

Planificador de rutas multimodal usando servicios IDE (Bus, Metro y Bici)

Francisco José Peñarrubia¹, José Badía¹

¹ SCOLAB

fjp@scolab.es
jbadia@scolab.es

Resumen

La solución emplea servicios estándares en la medida de lo posible (WMS y WFS). En este trabajo se explica qué ventajas se obtienen al utilizar esos servicios y se analizan cuales son las ventajas e inconvenientes evaluados a la hora de decidir qué servicios IDE son adecuados, y cuales no para este tipo de problemas.

Palabras clave: cálculo de rutas, planificador multimodal, bus, metro, bike sharing, WFS, WMS, servicios web, software libre

1 Introducción

En 2010 se comenzó un proyecto de migración y actualización de la página web de la EMT (Empresa Municipal de Transportes de Valencia) [1].

Entre otras cosas, se quería hacer una remodelación del servicio de cálculo de rutas de autobuses, y se aprovechó un proyecto promovido por el Ayto. de Valencia, de forma que al ser un proyecto del Ayto., el cálculo de rutas debería tener en cuenta además de los autobuses urbanos, los servicios de Metro y ValenBisi (bike sharing).

En el diseño de la nueva web había que tener muy presente la facilidad de uso, así como la rapidez de respuesta y la interconexión con otros

servicios existentes (por ejemplo, información en tiempo real extraída del Sistema de Ayuda a la Explotación (SAE)).

2 Implementación del sistema

Desde el primer momento, se pensó utilizar en la medida de lo posible los estándares OGC [6] que se adaptaran bien a las premisas iniciales, de manera que se pudiera interoperar con otros servicios.

Además, la solución está basada en tecnologías libres, y se han utilizado varias librerías y componentes de software libre tanto en el cliente como en el servidor.

En cuanto a la arquitectura física, se ha utilizado la infraestructura facilitada por Tissat, que mantiene los servidores (virtuales) con las bases de datos empleadas, así como el entorno de desarrollo y el de producción.

En la figura 1 podemos ver un esquema con los servidores:

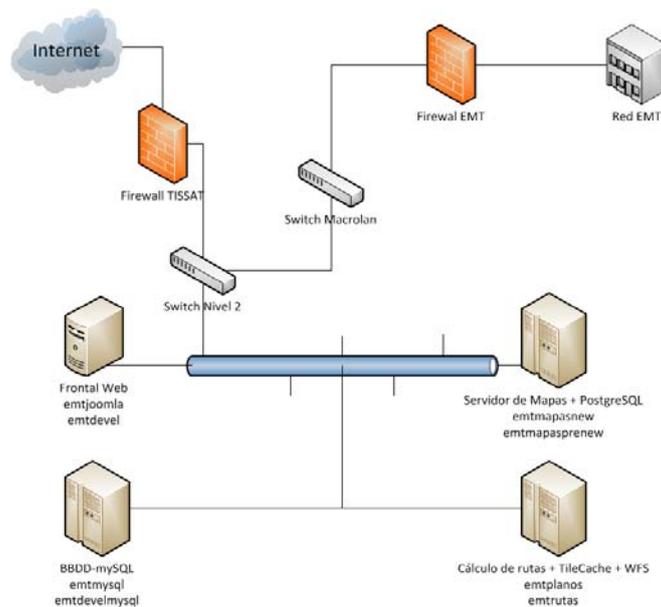


Figura 1: Esquema de Servidores

En la figura se aprecian 2 zonas diferenciadas, con la red de servidores que dan el servicio público principal, y la red interna de EMT.

En el diagrama podemos ver 2 nombres para cada servidor. Es para indicar que cada servidor en realidad son 2 servidores. El primer nombre es el del servidor del entorno de producción, y el segundo, el del entorno de desarrollo.

Todas las modificaciones se prueban primero en el entorno de desarrollo, donde tenemos libertad para hacer cualquier cambio, y una vez validadas, se ponen en marcha en el entorno de producción.

El portal con las páginas web normales está basado en Joomla y MySQL (los dos servidores de la izquierda), y el frontal web redirecciona todo lo referente a las peticiones de mapas y de rutas a los servidores de la derecha.

