

Francisco Javier Martín Otero
Programador senior en Batkiwi; Ganador del VI
Concurso Universitario de Software Libre en la
categoría "Mejor proyecto comunitario"

<fjavier.mo@gmail.com>

GeoTask: Servicios basados en localización para sistemas Android

1. Antecedentes técnicos: Recursos Android basados en geolocalización

Determinar en todo el mundo la posición de un objeto, una persona o un vehículo con la mayor precisión posible, es algo que lleva haciéndose desde hace años gracias a GPS (*Global Positioning System*, Sistema de Posicionamiento Global).

Actualmente, dentro del mercado de la telefonía móvil la tendencia es la de integrar, por parte de los fabricantes, la tecnología GPS dentro de sus dispositivos. Esto ha hecho surgir todo un ecosistema de software para este tipo de dispositivos, así como nuevos modelos de negocio que van desde el uso del terminal móvil para la navegación tradicional punto-a-punto hasta la prestación de los llamados Servicios Basados en la Localización (LBS).

Dentro del mercado de los *smartphones*, podemos observar que el sistema operativo más usado es Android con un 72,4% de cuota de mercado a escala mundial, por delante de iOS que posee una cuota aproximada del 13,9%. En tercer lugar se sitúa Blackberry OS con una cuota de mercado del 5,3%.

Android es un sistema operativo basado en el núcleo Linux diseñado originalmente para dispositivos móviles, tales como *smartphones*, pero que posteriormente expandió su desarrollo para soportar otros dispositivos tales como tablets, reproductores mp3, netbooks, PC, televisores, lectores de e-books e incluso, se han llegado a ver en la última edición de CES¹, microondas y lavadoras.

Fue desarrollado inicialmente por Android Inc., una empresa comprada por Google en 2005. Es el principal producto de la *Open Handset Alliance*, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio².

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos [1][2][3].

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un *framework* Java de aplicaciones orienta-

Resumen: Los Servicios Basados en Localización (LBS) buscan ofrecer un servicio personalizado a los usuarios basándose en la mayoría de situaciones en información de ubicación geográfica de éstos. Para su operación utilizan tecnologías de Sistemas de Información Geográfica, algunas tecnologías de posicionamiento, bien sea del lado del cliente (por ej. GPS) o del lado del servidor (por ej. servicio de posicionamiento suministrado por el operador de la red), y tecnologías de comunicación de redes para transmitir información hacia una aplicación LBS que pueda procesar y responder la solicitud. Geotask es una aplicación basada en este tipo de servicios que explota la posibilidad de geolocalización de los dispositivos con sistemas operativos basados en Android para poder ejecutar tareas de forma totalmente automática.

Palabras clave: Android, automatizar, geolocalización, software libre, tarea.

Autor

Francisco Javier Martín Otero es Ingeniero Informático y programador senior en Batkiwi. Entusiasta del software libre, ha sido miembro del grupo de usuarios y usuarias de GNU/Linux de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla (SUGUS). En 2011 comenzó su interés por la programación en dispositivos móviles, especialmente en Android. Desde entonces ha estado desarrollando aplicaciones como MedsAlarm. Pero sin duda, su aplicación más exitosa ha sido GeoTask, proyecto galardonado con el Premio al mejor proyecto comunitario en la fase final del VI Concurso Universitario de Software Libre.

das a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (*surface manager*), un *framework* OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C.

El *framework* de Android permite una sencilla reutilización de componentes y comunicación entre aplicaciones, siempre sujetas a ciertas medidas de seguridad, que facilita, por ejemplo, la actualización o sustitución de componentes por parte del usuario, resultando un sencillo y efectivo método para utilizar novedades o introducir mejoras en el software.

Otra peculiaridad de Android es el ciclo de vida de las aplicaciones, ya que no se trata únicamente de abrir y cerrar a criterio del usuario, sino que éstas, una vez iniciadas, permanecen cargadas en memoria siempre que se disponga de recursos para ello. En caso contrario, el propio sistema operativo se encargará de destruirlas definitivamente. Dicho ciclo de vida se rige por las llamadas a los métodos *onCreate*, *onStart*, *onResume*, *onPause*, *onStop*, *onDestroy* y *onRestart*.

