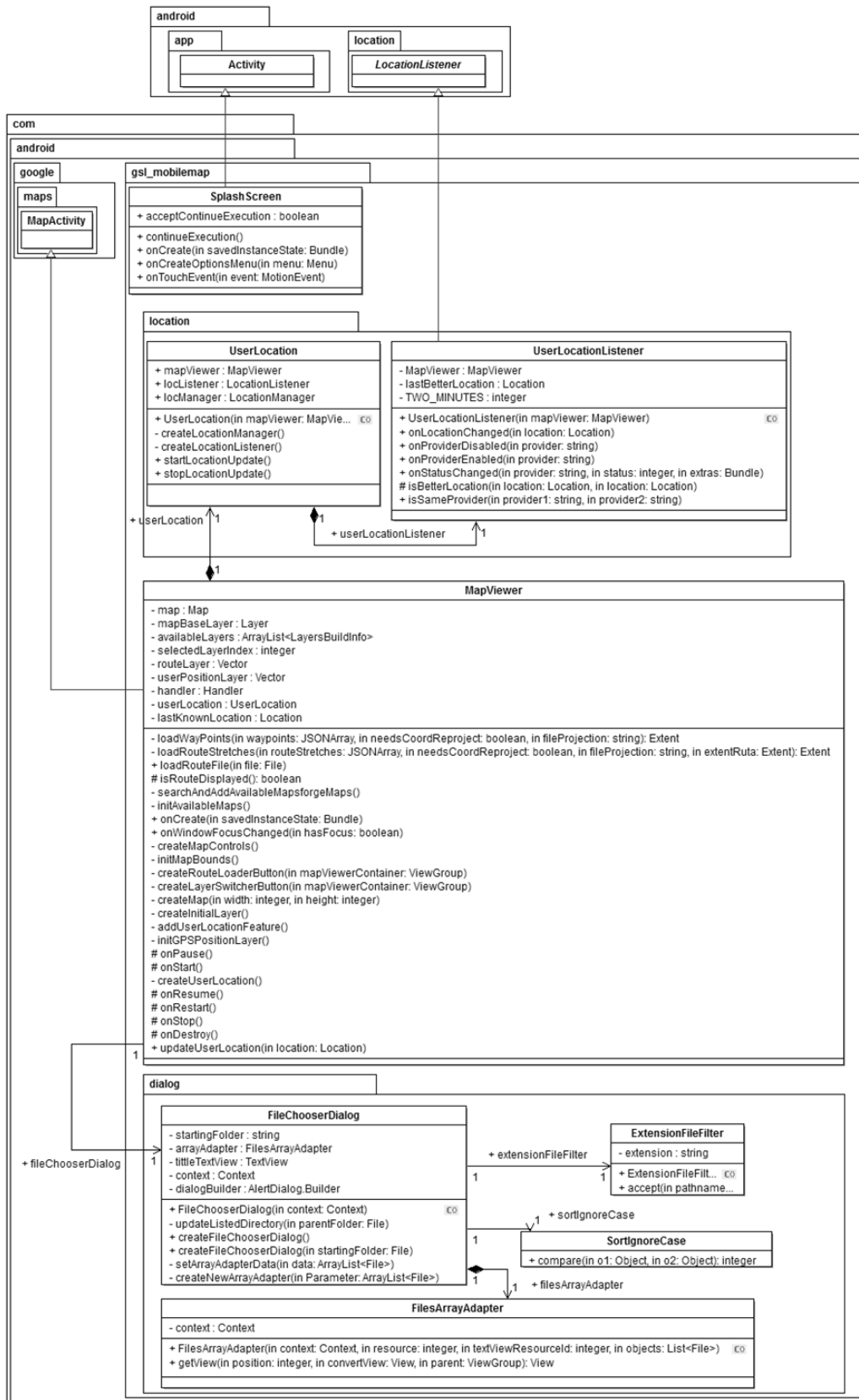


Figura I - 9 Diagrama de clases de la aplicación creada para el caso de uso y que utilice la librería de GeosLab y el trabajo realizado en ella.

El diagrama anterior muestra las clases que se han implementado en la aplicación cliente. A continuación se muestra información adicional sobre cada clase:

- **SplashScreen**
Esta clase muestra una SplashScreen en el inicio de la aplicación e inicia la actividad de la aplicación en si tras un toque del usuario en la pantalla.
- **UserLocation**
Clase encargada de gestionar la localización del usuario.
- **UserLocationListener**
Listener de la ubicación del usuario.
- **MapView**
Actividad principal de la aplicación de Android correspondiente al visor del mapa.
- **FileChooserDialog**
Implementación de un selector de ficheros en Android que permite elegir entre carpetas o archivos .txt usado para la selección y carga de archivos de rutas.
- **ExtensionFileFilter**
Filtro de tipos de archivos que sólo acepta archivos que tengan una extensión concreta o bien si son directorios.
- **SortIgnoreCase**
Clase para ordenar un array según el contenido sin diferenciar mayúsculas de minúsculas.
- **FilesArrayAdapter**
ArrayAdapter usado para almacenar la información de los archivos que se están mostrando en el dialog de selección ficheros.

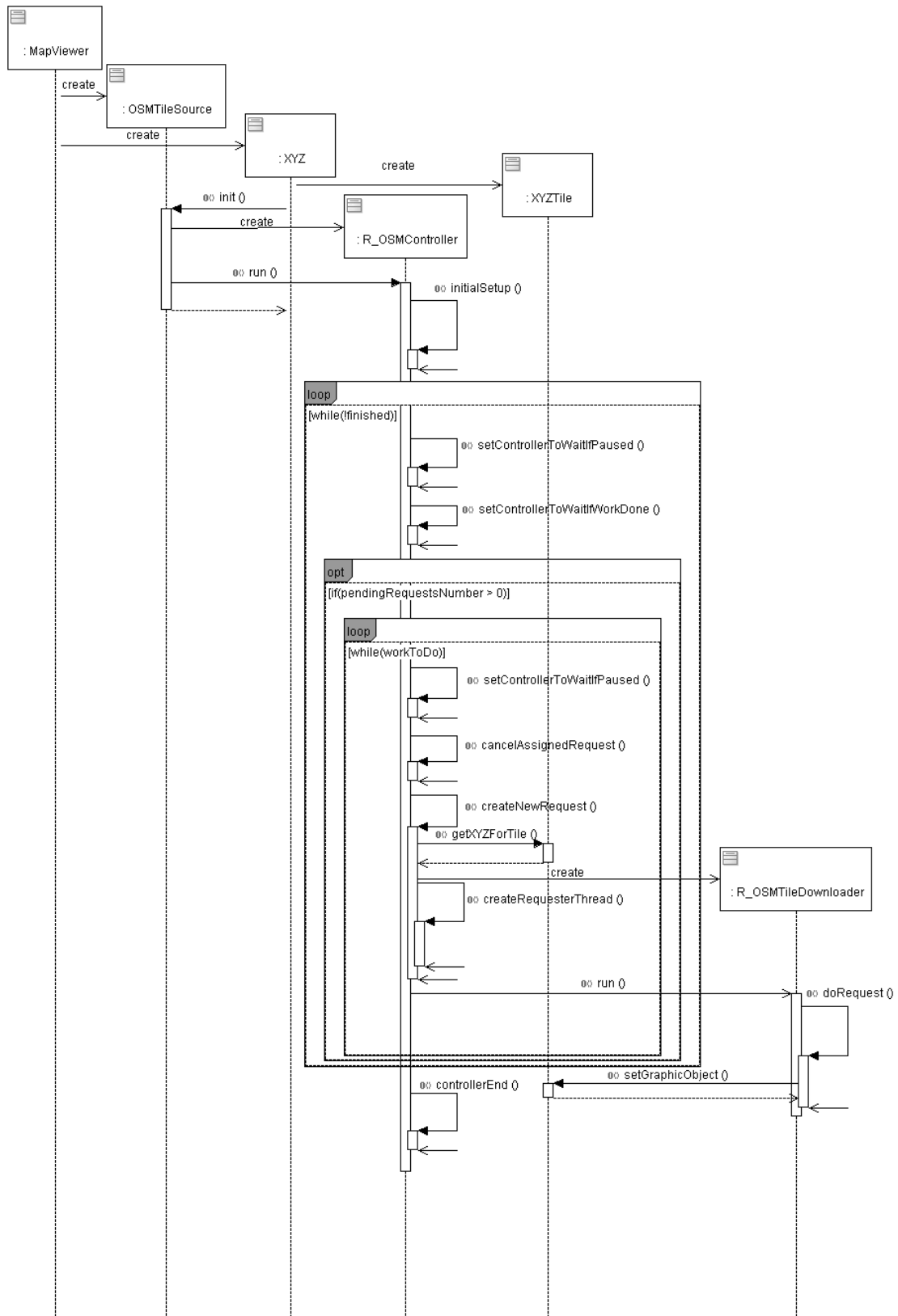


Figura I - 10Diagrama de secuencia de la carga de teselas de las capas basadas en teselas de OpenStreetMap como OSM o Mapquest.

