

Desarrollo de aplicación para dispositivo móvil con software libre para el uso en zonas de alud

M. Pérez, N. Querol, J. Marturià⁽¹⁾

⁽¹⁾ Institut Geològic de Catalunya (IGC), Carrer Balmaes 209, Barcelona, jmarturia@igc.cat

RESUMEN

*Identificar las diferentes zonas de aludes cartografiadas en campo no siempre es fácil, debido a que no suelen estar bien delimitadas por la orografía; la cobertura nival dificulta aún más distinguirlas de forma correcta y puede haber más de una en la misma pendiente. Por esta razón, desde el Institut Geològic de Catalunya (IGC) se ha decidido realizar una **aplicación para dispositivos móviles** que ayude en el trabajo de campo de los técnicos encargados de hacer un seguimiento de estas zonas. Se utilizan dos sistemas complementarios para facilitar la localización de las zonas de alud en el campo. En primer lugar se dispone de un mapa donde se identifican las diferentes zonas de alud junto con la posición actualizada del usuario. Debido a la mala cobertura en las zonas de montaña, utiliza mapas offline (cargados en el dispositivo) y para posicionarse sólo utiliza el **GPS** del dispositivo puesto que hay poca cobertura de redes para ubicarse. En segundo lugar usa la tecnología de realidad aumentada para visualizar la localización de estas áreas superpuestas en la imagen captada por la cámara del dispositivo, así se logra identificar cada zona de alud en la propia pendiente de la montaña. Para realizar esta aplicación se ha utilizado software libre. La aplicación está desarrollada para el sistema operativo de Google para dispositivos móviles **Android**. Para guardar los datos en el dispositivo se utiliza la propia Api de Android que permite el uso de la base de datos **SQLite**. Se ha recurrido a la API para dispositivos móviles **Osmdroid** para representar el mapa. El uso de ésta facilita también el uso del mapa offline con el "**Mobil Atlas Creator**". Finalmente, también se ha utilizado la API de software libre llamada **Appunta** para implementar la **realidad aumentada** en la aplicación.*

Palabras clave: Aplicación móvil, alud, SIG, Software Libre, Osmdroid, Realidad Aumentada, RA, Appunta, Mobil Atlas Creator, GPS

ABSTRACT

Fieldwork identification of different avalanche zones is not always easy. These are not always well defined on the terrain, the snow makes it more difficult to distinguish correctly and could be several on the same slope. For this reason, the Institut Geològic de Catalunya (IGC) has decided to make an application for mobile devices to assist to the technicians in the field.

We use two complementary systems to facilitate avalanche location in fieldwork activities. The first is a map which symbolizes the different avalanche areas along with the updated position of the user. Due to poor coverage in mountain areas, the device uses offline maps (loaded on the device) and only uses the GPS position since there is little network coverage. Secondly it uses augmented reality technology to display these overlapping avalanche areas in the image captured by the camera device.

In the development of this application was used open source software. The application is developed for the operating system Google Android for mobile devices. The data on the device is stored using Android APIs that allow the use of SQLite. We have used the API for mobile devices called osmdroid to represent the map. Using this API also facilitates the use of the map offline with the "Mobile Atlas Creator". Finally, we also used the open source API called Appunta to implement augmented reality in the application.

Key words: *mobile app, avalanche, GIS, open source, Android, Osmdroid, augmented reality, Appunta, Mobil Atlas Creator, GPS, SQLite*

INTRODUCCIÓN

En los últimos años los dispositivos móviles, como pueden ser los *smartphones* y *tablets*, se ha convertido en un objeto habitual con grandes capacidades de procesamiento. La dotación de estos aparatos con conectividad, cámara, sistemas de posicionamiento, brújula digital, acelerómetros, etc. han convertido a estos dispositivos en unas potentes herramientas con gran amplitud de funciones y usos.

Aprovechando su potencialidad, desde el Institut Geològic de Catalunya (IGC) [1], mediante el convenio de prácticas de empresa y como trabajo de final del Màster en Tecnologies de la Informació Geogràfica (MTIG) [2] en su 14ª edición (LIGIT, Departamento de Geografía, Universitat Autònoma de Barcelona), se ha decidido diseñar y desarrollar un prototipo de aplicación para móvil que permita mejorar la localización e identificación de las zonas de alud en campo. Partiendo de este planteamiento, se han fijado las siguientes metas:

- Facilitar el posicionamiento de las zonas de alud sobre cartografía.
- Ayudar a la identificación de las zonas de alud con realidad aumentada.
- Permitir la búsqueda de zonas de alud por proximidad.
- Trabajar *offline* puesto que en las zonas de montaña la cobertura 3G no está garantizada.