El principal atractivo que posee Android es que su código ha sido liberado, lo que lo

convierte en un sistema operativo totalmente libre para que un desarrollador no solo pueda ver su código sino también mejorarlo. A través de esas mejoras puede publicar el nuevo código y con él, ayudar a mejorar el sistema operativo para futuras versiones sin depender de fabricantes u operadoras para que liberen o no dicha mejora.

Del mismo modo, al ser código abierto se garantiza que, en caso de producirse una incidencia o error, éste sea detectado y reparado con mayor presteza al no existir ninguna traba legal para indagar en su interior ni depender de nadie para pedir autorización a su cambio.

Esto ha propiciado una gran acogida por parte de desarrolladores independientes, dando lugar a una gran comunidad internacional que, en apoyo a la propia documentación oficial disponible, abre puertas a casi cualquier persona que quiera iniciarse en la programación de dispositivos móviles.

Éste ha sido mi motivación fundamental para elegir Android en vez de cualquier otro sistema operativo. También ha influido la importante cuota de mercado que posee Android en todo el mundo y la cantidad de móviles a precios asequibles que se pueden encontrar en el mercado.

2. ¿Qué es GeoTask?

GeoTask es una aplicación para móviles con sistema operativo Android que nos permitirá

“ GeoTask se cimenta en el perfil como concepto fundamental. El perfil nos indica qué tareas queremos realizar en un área de proximidad determinada ”

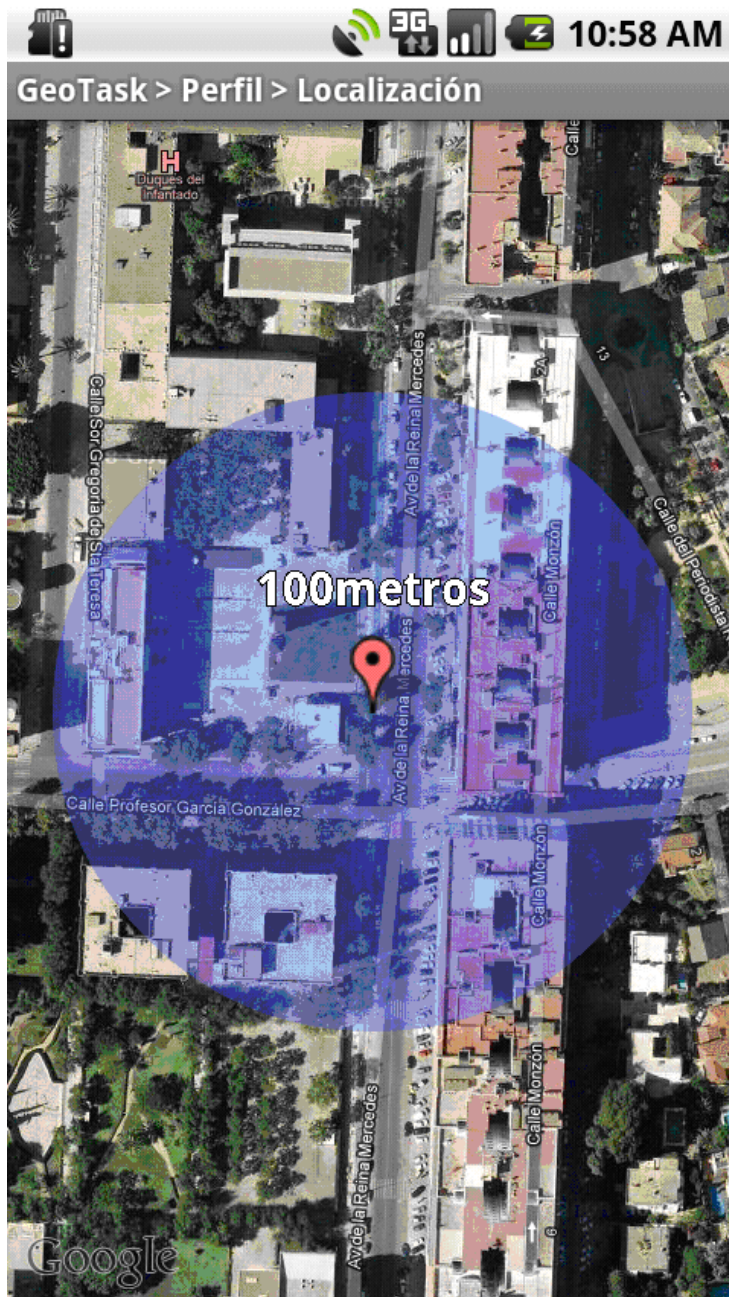


Figura 1. Interfaz de GeoTask para mostrar una ubicación relacionada con un perfil.