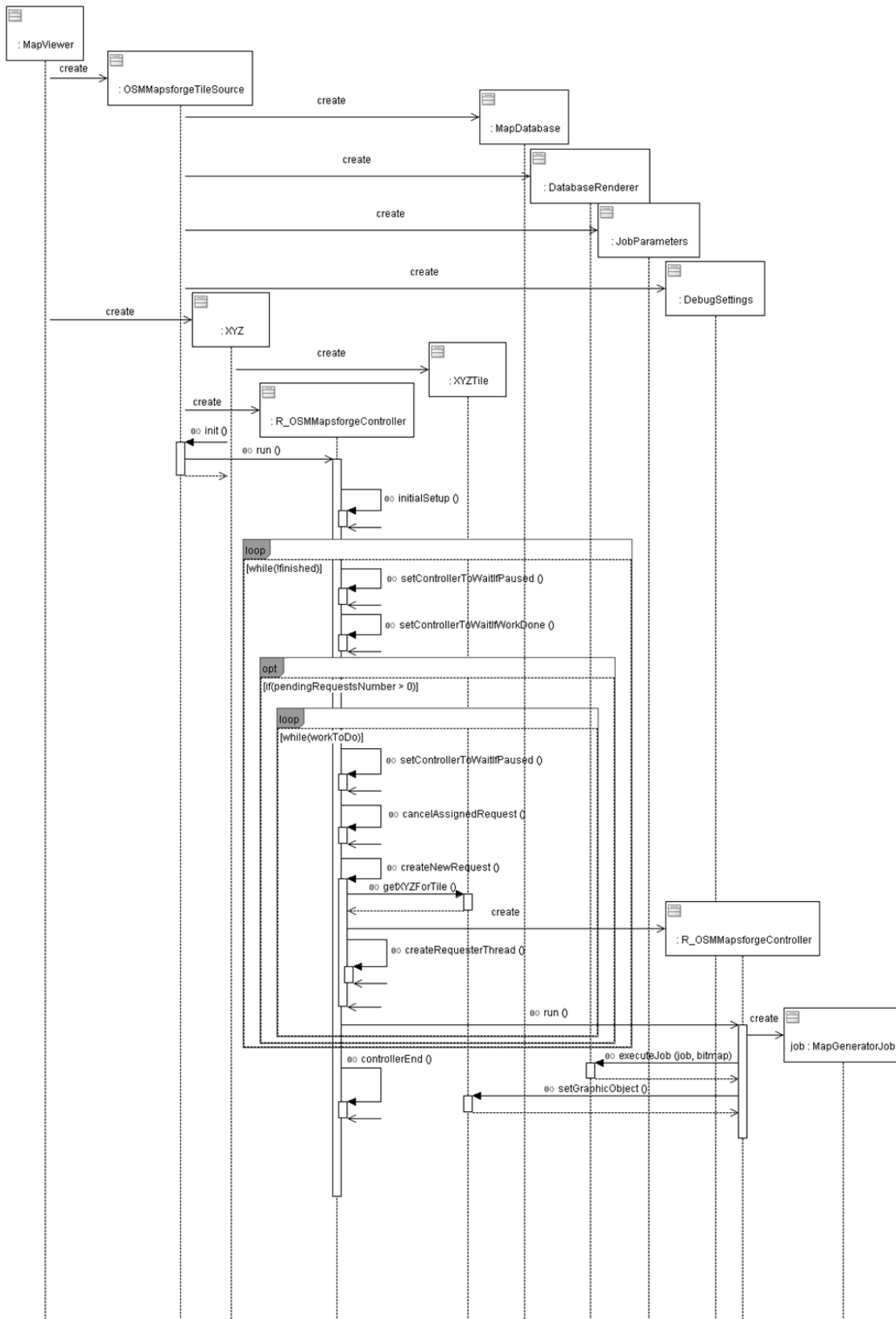


Figura I - 11 Diagrama de secuencia de la carga de teselas utilizando las clases de la librería de Mapsforge y la información del archivo del mapa en el dispositivo.

Anexo J. Casos de uso

En este anexo se van a mostrar los casos de uso creados para ambas partes y la funcionalidad en conjunto de ambos.

1. Parte Web

El caso de uso de la parte Web se ha configurado para que muestre todas las capas que se han creado en OpenLayers así como otras capas de referencia como las de Google, u OpenStreetMap.

La siguiente imagen muestra todas las capas seleccionables. Como capas base se pueden elegir las capas de Yandex del mapa en ruso y en inglés, Yandex satélite, Google Streets, Google Satélite, OpenStreetMap, mapa de Baidu, satélite de Baidu, y la capa del mapa integrada mediante la API de Baidu. Como capas de overlay están disponibles la siguiente capas: la capa que muestra los nombres del mapa de Baidu para usar con otras capas base como la de satélite y las alertas de Baidu (ambas sólo son seleccionables cuando se tiene como capa base una capa de Baidu), la capa de tráfico de Yandex que muestra los semáforos con la congestión del tráfico y en niveles más cercanos de zoom muestra las teselas que muestran visualmente la congestión del tráfico por calles, la capa de información del tráfico de Yandex que permite mostrar la velocidad en cada calle mediante un popup, las capas de nombres de Yandex que contienen la información del mapa usando transparencias de modo que se pueda superponer a otra capa base como por ejemplo una capa de satélite. Por último, está disponible también la capa de nombres de Baidu para usarla sobre una capa de satélite por ejemplo.

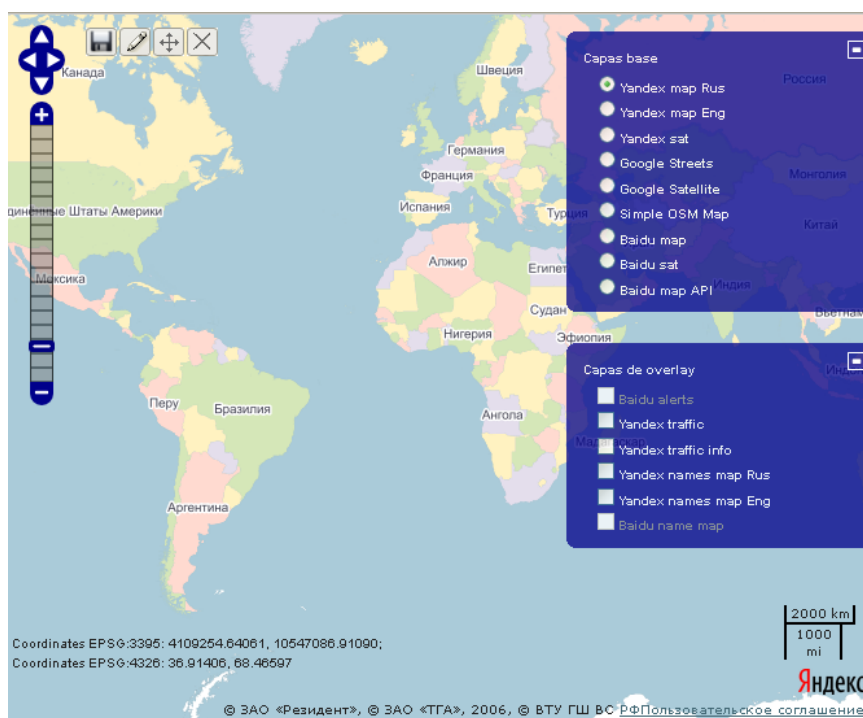


Figura J - 1 Imagen que muestra todas las capas disponibles. El desplegable superior permite seleccionar la capa base, y el desplegable de la parte inferior permite seleccionar la capa de overlay.

