

Aplicación de Turismo sobre teléfonos móviles.

Miguel Montesinos⁽¹⁾, Alberto Romeu⁽²⁾, Francisco J. Peñarrubia⁽³⁾, José Manuel Fuentes⁽⁴⁾

⁽¹⁾ Prodevelop, Pza. D. Juan de Villarrasa, 14 - 5, 46001 Valencia, mmontesinos@prodevelop.es.

⁽²⁾ Prodevelop, Pza. D. Juan de Villarrasa, 14 - 5, 46001 Valencia, aromeu@prodevelop.es.

⁽³⁾ Software Colaborativo, Valencia, fpenarrubia@scolab.es

⁽⁴⁾ Consejería de Cultura y Turismo de Extremadura, C/ Almendralejo 14 Mérida, 06800 Badajoz, josemanuel.fuentes@juntaextremadura.net

RESUMEN

Se ha desarrollado una aplicación móvil dirigida a turistas que visitan Extremadura, con los objetivos de ofrecerles un rápido y sencillo acceso a mapas y recursos culturales de interés de la Región de Extremadura.

La aplicación funciona sobre teléfonos móviles convencionales que soporten Java. Permite acceder de forma rápida a cartografía de toda la región, así como a puntos de interés turísticos, con capacidades de navegación y de consulta de información.

Se ha creado un sistema de cálculo de rutas para poder organizar desplazamientos entre dos puntos, así como planificar visitas pasando por varios puntos de interés.

El sistema está basado 100% en software libre (licencia GPL), y se encuentra disponible para ser utilizado por los turistas que visitan la región, así como por los desarrolladores interesados en conocer las tecnologías utilizadas..

Palabras clave: *Jornadas, SIG, software libre, Girona, teléfono, móvil, turismo, Extremadura, rutas, mapas.*

INTRODUCCIÓN

En este artículo se describe la creación de una aplicación cliente orientada a turistas que visitan Extremadura, utilizando teléfonos móviles como plataforma de consulta, y componentes 100% open-source.

Este proyecto ha sido desarrollado para la Consejería de Cultura y Turismo de la Junta de Extremadura, y bautizado internamente como SIGATEX-Móvil.



Figura 1: Pantalla de inicio de la aplicación.

Requerimientos iniciales

Los requerimientos establecidos al principio del proyecto incluían las siguientes características generales:

- Funcionar en una amplia gama de teléfonos móviles.
- Poder instalarse directamente desde el teléfono móvil.
- Consultar información de mapas con elementos turísticos, con las capacidades típicas de navegación (zooms, desplazamientos, ...)
- Poder obtener rutas de visita entre dos puntos.
- Poder obtener rutas de visita óptima pasando por N puntos.
- Poder obtener instrucciones para seguir las rutas calculadas.
- Poder visualizar puntos de interés para el turista.
- Poder obtener información de un punto de interés.
- Poder buscar puntos de interés.
- Funcionamiento ágil incluso en teléfonos con capacidades limitadas.
- Compartir servicios existentes en la Consejería de Cultura y Turismo de Extremadura.

DISEÑO DEL SISTEMA

Arquitectura de módulos

El sistema se ha diseñado de forma modular, con componentes desacoplados que comparten información y funcionalidades entre sí. La estructura de módulos del sistema es la siguiente:

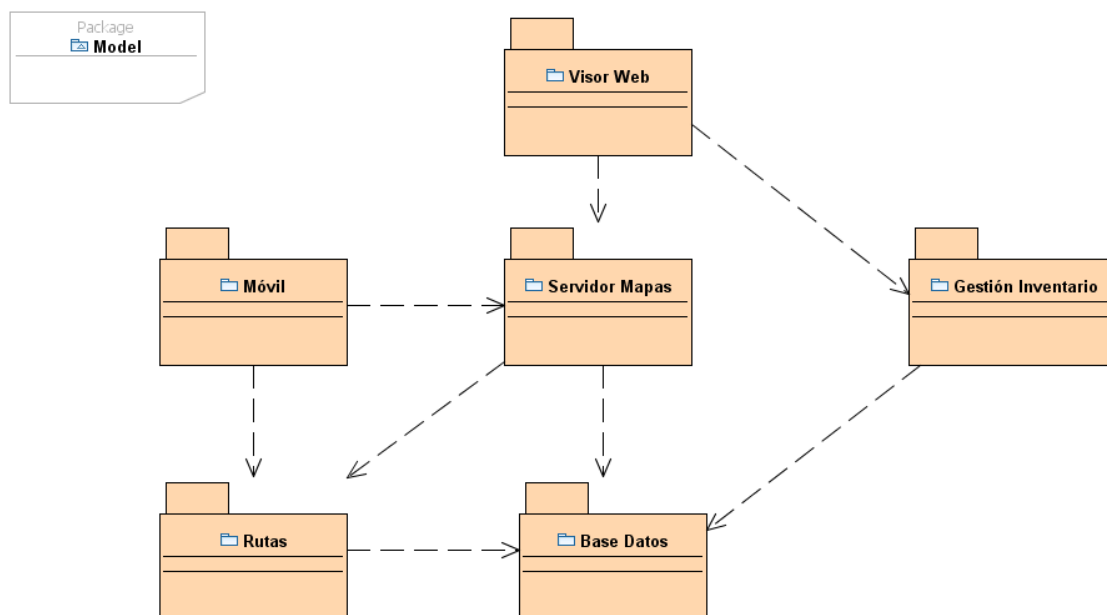


Figura 2: Esquema de módulos de SIGATEX.

En este artículo nos centraremos en la descripción de la aplicación móvil y los servicios de cálculo de rutas necesarios, aunque se hagan algunas referencias a otros subsistemas.

Requisitos del terminal

A la hora de definir los requisitos del terminal para realizar la implementación de la aplicación móvil, se ha decidido utilizar lenguajes estándar que puedan funcionar sobre plataformas libres. Se ha optado por la plataforma Java Micro Edition, que funciona en la gran mayoría de teléfonos móviles del mercado. Los requisitos establecidos para los terminales han sido los siguientes:

Requisitos Mínimos:

- Teléfono móvil con máquina virtual Java Micro Edition ([Java ME¹](http://java.sun.com/javame/technology/index.jsp)).
- Configuración [CLDC²](http://java.sun.com/products/cldc/) 1.1 ([lista de teléfonos con esta configuración³](http://developers.sun.com/mobility/device/pub/device/list.do?filterIds=124) aprox. 260)
- Perfil [MIDP⁴](http://java.sun.com/products/midp/) 2.0 ([lista de teléfonos con este perfil⁵](http://developers.sun.com/mobility/device/pub/device/list.do?filterIds=61) aprox. 360)
- Transmisión de datos GPRS

Compatible con la mayoría de teléfonos existentes en el mercado.

1 <http://java.sun.com/javame/technology/index.jsp>

2 <http://java.sun.com/products/cldc/>

3 <http://developers.sun.com/mobility/device/pub/device/list.do?filterIds=124>

4 <http://java.sun.com/products/midp/>

5 <http://developers.sun.com/mobility/device/pub/device/list.do?filterIds=61>

Requisitos Recomendados Adicionales:

- [JSR-179 Location API¹](#) para soporte local de GPS. ([lista de teléfonos con este API²](#) aprox. 60)
- Transmisión de datos 3G (UMTS) o 3,5G (HSDPA).

Interfaz de usuario

Para la construcción del interfaz de usuario, se han valorado dos frameworks open-source existentes:

- [J2ME Polish³](#). Framework con múltiples utilidades, que ofrece la posibilidad de configurar nuestra aplicación para diferentes dispositivos, personalizando controles. Licenciado como GPL.
- [Lightweight UI Toolkit⁴](#) Framework desarrollado por Sun, para la creación de interfaces de usuario que se comporten igual en todos los dispositivos, de forma análoga a Swing en Java SE. Licenciado como GPL.

Se ha decidido desarrollar el proyecto utilizando LWUIT, que dispone ya de las características necesarias para el proyecto, y presenta una mejor proyección futura.

Visor de mapas

Para poder disponer de un visor de mapas ligero y rápido, se ha decidido implementar una aplicación Java ME, que acceda a un servidor de mapas para mostrar la cartografía, evitando tener que almacenar una enorme cantidad de datos en local.

Utilización de tiles

Para que la sensación del usuario sea de rapidez, se ha diseñado un sistema basado en *tiles* (teselas), que consiste en dividir la pantalla en una malla (o grid) de *tiles* de 256 x 256 píxeles, cada uno de los cuales contiene una imagen solicitada al servidor.