realizar automáticamente ciertas tareas, sin la intervención del usuario, al cumplirse una determinada condición [4][5][6]. En nuestro caso, dicha condición será que la aplicación detecte que nos encontramos en una determinada zona (ver **figura 1**).

GeoTask se cimenta en el perfil como concepto fundamental. El perfil nos indica qué tareas

queremos realizar en un área de proximidad determinada.

Un perfil está compuesto por:

- Nombre: El nombre que pongamos, será el que aparezca en la pantalla principal.
- Área de proximidad: El área de proximidad no es más que el área geográfica que se encuentra a una distancia del punto que ele-

gimos como dirección, menor o igual al radio que hemos seleccionado. Se compone de un punto central (dirección) y el radio.

- Tareas que queremos realizar.

Estas tareas pueden ser :

- Recibir recordatorios.
- Enviar SMS.
- Realizar llamadas.
- Realizar una lista negra de nuestros contactos.
- Realizar una lista blanca de nuestros contactos.
- *Enviar tweets.*
- Actualizar nuestro estado en Facebook.

Para trabajar con GeoTask, el primer paso que debemos hacer es crear un perfil. Para ello debemos proporcionarle un nombre y elegir el área de proximidad que queremos que tengan las tareas a realizar.

El último paso es elegir las distintas tareas que queremos que se hagan. Algunas tareas podrán ser seleccionadas una sola vez, como realizar llamadas y crear una lista negra o blanca de nuestros contactos. Las demás podrán ser seleccionadas las veces que queramos (ver **figura 2**).

Cada perfil podrá ser activado/desactivado desde la pantalla principal.

Para que nuestra aplicación funcione correctamente y se realicen automáticamente las tareas, debemos habilitar un servicio, que se encargará de actualizar la ubicación en tiempo real del teléfono y que podremos parar en el momento que consideremos oportuno.

3. Desarrollo de GeoTask

Hemos distribuido las funcionalidades previstas de la forma más óptima posible, proporcionando un entendimiento fácil e intuitivo, así como un mantenimiento más sencillo.

Hemos organizado el proyecto en 7 paquetes. Un paquete en Java es un contenedor de clases que permite agrupar las distintas partes de un programa cuya funcionalidad tienen elementos comunes.

Nuestra aplicación contará con una clase principal llamada *Main* y que será la clase a la que accedemos cuando arrancamos la aplicación. En ella es donde se mostrarán los perfiles que ya hayamos creado, con la posibilidad

de editarlos, eliminarlos o activarlos/desactivarlos, también podremos añadir nuevos perfiles, ver las cuentas de las redes sociales en las que nuestra aplicación tiene permiso para usar (clase *Accounts*) y habilitar/deshabilitar el servicio encargado de obtener nuestra localización.

Esta última clase extiende la clase *Activity*. Una *Activity* representa el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una actividad como el elemento análogo a una ventana en cualquier otro lenguaje visual.

Otra clase importante es *Perfil*, que es la encargada de crear o editar los perfiles y de crear la *Activity Map*, que tiene asignadas 3 tareas básicas: construcción del mapa, designación de área de proximidad donde el centro es un localizador de posición en el que podremos aumentar/disminuir el radio e interactuar con el localizador de posición anterior y configurar el modo de vista del mapa. El modo de vista nos ofrecerá la posibilidad de elegir entre *Mapy Satellite*. Estas clases se encuentran en el paquete `com.pfc.geotask`.

En el paquete `com.pfc.database`, se encuentran todas las tablas necesarias para el correcto funcionamiento de nuestra aplicación. La clase principal de este paquete es la clase *DBAdapter*, que es la encargada de crear todas las bases de datos y triggers que necesita la aplicación.