A continuación se muestra una imagen de cada una de las capas de Yandex y Baidu que se han integrado en OpenLayers por el trabajo realizado en este PFC.

Capas base:

a) Yandex mapa en Ruso

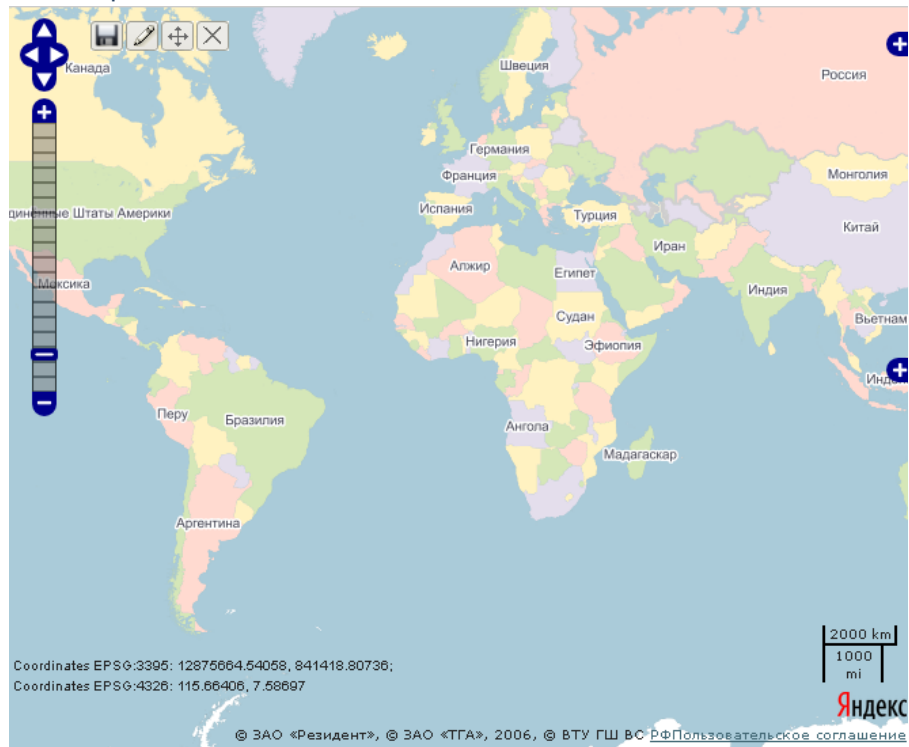


Figura J - 2 Mapa mostrando la capa del mapa de Yandex en ruso.

b) Yandex mapa en inglés

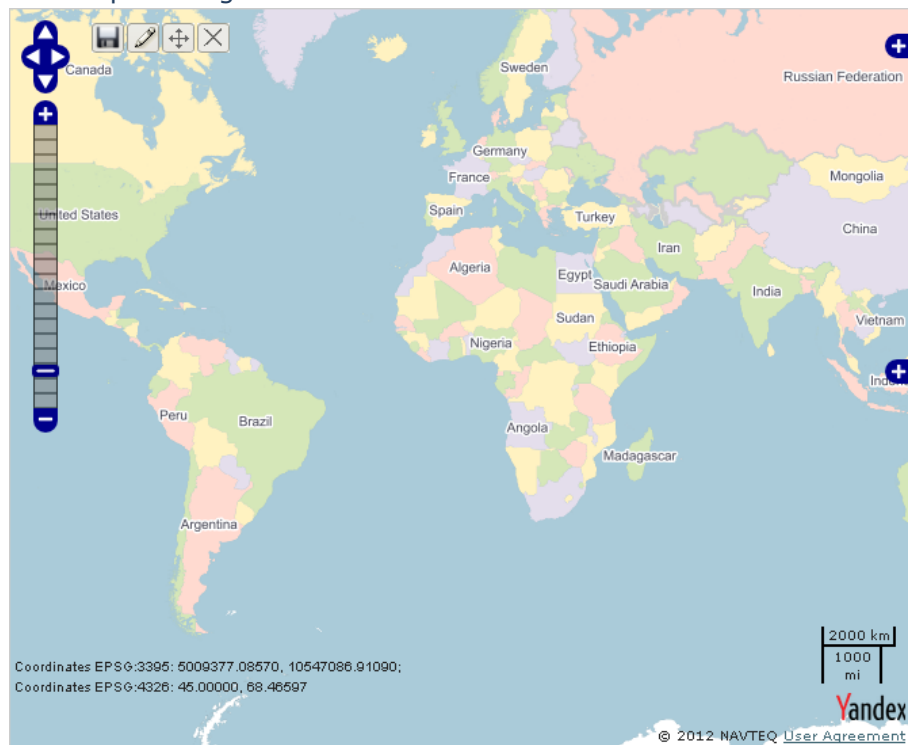


Figura J - 3 Mapa mostrando la capa del mapa de Yandex en inglés.

c) Yandex satellite



Figura J - 4 Mapa mostrando la capa de satélite de Yandex.

d) Baidu mapa



Figura J - 5 Mapa mostrando la capa de mapa de Baidu.

e) Baidu satélite



Figura J - 6 Mapa mostrando la capa de satélite de Baidu

f) Baidu mapa usando la API



Figura J - 7 Mapa mostrando la capa de mapa de Baidu de su API.

Overlays:

a) Baidu alertas



Figura J - 8 Mapa mostrando la capa de alertas de Baidu



Figura J - 9 Ejemplo de una incidencia ocurrida en China mostrada sobre el mapa tomando la información de una incidencia del servicio de Baidu. Si el usuario lo desea, es posible usar herramientas de traducción automática como la incluida en el navegador Google Chrome para traducir la información de la incidencia de forma automática.

b) Yandex tráfico

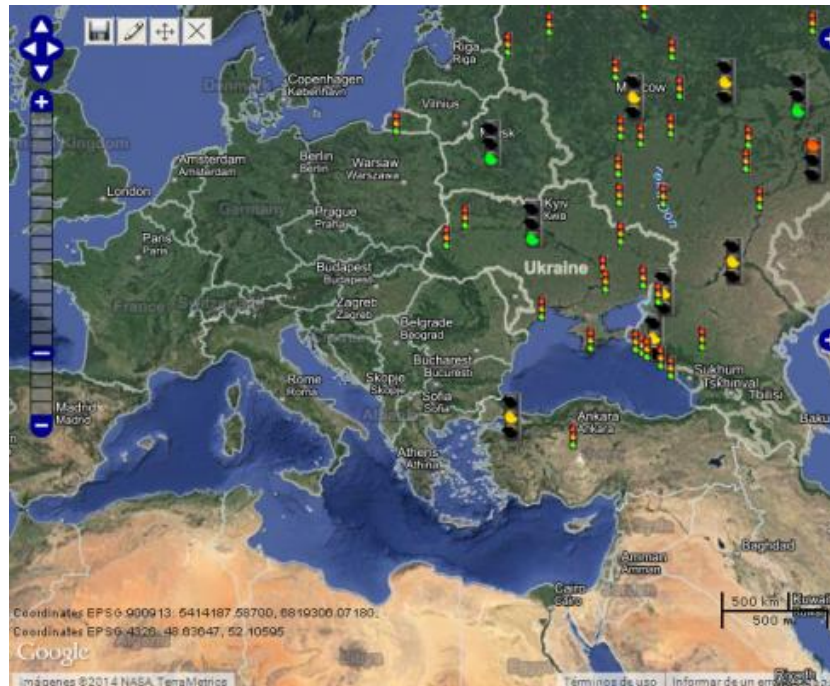


Figura J - 10 Mapa mostrando la capa de tráfico de Yandex. En la imagen se muestran semáforos de varios tipos. Los semáforos con código de colores indican el nivel de congestión del tráfico, mientras que los semáforos de menor tamaño que tienen los 3 colores sólo indican la presencia del servicio de tráfico en esa ciudad, aunque no dispone de información global del tráfico.