Los *tiles* recibidos se componen en local para configurar la imagen mostrada al usuario (de modo similar a como realizan otros visores Web como Google Maps, OpenLayers, ...). Esto ofrece las siguientes ventajas:

- Se realizan peticiones en paralelo al servidor, mejorándose la respuesta media.
- Cada *tile* se muestra en pantalla conforme se va recibiendo, con lo que la percepción del usuario mejora mucho, la no tener que esperar a recibir toda la información para ver el mapa.
- Se dispone de una cierta capacidad de caché local en forma de buffer alrededor de la zona visualizada, que mejora el desplazamiento (pan) por el mapa.

1 <http://jcp.org/jsr/detail/179.jsp>

2 <http://developers.sun.com/mobility/device/pub/device/list.do?filterIds=125>

3 <http://www.j2mepolish.org/cms/>

4 <https://lwuit.dev.java.net/>

- Se permite acceder a servidores con la información ya “tileada” y “cacheada”, es decir, con las imágenes correspondientes a todos los *tiles* ya generadas y almacenadas en disco. Esto permite acelerar enormemente el acceso remoto a los mapas.

Arquitectura multihilo

Para poder gestionar de forma eficiente una visualización basada en *tiles*, se ha diseñado una arquitectura multihilo que dé soporte a las siguientes necesidades:

- Cancelar peticiones de imágenes que ya no se necesitan. Por ejemplo, al realizar varias veces zoom, nos interesa pedir únicamente las imágenes del último nivel de zoom seleccionado, en vez de esperar a encadenar todas las respuestas.
- Lanzar en paralelo un hilo por cada *tile* del Grid, con lo cual cada *tile* puede ir mostrándose conforme se va recibiendo, sin necesidad a esperar a recibir toda la imagen.

Para la gestión de hilos se ha creado un pool de hilos que permite disponer de un hilo por cada *tile*, siendo reutilizados al necesitar un nuevo *tile*.

Acceso a servidor de tiles

Los *tiles* se solicitan a un servidor de *tiles*. Este servidor (no descrito en el presente artículo) se ha implementado utilizando *TileCache*, que realiza las funciones de “*tileado*” o teselado de un conjunto de niveles de zoom en *tiles* de 256 x 256 píxeles, a las que luego accede la aplicación móvil.

El acceso a los *tiles* se realiza por medio de un servicio WMS-C, sobre HTTP. Se ha probado el cliente de *tiles*, también contra otros servicios WMS-C, como el de cartociudad¹ o metacarta².

Los *tiles* se reciben en el sistema de referencia EPSG:2030. El resto de la cartografía utilizada por la aplicación (geometrías de rutas, puntos de interés) se gestiona en el mismo sistema de referencia, para no obligar a realizar reproyecciones en el lado del teléfono móvil.

Información en local

Para la gestión de puntos de interés, se ha decidido almacenar en local toda la información de puntos de interés turísticos de la región de Extremadura.

Esta información se almacena en el teléfono móvil, por motivos de rendimiento. Esto permite al usuario realizar búsquedas de información en local, así como poder representar a voluntad sobre el mapa puntos de interés turísticos de diferentes categorías sin necesidad de volver a solicitar mapas al servidor, reduciendo al mismo tiempo el tráfico de datos.

1 <http://www.cartociudad.es/wms-c/CARTOCIUDAD/CARTOCIUDAD?>

2 <http://labs.metacarta.com/wms-c/Basic.py?>

CÁLCULO DE RUTAS

Arquitectura del sistema general de cálculo de rutas

El sistema de gestión de rutas se ha diseñado con dos componentes:

- Subsistema móvil de rutas. Se encarga de gestionar la interacción con el usuario en el teléfono móvil.
- Subsistema de cálculo de rutas. Servicio encargado del cálculo de rutas e instrucciones, que da servicio a las peticiones del teléfono móvil, así como a peticiones de otro cliente Web, no descrito en este artículo.

El acceso desde el cliente móvil al servicio de rutas se ha implementado mediante un servicio Web. Para ello se ha seguido la especificación JSR-172¹ (J2ME Web Services Specification), que estandariza el acceso a servicios Web desde aplicaciones móviles J2ME.

La información de las rutas generadas por el servicio de rutas se envían al teléfono móvil con el formato [GeoJSON](#)². Se han desechado otros formatos como GML u otros basados en XML, ya que la sobrecarga de información superflua afectaba al tiempo de transmisión, especialmente en ambientes de cobertura limitada (2G, 2,5G).