En el paquete `com.pfc.task` se encuentran todas las *Activities* correspondientes a las tareas que puede ejecutar nuestra aplicación.

Todas estas clases cuentan con la misma estructura, a excepción de 3 de ellas:

- La clase *BlackWhitelist*, que se encarga tanto de crear una lista negra como una lista blanca de nuestros contactos. Tiene la particularidad de que podremos cambiar de tarea desde la misma aplicación a través del menú.
- Las clases *Twitter* y *TFacebook*, que se encargan de las tareas relacionadas con las redes sociales Twitter y Facebook, cuentan con la particularidad de que podremos autorizar el uso de nuevas cuentas a través del menú. Para ello cada una de ellas crea una *Activity AutorizaTwitter* y *AutorizaFacebook*, respectivamente, que son las encargadas de realizar todo el proceso de autorización necesario para ello.

Estas dos *Activities* encargadas de dar permisos a nuestra aplicación, se encuentran en el paquete `com.pfc.geotask.authorize`.

La clase fundamental de nuestra aplicación se encuentra en el paquete `com.pfc.geotask.service`. Es la clase *GeoService* y su principal funcionalidad es asignar el listener que se encargará de actualizar perío-

dicamente la posición del usuario y lanzar las alertas de proximidad de todos los perfiles que se encuentren activos.

Esta clase extiende de *Service*. Un *Service* es un componente sin interfaz gráfica que se ejecuta en segundo plano. En concepto, es exactamente igual a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (*Activities*) si se necesita en algún momento la interacción con del usuario.

Cuando entremos dentro del área de proximidad de algún perfil, entrarán en juego las clases que se encuentran en el paquete `com.pfc.broadcast`.

Cuando entramos dentro de un área de proximidad se activa la clase *AreaProximityReceiver*. Su función principal es la de detectar si hemos salido del área de proximidad del perfil o estamos entrando.

Si hemos salido deberemos activar la clase *RemoveBWLlistReceiver* para eliminar del sistema las listas blanca o negra que pudiera tener el perfil. Si estamos entrando, deberemos activar todas las clases que lanzan las tareas automáticamente que estén asociadas al perfil.

Esta clase extiende de *BroadcastReceiver*. Un *BroadcastReceiver* es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: "Batería baja", "Sms recibido", "Tarjeta SD insertada") o por otras aplicaciones.

Cualquier aplicación puede generar mensajes

(*intents*, en terminología Android) *broadcast*, es decir no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo).

Un *intent* es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se puede entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un *intent* se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc.

Nos falta hablar del paquete `com.pfc.geotask.utils`. A este paquete pertenecen las clases Java, la mayoría son estáticas, que nos ofrecen ciertas funcionalidades necesarias para varias clases, por ejemplo: saber si un número es un número de teléfono válido, etc.

4. Objetivos del proyecto

El principal objetivo que se ha perseguido a lo largo del desarrollo de este proyecto ha sido crear una aplicación que, dada una condición, ejecute de forma automática y sin la intervención del usuario de una serie de tareas que hayamos elegido previamente.

Otros objetivos también importantes han sido:

a) Asegurar que el sistema sea fácil de mantener, extender y escalar. Para ello tomamos una serie de medidas como, por ejemplo, hacer independiente nuestro código a la base de datos usada para guardar los datos necesarios para el correcto funcionamiento de nuestra aplicación, estructurar el código para conseguir que añadir una nueva tarea sea lo más sencillo posible, etc.



Figura 2. Creación de un perfil en GeoTask

“ En la actualidad Geotask cuenta con una versión beta aunque estamos trabajando en una nueva versión para que pueda ser utilizada en Android 4.0, mejorando su interfaz y rendimiento ”

b) Investigación del funcionamiento del sistema operativo Android, las posibilidades que nos ofrece y su gestión por parte de las APIs disponibles y el manejo de aplicaciones. Centrandonos en las tareas de desarrollo, es indispensable el conocimiento de la herramienta completa ofrecida para tal fin, así como su integración y uso en entornos de desarrollo ya existentes.