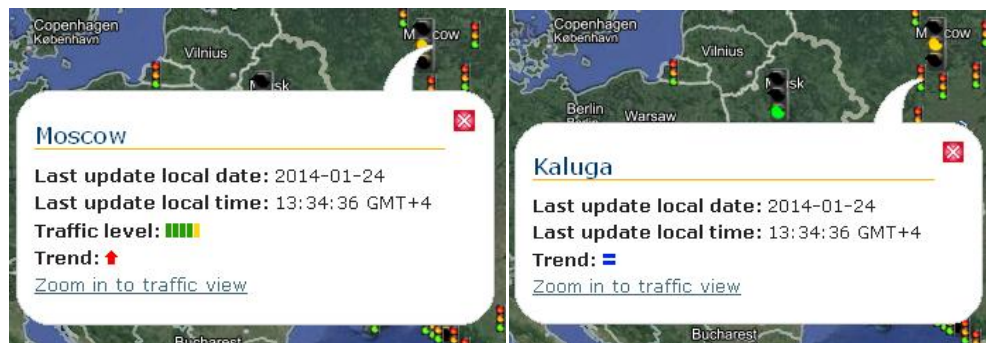


Figura J - 11 La imagen de la izquierda corresponde al aviso mostrado al hacer clic sobre un semáforo con código de colores. En esta información se incluye la ciudad sobre la que se ha realizado el clic, la última actualización de la información de tráfico, el nivel de carga del tráfico (sólo disponible en ciudades con semáforo de código de colores) así como una tendencia del tráfico. Además en el popup se incluye un enlace que permite acceder directamente al nivel de zoom en el que se muestra el tráfico por calles en esa zona. La imagen de la derecha corresponde a la información mostrada por los semáforos de menor tamaño que tienen los 3 colores.

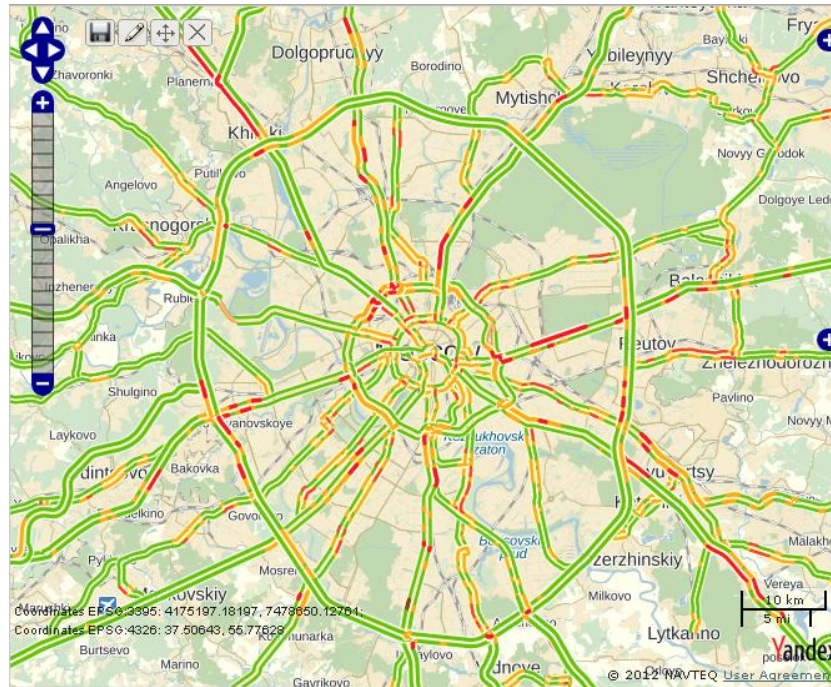


Figura J - 12 Mapa mostrando la capa de tráfico con la información de la congestión del tráfico a nivel de calles.

c) Yandex información del tráfico

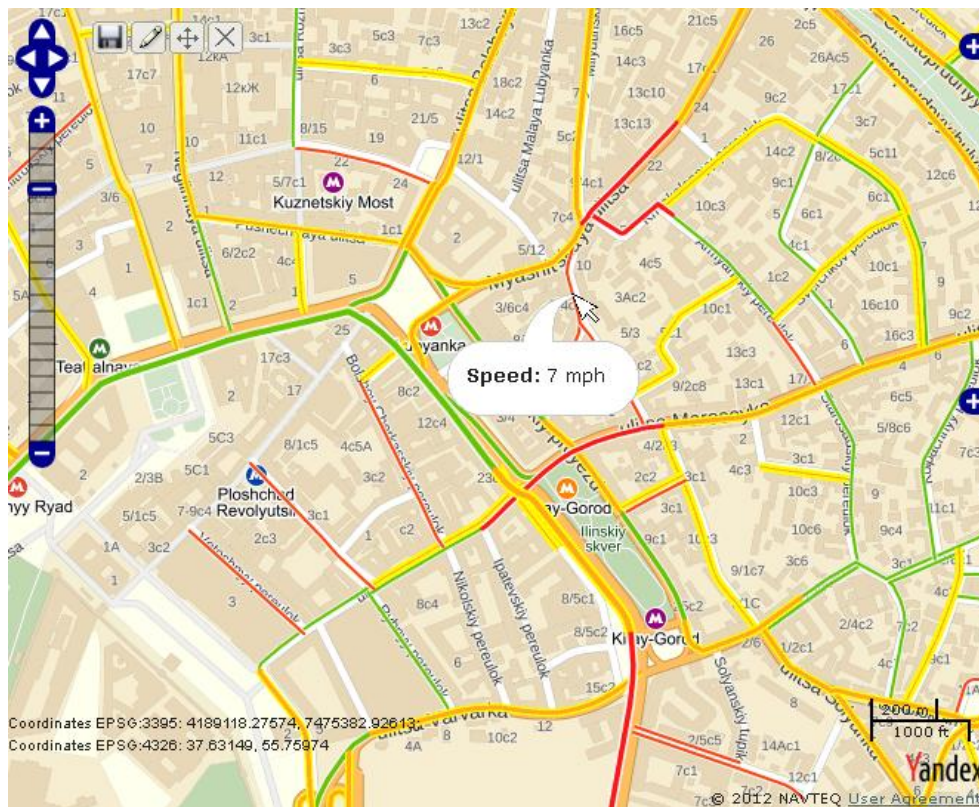


Figura J - 13 Mapa mostrando la capa de tráfico de Yandex además de la capa de información de tráfico de Yandex que muestra la velocidad en los diferentes tramos al pasar el ratón sobre ese punto.

d) Yandex nombres del mapa en ruso



Figura J - 14 Mapa mostrando la capa de nombres en ruso de Yandex usando como capa base la capa de satélite de Yandex.

e) Yandex nombres del mapa en inglés



Figura J - 15 Mapa mostrando la capa de nombres en inglés de Yandex usando como capa base la capa de satélite de Yandex.

f) Baidu nombres del mapa

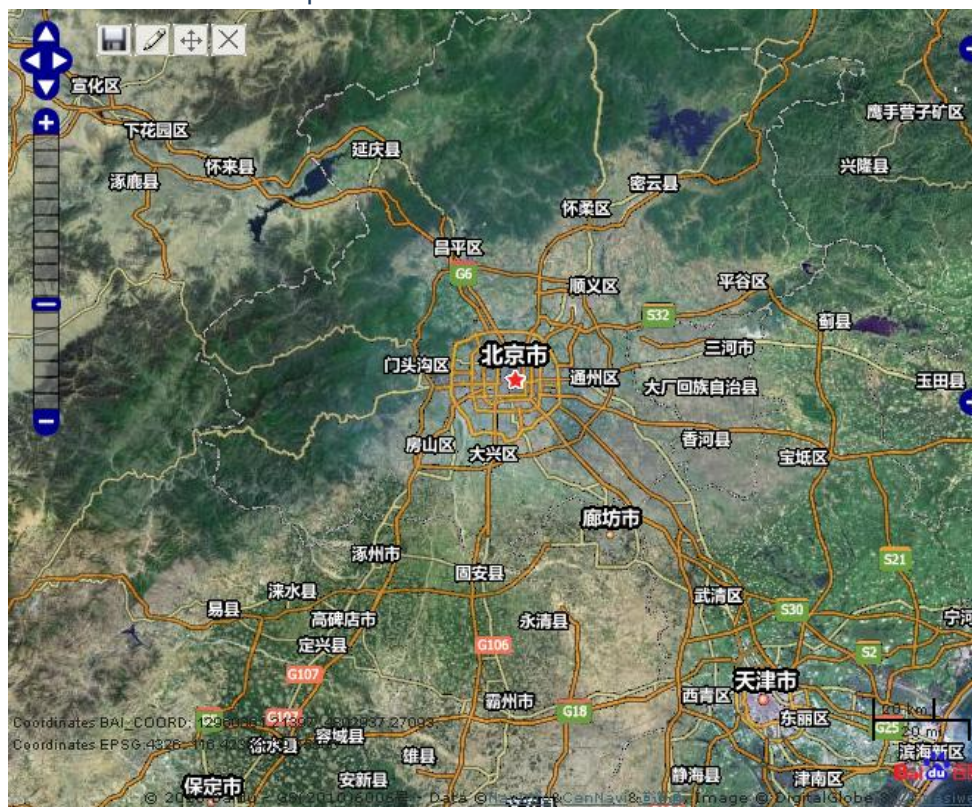


Figura J - 16 Mapa mostrando la capa de nombres de Baidu usando como capa base la capa de satélite de Baidu.