GeoJSON es una iniciativa abierta para codificar contenido geográfico en [JSON](#)³. JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero sencillo de entender y de leer y parsear en cualquier lenguaje de programación. GeoJSON ya se está utilizando habitualmente en otros proyectos open-source del mundo Geo, como openlayers, MapFish, GeoServer, FeatureServer, MapBender, PostGIS,...

Gestión de rutas en el móvil

La gestión de rutas tiene como origen el interfaz gráfico. Desde la aplicación móvil, el usuario puede configurar la ruta que desea calcular. El usuario dispone de las siguientes opciones:

- Establecer puntos de origen / destino de la ruta. Al establecer ambos, automáticamente se lanza la petición de cálculo de ruta.



Figura 3: Pantalla mostrando ruta entre dos puntos.

1 <http://jcp.org/en/jsr/detail?id=172>

2 <http://geojson.org/>

3 <http://json.org/>

- Modificar punto de origen / destino, recalculándose automáticamente la ruta.
- Utilizar el cursor para definir puntos de ruta, o establecer un punto de interés turístico como origen, destino o punto de paso de una ruta.
- Indicar algunas características de la ruta, como por ejemplo si es a pie o en coche.
- Definir varios puntos de paso de ruta. Esta opción está pensada para el caso en el que el turista desea visitar N puntos de interés. En este caso el sistema le calcula la ruta óptima para minimizar el recorrido pasando por todos los puntos.
- Obtener indicaciones o instrucciones para seguir la ruta calculada. Por cuestiones de espacio en pantalla se ha implementado un scroll automático horizontal del texto de cada instrucción.

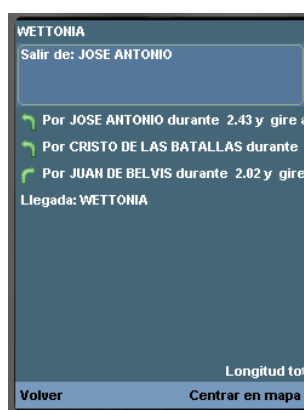


Figura 4: Pantalla mostrando indicaciones de ruta.

- Centrar en el mapa un punto de la ruta.
- Modificar puntos de la ruta o anularla.

Una vez que se ha realizado la petición de una ruta, el resultado se obtiene en formato GeoJSON, como se ha descrito anteriormente. La información en GeoJSON, se parsea y se guarda en un modelo de objetos en memoria. A partir de este momento, esta información es accesible en local, con lo que en cada zoom o repintado, se dibuja la geometría de la ruta sin necesidad de acceder al servidor.

La información de instrucciones de ruta se recibe en un formato XML definido para este proyecto. Este XML se parsea en local para ser mostrado al usuario de manera entendible.

Servicios de rutas en el servidor

En el lado del servidor se ha creado un servicio Web RESTful, que recibe peticiones procedentes del cliente móvil o de un cliente Web, y devuelve la geometría de la ruta, así como las instrucciones a seguir.

El servicio de rutas se ha construido tomando como base las capacidades de gestión de redes del proyecto [gvsig](http://www.gvsig.gva.es/)¹. gvsig es una herramienta orientada al manejo de información geográfica. Se trata de una herramienta avanzada de escritorio open-

1 <http://www.gvsig.gva.es/>

source, licenciada bajo GNU/GPL, que ofrece un completo conjunto de capacidades de acceso a información geográfica, así como de análisis espacial. Entre las características de análisis, posee capacidades de gestión de redes, con cálculo de caminos óptimos.



Figura 5: Logo de gvSIG, reutilizado para el cálculo de rutas.

El código de gvSIG se ha adaptado para que funcione en un componente de servidor [Java EE²](http://java.sun.com/javase/). Esto ha permitido configurar el componente de rutas como una aplicación Java EE (J2EE), que se despliega en un contenedor de aplicaciones J2EE, en este caso Tomcat. Este componente es accesible a través de un servicio Web estándar.

La adaptación a Java EE ha implicado, entre otros aspectos, la posibilidad de calcular y servir en paralelo varias rutas de forma simultánea.

El cálculo de rutas utiliza dos algoritmos:

- Camino óptimo entre dos puntos, utilizando el algoritmo A*.
- Camino óptimo pasando por N puntos, también conocido como problema del viajante o TSP (Travelling Salesman Problem).

PUNTOS DE INTERÉS

La gestión de puntos de interés se realiza en el lado del teléfono, para optimizar la velocidad de uso de la aplicación.