c) Estudiar el entorno de desarrollo de Android. Al lanzarse bajo una licencia de software libre, el SDK completo está disponible para cualquier desarrollador que desee descargarlo. Éste incluye numerosas ayudas para comenzar a crear aplicaciones en Android, desde la API completa con todas las clases y paquetes, hasta herramientas de programación y un completo emulador para poder realizar pruebas. Todos estos elementos han de ser estudiados y explicados. Dispondremos de una completa guía de instalación de su SDK (kit de desarrollo de software) y explicaremos como instalar el plugin ADT en Eclipse o en el entorno de desarrollo que estemos utilizando, necesario para empezar a crear aplicaciones propias en Android.

d) Desarrollar una aplicación para Android totalmente funcional que sea muy intuitiva y fácil de utilizar. Para ello la aplicación está siendo probada por un conjunto de personas. Así conseguiremos comprobar si la aplicación cumple con este criterio y, en caso negativo, podremos mejorarla siguiendo las pautas que nos sean marcadas.

e) Demostrarlo sencillo que resulta en Android realizar tareas de forma automática. En este caso nos basamos en nuestra localización, pero se podría realizar igualmente basándonos en la hora, en el nivel de batería, etc.

5. Estado del proyecto

En la actualidad Geotask cuenta con una versión beta [7] aunque estamos trabajando en una nueva versión [8] para que pueda ser utilizada en Android 4.0, mejorando su interfaz y rendimiento.

Esta nueva versión será reescrita desde cero para evitar fallos de diseño de la primera versión.

6. Llamamiento a la colaboración en el proyecto GeoTask

GeoTask es una aplicación de software libre. Por lo tanto su código fuente está disponible para cualquier persona y cualquier colaboración es bienvenida.

Hay muchos aspectos donde es posible colaborar con el proyecto:

a) Código: Suele ser la opción más usual para colaborar con un proyecto de software libre. GeoTask está siendo reescrito para soportar Android 4.0 y mejorar su eficiencia. Es una tecnología bastante atractiva y muy fácil de aprender.

b) Interfaz de usuario: Uno de los elementos más importantes de una aplicación es la interfaz de usuario y su usabilidad. Está demostrado que una aplicación con una interfaz amigable y atractiva a la vista tiene más opciones de destacar sobre el resto aún cuando no sea la mejor. Es un aspecto que en GeoTask está bastante olvidado por lo que necesitamos la ayuda colaboradores expertos en este ámbito para darle un toque de calidad a GeoTask, al igual que diseñadores gráficos para la creación de las diferentes imágenes y elementos gráficos que se usarán en la aplicación.

c) Pruebas: El principal problema que nos encontramos los programadores de Android es la cantidad de dispositivos distintos que existen, muchos de ellos con distintas resoluciones de pantalla. Sería de gran ayuda para GeoTask obtener colaboradores en este aspecto, para que la aplicación sea probada por el mayor número de dispositivos diferentes posibles. Con esto, podríamos también buscar y reportar errores para que sean solucionados a la mayor brevedad posible.

d) Otros: Cualquier otra aportación al proyecto será bienvenida, por ejemplo, aportes en la documentación, aportes de ideas para futuras versiones, ayudas en la traducción de la aplicación a cualquier idioma, ayudas en la difusión, etc.

Muchas gracias de antemano a todos los interesados, a quienes animo a ponerse en contacto conmigo a través del correo electrónico <a.geotask@gmail.com>.

Referencias

- [1] **Android.** Web oficial para desarrolladores, <<http://developer.android.com>>.
- [2] **Sgoliver.** Blog con una serie de artículos sobre programación Android, <<http://www.sgoliver.net/blog>>.
- [3] **Google.** Grupo de desarrolladores Android, <<https://groups.google.com/forum/#!forum/desarrolladores-android>>.
- [4] **GeoTask.** Web oficial, <<http://geotask.wordpress.com>>.
- [5] **Twitter.** Presentación oficial, <<http://twitter.com/AppGeoTask>>.
- [6] **Facebook.** Presentación oficial, <<https://www.facebook.com/AppGeoTask>>.
- [7] **Rediris.** Forja del proyecto, <<https://forja.rediris.es/projects/geotask>>.
- [8] **GitHub.** Forja de la nueva versión, <<https://github.com/fjaviermo/GeoTask>>.

Notas

- ¹ *International Consumer Electronics Show, Las Vegas 2013.*
- ² <<http://www.openhandsetalliance.com>>.