Los servicios de mapas y de rutas se han implementado gracias a estos dos servidores, que tienen instalados Apache + MapServer [3] y Tomcat + Geoserver [4] + TileCache [5].

La cartografía se guarda en una base de datos PostgreSQL + PostGIS [8], y periódicamente se realizan importaciones desde los servidores de EMT, que a su vez se actualizan con los datos propios de la red de autobuses y los callejeros y demás cartografía suministrada por el Ayuntamiento de Valencia.

Para terminar, el servicio de planificación de rutas está basado en el proyecto OpenTripPlanner [2], con una serie de modificaciones y nuevas funcionalidades que se han desarrollado para adaptarlo a los requerimientos de EMT.

3 Características del sistema

Los rasgos a destacar del planificador son los siguientes:

- Planificador de rutas multimodal.- Se tiene en cuenta la posibilidad de utilizar y mezclar distintos medios de transporte, por ejemplo, Bus y Metro.
- Se calculan varias alternativas de viaje en una única petición.

- Se ofrece información en tiempo real acerca de los tiempos estimados de llegada a una parada, y de las incidencias del sistema (por ejemplo, desvíos).
- Es multilinguaje (Castellano, Valenciano e Inglés)
- Posibilidad de impresión del plano junto con las instrucciones.
- Añade información relevante (horarios, puntos de interés, etc).
- Consultas de carácter espacial (puntos cercanos a una parada)
- Se integra con otros servicios OGC (WMS y WFS).

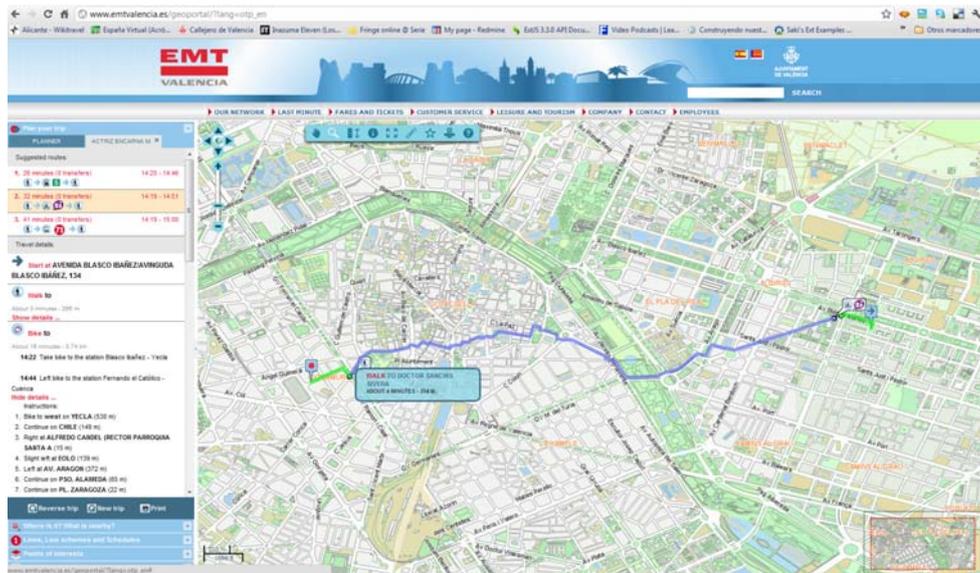


Figura 2: Ejemplo de cálculo de ruta con Metro, ValenBisi y Bus

El núcleo del sistema es el servicio de cálculo de rutas, basado en el proyecto OpenTripPlanner. Este servicio se compone principalmente de 3 subproyectos.

El primero subproyecto se utiliza para generar el grafo a partir de los ficheros GTFS (EMT y Metro), las calles de Valencia y poblaciones limítrofes, la red de calles ciclables y carril bici y los puntos de recogida de ValenBisi.

El segundo subproyecto (planner) se encarga realmente de planificar las rutas. Lee el grafo y recibe las peticiones del cliente.