2. Parte Móvil

El caso de uso de la parte móvil se ha creado de modo que muestre todas las capas que se han creado para la librería de GeosLab. Al arrancar lo primero que se ve es una pantalla con la información del desarrollo como se puede ver en la figura



Figura J - 17 Pantalla inicial de la aplicación creada como caso de uso.

Una vez pasada la pantalla inicial (SplashScreen) se muestra una vista de un mapa mostrando por defecto la capa de Mapquest basada en OpenStreetMap (OSM) que requiere conexión a Internet para mostrar las teselas (Figura J - 18). El usuario también dispone de dos botones en la parte superior derecha de la pantalla. El botón superior permite cambiar la capa que se muestra en el mapa, y el botón debajo de él permite la carga de una ruta de un archivo.



Figura J - 18 Capa de Mapquest basada en OSM.

En la aplicación creada como demo están disponibles para selección las capas de OSM oficial, OSM Mapquest Streets (capa de Mapquest basada en OpenStreetMap), OSM Mapquest Aerial (capa de Mapquest basada en tiles aéreas con licencia abierta y con estructura estilo OSM). Además de estas capas, es posible elegir capas adicionales si el dispositivo tiene mapas de Mapsforge cargados. La Figura J - 19 muestra la lista de capas disponibles para un dispositivo que dispone de los mapas de Mapsforge para China, Portugal, Rusia y España.

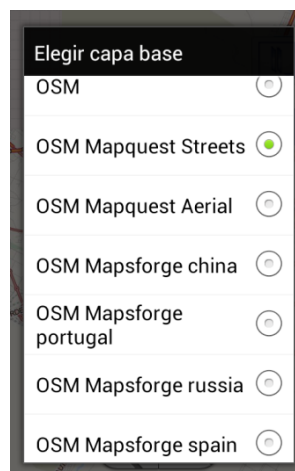


Figura J - 19 Menú de selección de capa en la aplicación de Android creada.

A continuación se va a mostrar una captura de pantalla de los diferentes tipos de capas disponibles:

- OSM:

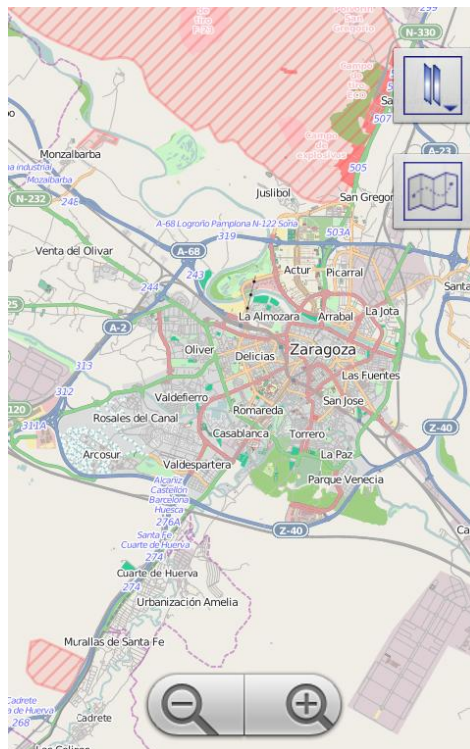


Figura J - 20 Capa de OSM.

- OSM Mapquest Streets:



Figura J - 21 Capa de calles de Mapquest basada en OpenStreetMap

- OSM Mapquest Aerial:



Figura J - 22 Capa de imágenes aéreas de Mapquest basada en imágenes de reconocimiento con licencia abierta y con una estructura similar a OSM. Tiene buen nivel de zoom para EEUU, fuera de ahí el zoom está limitado siendo el máximo el que se muestra en esta imagen que corresponde al área de Zaragoza.

- OSM Mapsforge China:

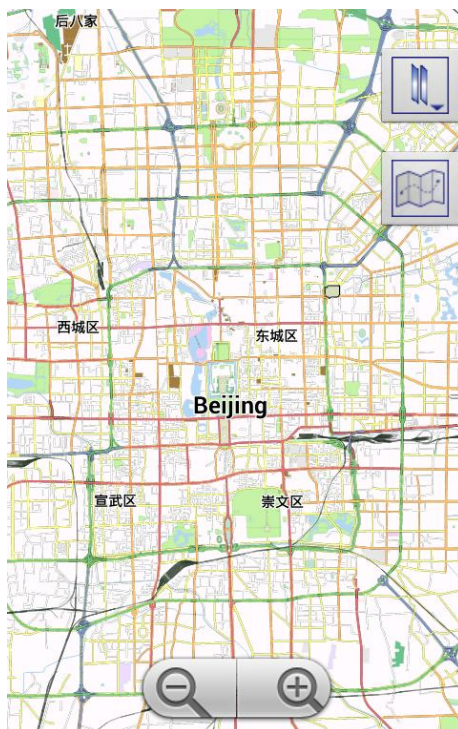


Figura J - 23 – Capa de Mapsforge mostrando el mapa offline para China.

- OSM Mapsforge Rusia:

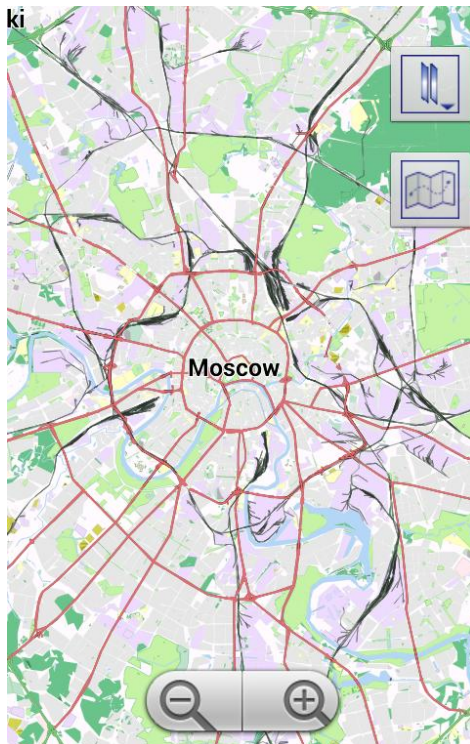


Figura J - 24Capa de Mapsforge mostrando el mapa offline para Rusia.

- OSM Mapsforge Spain:

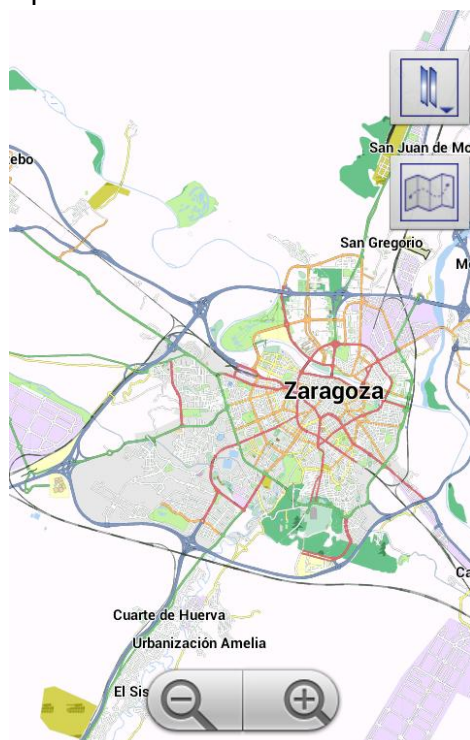


Figura J - 25Capa de Mapsforge mostrando el mapa offline para España.