Los puntos de interés están almacenados en la memoria del teléfono, y se accede a ellos en local. A partir de un cierto nivel de zoom, los puntos de interés se muestran en pantalla.

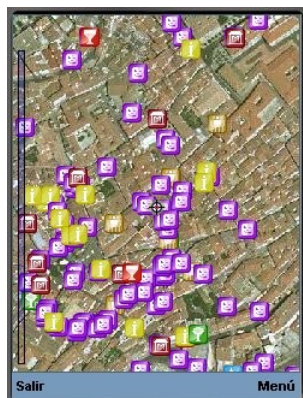


Figura 5: Pantalla mostrando puntos de interés.

² <http://java.sun.com/javase/>

El número de puntos de interés disponibles es bastante elevado (cerca de 20.000). Para poder representar estos puntos de forma ágil y disponer de capacidades de consulta con tiempos de respuesta reducidos, se ha implementado un índice espacial.

Las funcionalidades de las que dispone el turista son principalmente las siguientes:

- Visualizar puntos de interés en el mapa, con simbología según categoría..
- Consultar información acerca de un punto de interés. Para ello, se utiliza un cursor en el centro de la pantalla, y se buscan los puntos cercanos al cursor, indicándole al usuario la distancia aproximada.

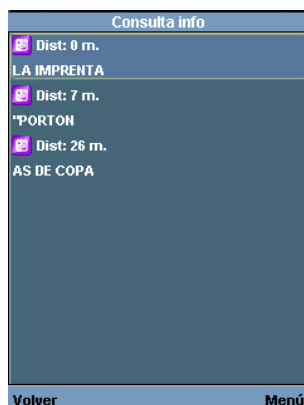


Figura 6: Pantalla mostrando consulta de puntos de interés.

- Realizar algunas acciones sobre un punto de interés, como centrarlo en pantalla, utilizarlo como punto de ruta, etc.
- Buscar puntos de interés, seleccionando una categoría e introduciendo un texto.

LOCALIZACIÓN

La aplicación posee también la posibilidad de utilizar posicionamiento GPS, si el terminal dispone de esta capacidad.¹

Para ello se ha implementado soporte al estándar [JSR-179](#) (Location API for J2ME). Los teléfonos que implementan esta especificación, ofrecen un API estándar, que es el que se ha utilizado en la aplicación cliente.

La funcionalidad que se le ofrece al usuario es la de poder localizar su ubicación a demanda, y centrar la posición en pantalla en base a esta localización.

Dado que la cartografía manejada por la aplicación se encuentra en el sistema de referencia EPSG:2030 y la entregada por los GPS se encuentran en EPSG:4326, ha sido necesario implementar la transformación entre ambos sistemas.

Para esta reproyección se ha reutilizado código de [gvSIG Mobile²](#), añadiendo el soporte de librerías matemáticas no disponibles en la configuración CLDC 1.1/MIDP 2.0.

¹ <http://jcp.org/jsr/detail/179.jsp>

² <http://www.gvsig.gva.es/index.php?id=gvsig-mobile&L=2%2Findex.php%3Fid%3D&K=1&L=0>

CONCLUSIONES

La aplicación de gestión de recursos turísticos sobre teléfonos móviles ha servido para demostrar la posibilidad de utilizar estos terminales como plataforma de desarrollo de aplicaciones Java, utilizando exclusivamente software open-source.

En lo que constituye una prueba de la sinergia que se produce en la adopción de software libre, se ha utilizado, entre otros, código de los siguientes proyectos open-source:

- gvSIG [1].
- gvSIG Mobile [2]
- LightWeight UI Toolkit [3].
- JSON in Java [4]
- MapFish [5].

Este proyecto será publicado en la forja de la Junta de Extremadura, bajo licencia GNU/GPL 2.0.

AGRADECIMIENTOS

Agradecemos a la Consejería de Cultura y Turismo de Extremadura la posibilidad de desarrollar este proyecto y su liderazgo en él, así como su compromiso con la adopción de software libre.

REFERENCIAS

- ◆ [1] gvSIG: <http://www.gvsig.gva.es/>
- ◆ [2] gvSIG Mobile: <http://www.gvsig.gva.es/index.php?id=gvsig-mobile&L=2%2Findex.php%3Fid%3D&K=1&L=0>
- ◆ [3] LightWeight UI Toolkit: <https://lwuit.dev.java.net/>
- ◆ [4] GeoJSON in Java: <http://www.json.org/java/>
- ◆ [5] MapFish: <http://www.mapfish.org/>