El tercer subproyecto es la parte del cliente web, que ofrece al usuario un interface amigable que le permite especificar el origen y el destino de su viaje, además de la hora de salida y algunos otros parámetros que puede ajustar. En este subproyecto se usan principalmente 2 librerías JavaScript para conseguir que el cliente sea compatible con la mayoría de los navegadores web (Internet Explorer, Firefox, Chrome, Safari...) y a la vez permitir una buena experiencia de usuario. Las librerías que se han utilizado son ExtJS [9] y OpenLayers [10].

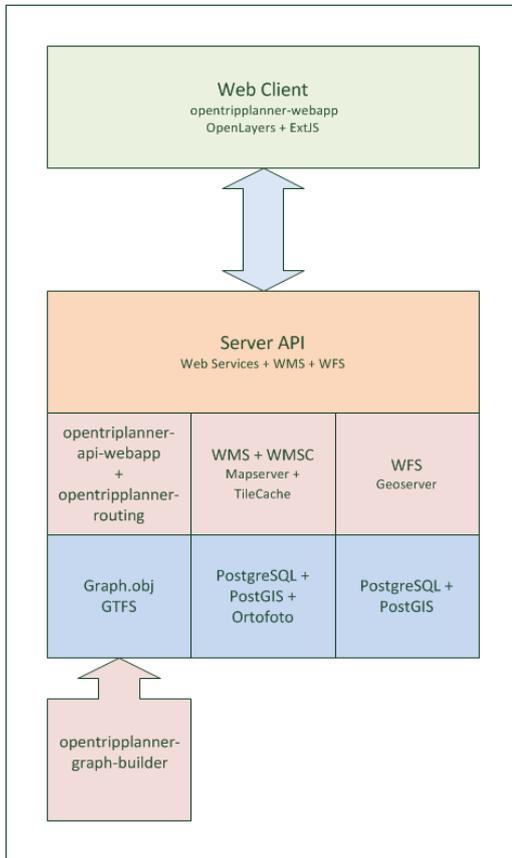


Figura 3: Arquitectura del Sistema

Para preparar la cartografía y los datos, se ha utilizado gvSIG [11], y se han desarrollado nuevas herramientas y extensiones que permitir comprobar la topología e incorporar nuevos datos.

En el proyecto se ha conseguido realizar algo que no se había conseguido todavía, que es la integración de un sistema de Bike Sharing con el autobuses y metro. El problema de bike sharing es diferente al resto de problemas de transporte en los siguientes puntos:

- Bus y metro dan información acerca de los tiempos de paso o la frecuencia mientras que en bike sharing, no hay un tiempo de paso, las bicicletas en principio están disponibles, el usuario llega, toma la bici y se va.
- Bus y metro siempre siguen el mismo camino (hay líneas definidas) pero en bike sharing, el usuario puede usar toda la red de carreteras.
- Hay también otros parámetros menores que hay que tener en cuenta (uso del carril bici cuando se pueda, seguir el sentido de las calles, etc).

Para solucionar el problema de integración del servicio de ValenBisi [12] se crea internamente una red virtual por donde circula la bicicleta. El usuario toma la bici en una estación de ValenBisi, y en ese momento el etiquetado de los nodos se hace sobre la red virtual. De este modo, el tiempo mínimo en cada nodo es distinto si el usuario llega andando o en bici.

El usuario conduce la bici hasta la estación más cercana a su punto de destino, y deja la bici (abandona la red virtual), quedando libre para andar hasta su destino, o hasta una parada de bus o metro que le llevará al destino final.

4 Puesta en marcha del sistema

A la hora de poner en marcha el sistema, hubo que resolver una serie de dificultades.

La primera fue el requisito de conservar la compatibilidad con algunos servicios viejos que estaban en uso. Para cumplir con este requisito, se tomó la decisión de trabajar con sistemas abiertos en la medida de los

posible. En este sentido, la decisión de partir del fichero de Goggle Transit Feed System (GTFS) fue algo muy bueno. El fichero ya se estaba generando para entregarlo a Google Transit, así que parte del trabajo ya estaba hecho. El punto de entrada para generar el grafo es este fichero, y además permite el intercambio de datos con Metro.