Al cargar un archivo de ruta, el mapa cambia a la posición donde se encuentre la ruta mostrando la ruta completa en la pantalla. Además si es posible la localización del usuario bien mediante GPS o ubicación por red, se muestra en pantalla un marcador negro con la posición actual del usuario. Al tocar el botón para cargar la ruta, se muestra al usuario un dialog para permitirle la selección del archivo de ruta que quiere cargar (Figura J - 26 imagen de la izquierda). Utilizando ese dialog puede navegar por el sistema de archivos del teléfono y buscar la ubicación dónde se encuentra el archivo que quiere cargar y cargarlo con un toque (Figura J - 26 imagen de la derecha).

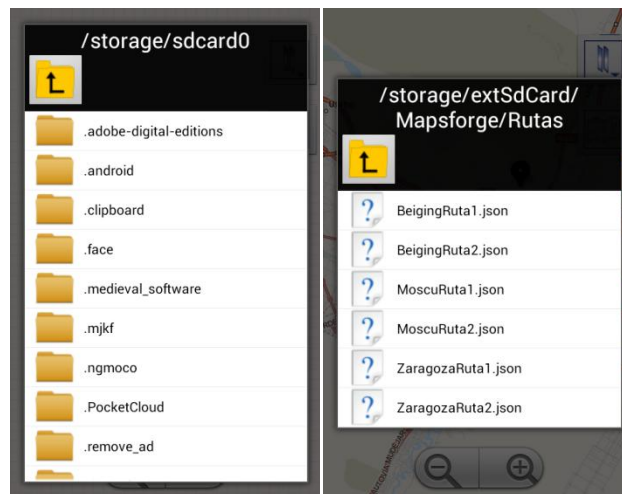


Figura J - 26 - Dialog de selección de archivo de ruta.

La Figura J - 27 muestra una ruta en Zaragoza, y la ubicación del usuario que en este caso corresponde al edificio Ada Byron de la Escuela de Ingeniería y Arquitectura (EINA) de la Universidad de Zaragoza que es donde se tomo la captura de pantalla.

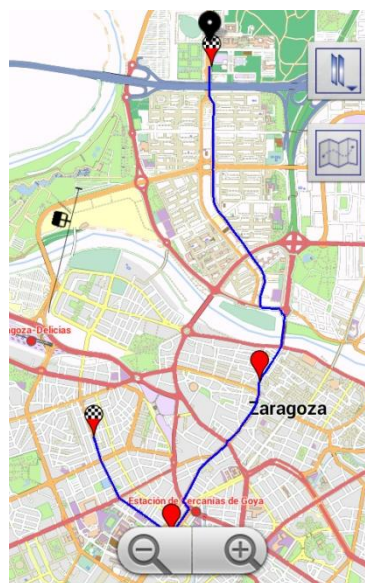


Figura J - 27 Ruta en Zaragoza mostrando la posición del usuario en la EINA de la Universidad de Zaragoza.

La Figura J - 28 muestra dos rutas de ejemplo utilizando los respectivos mapas de Mapsforge como capas base.

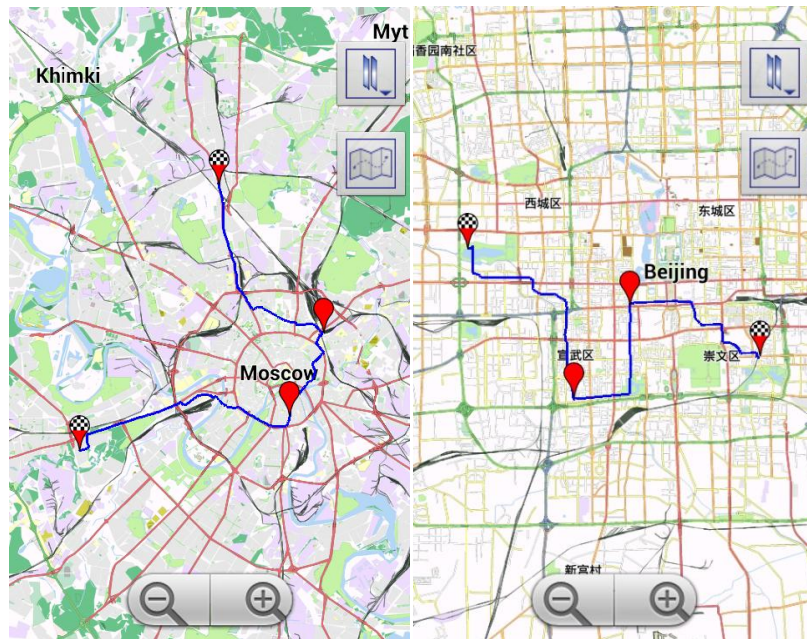


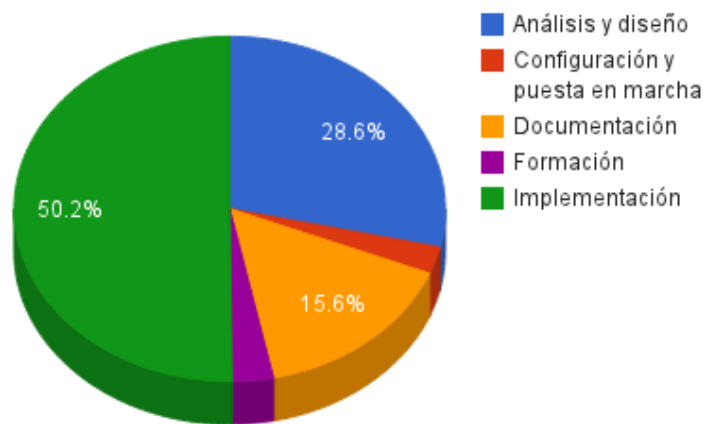
Figura J - 28 La imagen de la izquierda muestra una ruta de ejemplo en Moscú cargada sobre el mapa offline de Mapsforge de Rusia. La imagen de la derecha muestra una ruta de ejemplo en Beijing (Pekín) sobre el mapa offline de Mapsforge de China.

Anexo K. Resumen de horas del PFC

En este anexo se incluye un resumen de las horas que se han dedicado al desarrollo de este PFC. En total se han dedicado más de 1563 horas a todo el PFC. Todas las tareas del proyecto se han clasificado en varias categorías (Documentación, implementación, configuración y puesta en marcha, análisis y formación). La siguiente tabla muestra las horas realizadas de cada una de las categorías mencionadas.

Categorías:	Horas totales:
Análisis y diseño	446,8333333333333
Configuración y puesta en marcha	39,58333333333333
Documentación	244,4333333333333
Formación	48,1
Implementación	784,3333333333333
Total:	1563,2833333333333

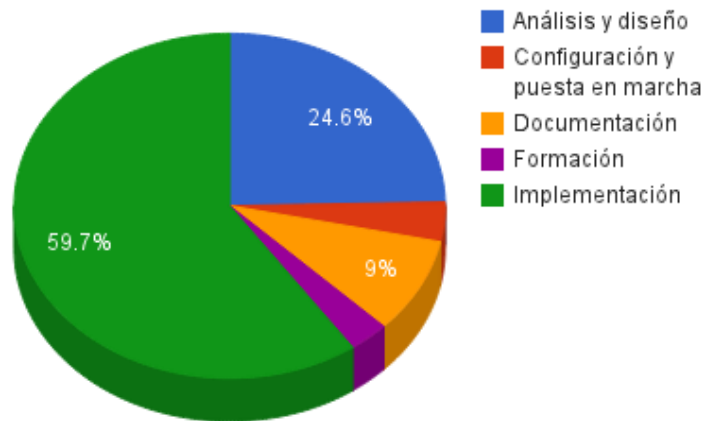
Distribución de horas en el PFC



Distribución de horas de la parte Web

Categorías:	Horas totales:
Análisis y diseño	243,6666666666667
Configuración y puesta en marcha	36,66666666666667
Documentación	88,75
Formación	29,68333333333333
Implementación	590
Total:	988,7666666666667

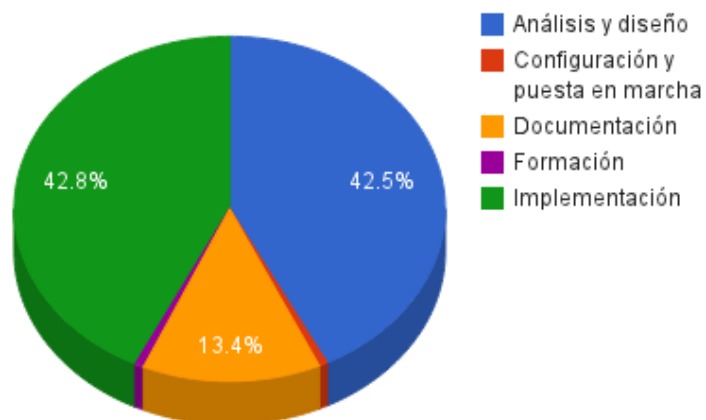
Distribución de horas parte Web



Distribución de horas de la parte Android

Categorías:	Horas totales:
Análisis y diseño	193,166666666667
Configuración y puesta en marcha	2,91666666666667
Documentación	60,75
Formación	3
Implementación	194,333333333333
Total:	454,166666666667

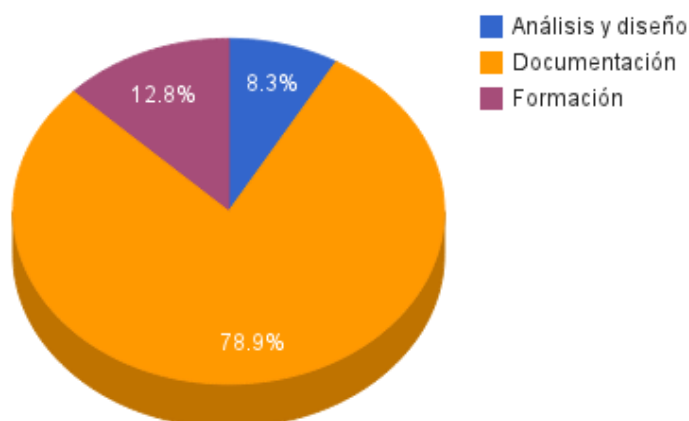
Distribución de horas parte Android



Distribución de horas en tareas no exclusivas de Web o Android.

Categorías:	Horas totales:
Análisis y diseño	10
Documentación	94,93333333333333
Formación	15,41666666666667
Total:	120,35

Distribución de horas partes comunes

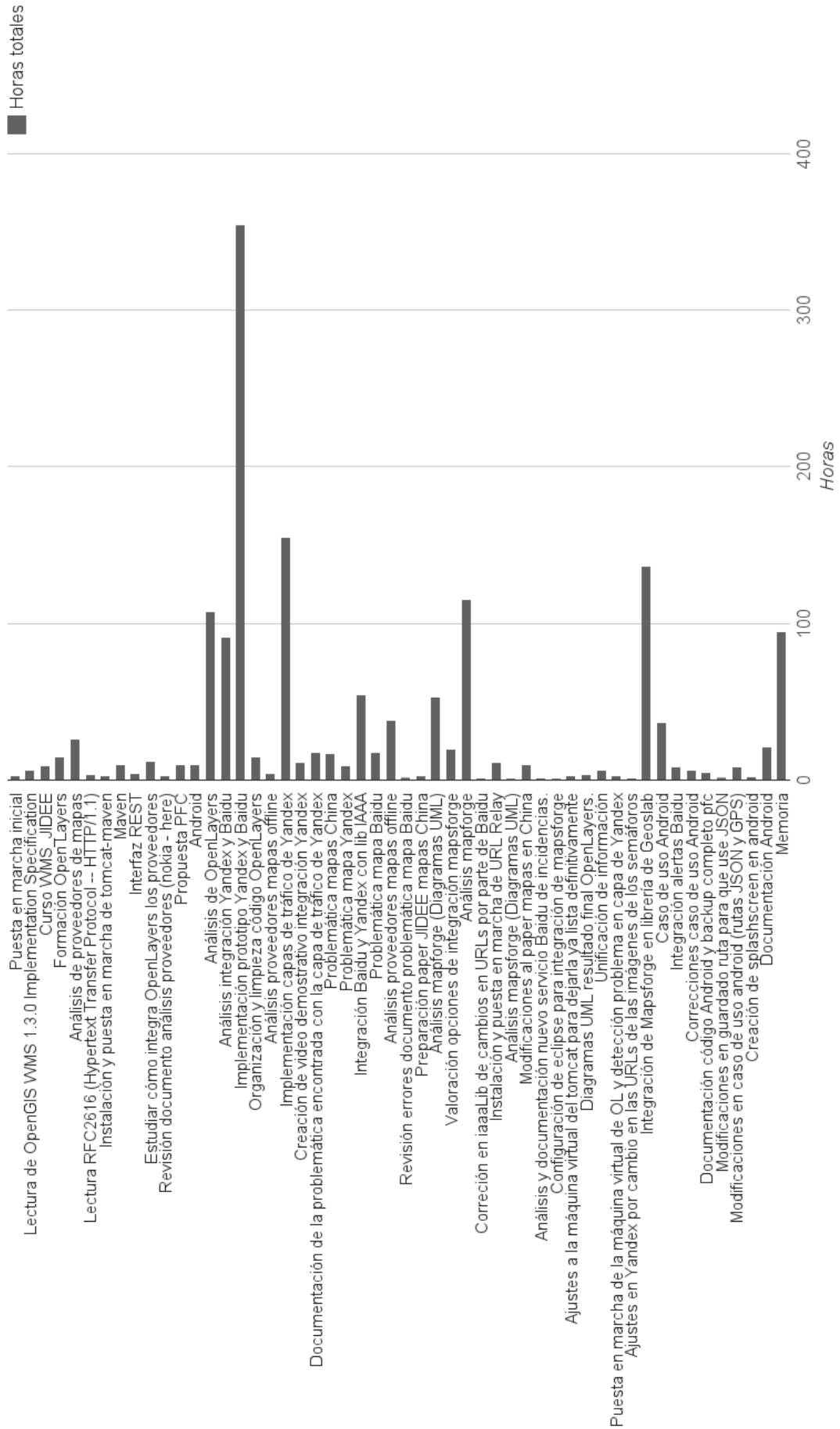


Por último se incluye la lista de todas las tareas realizadas a lo largo del PFC además de un gráfico de barras dónde se puede ver una comparación del tiempo dedicado a cada una de esas tareas

Tarea	Horas totales
Puesta en marcha inicial	2,75
Lectura de OpenGIS WMS 1.3.0 Implementation Specification	6,25
Curso WMS_JIDEE	9,16666666666667
Formación Open Layers	15,21666666666667
Análisis de proveedores de mapas	26,41666666666667
Lectura RFC2616 (Hypertext Transfer Protocol -- HTTP/1.1)	3,21666666666667
Instalación y puesta en marcha de tomcat-maven	3
Maven	10,16666666666667
Interfaz REST	4
Estudiar cómo integra OpenLayers los proveedores	11,91666666666667
Revisión documento análisis proveedores (Nokia - Here)	3
Propuesta PFC	10
Android	10,25
Análisis de OpenLayers	107,75
Análisis integración Yandex y Baidu	91
Implementación prototipo Yandex y Baidu	355
Organización y limpieza código OpenLayers	15
Análisis proveedores mapas offline	4
Implementación capas de tráfico de Yandex	155
Creación de video demostrativo integración Yandex	11
Documentación de la problemática encontrada con la capa de	18