Aparecieron también muchos problemas de compatibilidad con navegadores debido a que el intérprete de JavaScript se comporta de manera distinta en los diferentes navegadores (especialmente Internet Explorer). Las librerías ExtJS y OpenLayers han ayudado mucho, pero aún así, se empleó mucho tiempo en el depurado de la parte cliente de JavaScript.

Y finalmente, cuando se puso el sistema en producción, hubo problemas de consumo de memoria y la consiguiente caída del servidor cuando se recibían muchas peticiones. Para evitar esto en el futuro, es muy recomendable diseñar un plan de pruebas adecuado, y estar preparado para aumentar la memoria y/o los procesadores en el servidor.

Para terminar, es una buena política separar los servicios de cálculo de rutas de los servicios de mapas.

En nuestro caso, las peticiones de mapa se sirven con MapServer + TileCache, así que cada vez que un usuario entra al geoportal, se emiten muchas peticiones (una por cada tile de cada capa), y también se lanzan varias peticiones WFS al buscar por calle y número de portal, además de las paradas de autobús que hay en la ventana visible.

Todas estas peticiones las sirve Apache, que por requerimientos de los servicios en php está configurado en modo pre-fork. Esto significa que cada petición se atiende por un nuevo proceso, y esto lleva a un alto consumo de memoria, lo cual provocó que al poner juntos Apache + Tomcat, el servidor se volviera inestable.

Así pues, es un buen consejo a seguir tener separado el servicio WMS (y el WMSC) del servicio WFS, y si se puede, ambos separados del servicio de rutas. Además, si hay algún problema, ayudará a identificar más fácilmente el origen del problema.

5 Conclusiones

Resumiendo, la experiencia ha sido muy positiva, y el cliente (EMT) está muy contento con el resultado. Cada día se hacen alrededor de 3000-4000 cálculos de ruta, y la experiencia de usuario se ha mejorado mucho (velocidad y sencillez de uso).

Las ventajas de este planificador frente a Google Transit son:

- Mejor información y fácil de mantener. Uno de los problemas que EMT quería resolver con este sistema estaba relacionado con el retraso entre la entrega del fichero GTFS y el uso de los nuevos datos por parte de Google. En el momento en el que se planteó el proyecto, se necesitaban cerca de 15 días, así que era imposible reflejar las incidencias en la página web. (Ahora Google está probando en USA un Nuevo formato que permite la comunicación de incidencias en tiempo real, pero todavía no está disponible en Europa).
- Uso de tecnologías abiertas e interoperables (servicios OGC y Software Libre)
- Información en tiempo real acerca de las llegadas a cada parada.
- Los ejes de calle y la información de puntos de interés se mantienen directamente por el servicio de cartografía del Ayuntamiento de Valencia.
- Mejor integración con el resto del portal web corporativo.
- Soporte para información de ValenBisi e integración real con el sistema de planificación multimodal.

Se está preparando una versión para dispositivos móviles para iPhone y Android. Además, en el futuro se va a ampliar el algoritmo de bike sharing para extender el sistema a otros tipos de vehículo compartido (car sharing, bicis eléctricas, etc).

5 Referencias bibliográficas

[1] EMT, 2012. Empresa Municipal de Transportes.
http://www.emtvalencia.es/geoportal/?lang=en_otp

[2] OpenTripPlanner, 2012. OpenTripPlanner.
<https://github.com/openplans/OpenTripPlanner/wiki/>

[3] MapServer, 2012. MapServer. <http://mapserver.org/>

[4] GeoServer, 2012. GeoServer. <http://geoserver.org/>

[5] TileCache, 2012. TileCache. <http://tilecache.org/>

[6] OGC, 2012. Open Geospatial Consortium.
<http://www.opengeospatial.org/>

[7] Refrations, 2012. Postgis. <http://postgis.refrations.net/>

[8] Google, 2012. GTFS format.
<https://developers.google.com/transit/gtfs/reference>

[9] Sencha, 2011. ExtJS library. <http://www.sencha.com/products/extjs3/>

[10] OpenLayers, 2012. OpenLayers. <http://openlayers.org/>

[11] gvSIG, 2012. gvSIG Project. <http://www.gvsig.org>

[12] JCDecaux, 2012. ValenBisi. <http://www.valenbisi.es/>