tráfico de Yandex	
Problemática mapas China	17
Problemática mapa Yandex	9
Integración Baidu y Yandex con lib IAAA	54,5
Problemática mapa Baidu	18
Análisis proveedores mapas offline	38
Revisión errores documento problemática mapa Baidu	1,5
Preparación paper JIDEE mapas China	2,5
Análisis Mapsforge (Diagramas UML)	53
Valoración opciones de integración Mapsforge	20
Análisis Mapsforge	115,5
Corrección en iaaaLib de cambios en URLs por parte de Baidu	1
Instalación y puesta en marcha de URL Relay	11,08333333333333
Análisis Mapsforge (Diagramas UML)	0,9166666666666667
Modificaciones al paper mapas en China	9,666666666666667
Análisis y documentación del nuevo servicio Baidu de incidencias.	1,0833333333333333
Configuración de eclipse para integración de Mapsforge	0,9166666666666667
Ajustes a la máquina virtual del Tomcat para dejarla ya lista definitivamente	2,9166666666666667
Diagramas UML resultado final OpenLayers.	3,5833333333333333
Unificación de información	6,25
Puesta en marcha de la máquina virtual de OL y detección problema en capa de Yandex	2,5
Ajustes en Yandex por cambio en las URLs de las imágenes de los semáforos	1
Integración de Mapsforge en librería de GeosLab	136,33333333333333
Caso de uso Android	36,5
Integración alertas Baidu	8,5
Correcciones caso de uso Android	6,5
Documentación código Android y backups completo PFC	5
Modificaciones en guardado ruta para que use JSON	1,5
Modificaciones en caso de uso Android (rutas JSON y GPS)	8,5
Creación de SplashScreen en Android	2
Documentación Android	21,5
Memoria	94,93333333333333
Total:	1563,2833333333333

Distribución de horas por tareas



5. Bibliografía

- [1] Curso WMS JIDEE Jorge Álvarez Poza y Javier Eced Cerdán (iaaa - Universidad de Zaragoza)
- [2] Guía de OpenLayers 1.2 Jorge Álvarez Poza (iaaa - Universidad de Zaragoza)
- [3] OpenGIS WMS 1.3.0 Implementation Specification Open Geospatial Consortium Inc.
- [4] RFC2616 (Hypertext Transfer Protocol -- HTTP/1.1) Internet Engineering Task Force (IETF)
- [5] Wikipedia Fundación Wikimedia
- [6] REST vs WebServices Rafael Navarro Maset - Universidad Politécnica de Valencia
<http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebService.s.pdf>
- [7] WGS84
- [8] <http://wiki.openstreetmap.org/wiki/> OpenStreetMap
- [9] The Mercator projections Peter Osborne
- [10] Map Projections - A working manual US Geological Survey
Professional paper 1395 by John P. Snyder
<http://pubs.er.usgs.gov/publication/pp1395>
- [11] Guía y referencias de desarrollo Android Google
<http://developer.android.com/>

6. Glosario

EPSG

Corresponde a las siglas de European Petroleum Survey Group (1986-2005). En el 2005 este grupo fue absorbido por la International Association of Oil & Gas Producers (OGP). El grupo estaba formado por especialistas en geodesia, topografía y la cartografía aplicadas en relación a la explotación petrolífera. Este grupo compiló y difundió el conjunto de parámetros geodésicos EPSG, una base de datos ampliamente usada que contiene elipsoides, datums, sistemas de coordenadas, proyecciones cartográficas, etc.

Dialog

Elemento que permite la interacción entre el usuario y un programa informático. Habitualmente se trata de elementos que tienen características bloqueantes y que no permiten realizar otras acciones con la aplicación hasta que se han cerrado. Un ejemplo de esto puede ser los avisos de error que utilizan muchas aplicaciones, o por ejemplo el selector de ficheros en aplicaciones en Java llamado "JFileChooserDialog".

Encriptación geográfica

Método que consiste en la aplicación de una fórmula no lineal a unas coordenadas, de modo que estas queden desplazadas respecto a su posición original. En el caso de China, se aplica este método a toda su cartografía pública de modo que las coordenadas de sus mapas no coinciden con las coordenadas usadas por el sistema GPS internacionalmente.

Geodesia

Ciencia que estudia la forma y dimensiones de la Tierra. Esto incluye la determinación del campo gravitatorio externo de la Tierra y la superficie del fondo oceánico. También incluye la orientación y posición de la tierra en el espacio.

GPS

Sistema de posicionamiento global que utiliza diferentes satélites en órbita sobre la Tierra para averiguar la posición del usuario sobre la superficie del planeta. El sistema fue desarrollado por el Departamento de Defensa de EEUU aunque actualmente es usado en gran parte de los dispositivos móviles y de navegación en todo el mundo.

JSON

Corresponde a las siglas de JavaScript Object Notation. Es un formato ligero para el intercambio de datos usando texto plano. Es un subconjunto de la notación literal de objetos de JavaScript y que no requiere el uso de XML.

Licencia BSD

Se trata de una licencia de software permisiva como puede ser el caso de la OpenSSL o bien la MIT License. Esta licencia permite el uso del código fuente en software no libre.

Licencia LGPLv3

Se trata de una licencia creada por la Free Software Foundation que pretende garantizar la libertad de compartir y modificar el software cubierto por ella. Esta licencia permite el uso del código fuente en software no libre.

Multi-thread

Corresponde a la posibilidad de ejecución de múltiples tareas separadas en diferentes hilos de ejecución. De esta manera es posible llegar a ejecutar código simultáneamente si lo permite la arquitectura que ejecuta el código.

OSM

Abreviatura utilizada comúnmente para referirse a OpenStreetMap.

Proyección

Técnica usada para construir la representación bidimensional de la Tierra. Dado que la Tierra es un elipsoide no existe una forma perfecta de representar la Tierra sobre un plano en dos dimensiones, por lo que existen multitud de proyecciones cada una con sus características.

Proyección Mercator

Proyección creada por Gerardus Mercator en 1569 y que actualmente es usada por multitud de servicios de mapas online en alguna de sus variantes. Google por ejemplo utiliza la proyección llamada Web Mercator, y Yandex utiliza la proyección Mercator elíptica.

OGC

Corresponde a las siglas de Open Geospatial Consortium que es grupo de organizaciones públicas y privadas y que ha creado varias especificaciones usadas ampliamente en servicios de mapas como pueden ser: GML, KML, WFS, WMS, WCS, CSW.

Tesela

Porción de un mapa correspondiente a un fragmento de él. Habitualmente en los servicios web de mapas, toda la cartografía se divide en una cuadrícula. Cada elemento de esa cuadrícula es una tesela y suelen tener un tamaño de 256x256 pixels. La finalidad de esta fragmentación en el mapa es optimizar recursos ya que al estar el mapa dividido en fragmentos, sólo es necesario enviar al usuario aquellos que requiera según el área del mapa que esté visualizando y el nivel de zoom que use.

Thread

Corresponde a la secuencia más pequeña de instrucciones que pueden ser manejadas de forma independiente por el sistema operativo. Se pueden separar ciertos comportamientos de la aplicación en diferentes hilos (Multi-threading) para así no dejar el interfaz gráfico del usuario congelado mientras se hacen operaciones.

Tile

Palabra usada en inglés para denominar una tesela.

Web Mapping

Corresponde al proceso de diseñar, aplicar, generar y visualizar mapas ofrecidos por Sistemas de Información Geográfica (GIS o SIG) a través de Internet.

WMS

Estándar definido por el OGC que produce mapas de datos referenciados especialmente, de forma dinámica a partir de información geográfica. A partir de la información de la solicitud se genera una imagen habitualmente en PNG, GIF, o JPG, aunque también podría ser posible usar en algunos servicios gráficos vectoriales en formato SVG o WebCGM.

XML

Siglas de eXtensible Markup Language. Se trata de un lenguaje de marcas desarrollados por el World Wide Web Consortium (W3C) y que se utiliza para almacenar o intercambiar datos en un formato legible.

XYZ

Arquitectura usada por multitud de servicios de Webmapping en la que el mundo se divide mediante una cuadrícula. A cada elemento de la cuadrícula se le asigna unas coordenadas con números enteros X e Y que indican la tesela dentro de la matriz. Estas coordenadas X, Y son propias de cada nivel de zoom, ya que habitualmente con cada nivel zoom se incrementa el número de teselas que representan el mundo. El número de teselas para la X o la Y del mundo para un nivel dado se suele calcular como 2^{zoom} de modo que para nivel de zoom 0 se tendría que habría $2^0=1$ tesela, para $2^1=2$ y así sucesivamente.