La aplicación, llamada **IdAllau**, se desarrolla para dispositivos con sistema operativo Android. Ésta utiliza el sistema de posicionamiento para situarse en el territorio y así poder mostrar las zonas de alud en realidad aumentada; es decir, sobre la imagen real captada a través de la cámara. Debido a la mala cobertura en zonas de montaña, sus funciones principales han de funcionar sin conectividad a internet (3G o WiFi), y por tanto todos los datos han de estar disponibles en el dispositivo previamente.

Teniendo en cuenta las funcionalidades planteadas, los usuarios potenciales de la aplicación serán el personal de IGC y de otras instituciones que participan en la recogida de información en el campo.

ESTADO DE LA TECNOLOGIA

Sistema operativo (SO)

Los principales SO usados en dispositivos móviles son dos: iOS de Apple y Android de Google. Crear aplicaciones para el primero de ellos implica el pago de una cuota para el iPhone Developer Program. También se ha de tener en cuenta que el lenguaje de programación es propio para iOS. Por otro lado, el pago de esa cuota garantiza una revisión de la aplicación por parte de Apple y que pueda utilizarse en sus productos.

En caso de Android, su rápida expansión y disponibilidad en multitud de fabricantes, así como unas tecnologías abiertas (con coste nulo para el desarrollo de aplicaciones), le han otorgado una gran comunidad de desarrolladores que comparten sus conocimientos. Por el contrario, ésta gran distribución provoca que las

aplicaciones no siempre funcionen en todos los dispositivos, puesto que la diversidad de éstos hace imposible un estándar de dispositivo para el programador.

Teniendo en cuenta las características de los dos SO, se escoge **Android** para el desarrollo de la aplicación debido a su gran expansión (hoy en día el SO más presente en el mercado [3]), los costes nulos de desarrollo, una documentación extensa, etc. Así pues, el resto de estado de tecnología se limitará a los recursos disponibles para Android.

Teniendo en cuenta los datos que facilita Google del uso de las versiones de su sistema operativo [4], se decidió en su día (lo datos han variado des del diseño al final de la programación) que la aplicación fuera compatible des de la versión 2.2 Froyo (API 8) hasta la actual 4.2 Jelly Bean (API 17), llegando a más del 97% de los dispositivos con Android.

Almacenamiento

Android incorpora en su código la plataforma **SQLite**. Ésta se caracteriza por ser ligera, no necesitar servidor, precisar de poca configuración, ser transaccional y de código libre. Al estar ya presente en el SO, no se ha buscado otras opciones puesto que ya cumplía con las necesidades de almacenamiento de datos requeridos en el proyecto.

Proyección cartográfica

La proyección de mapas en Android no sólo se limita a su dibujo en un sistema de coordenadas; también ha de implementar todos los métodos para que el usuario pueda interactuar. Por esta razón se han utilizado librerías que implementan todo lo necesario para facilitar la tarea de los programadores. Entre las librerías disponibles encontramos:

- *Google APIs Add-On*: es una extensión del entorno de desarrollo de Android SDK que entre otras incluye la biblioteca externa *Maps* que le permite añadir potentes capacidades de mapeo para aplicar a Android. Es necesaria la obtención de una clave de uso (API Key) y hay un límite de uso (gratuito) de 25.000 peticiones diarias por aplicación (durante el desarrollo de la aplicación, esta API ha sufrido una actualización importante) [5].
- *OsmoDroid*: es una aplicación y una API de OpenStreetMap que básicamente substituye a la API de Google para mapas. La diferencia principal recae en la licencia, que es libre. Tiene conexión directa a los mapas servidos por OpenStreetMap y puede usarse offline [6].
- *MapsForge*: API de código abierto que puede utilizarse offline. Puede dibujar tanto mapas raster como vectorial. Su objetivo es proporcionar herramientas libres y abiertas que permitan crear fácilmente nuevas aplicaciones basadas en OpenStreetMap [7].
- *MGM Maps Lib SDK*: es una API libre basada en Java Developer Kit Mobil Mapping que permite añadir características cartográficas. Se ha de generar una clave de evaluación de licencia para la aplicación y el nombre del proveedor [8].
- *MapQuest Android SDK*: puede utilizar datos de OpenStreetMap o su conjunto de datos con licencia. No tiene límites preestablecidos en los mapas con licencia libre Community Edition. Soporta archivos KML estándar y archivos GeoRSS [9].

- *ArcGIS Runtime SDK*: API de ArcGIS para Android que permite crear aplicaciones que utilizan las funciones personalizadas que proporciona ArcGIS Server utilizando Java. La API incluye un *plug-in* para el entorno de desarrollo integrado (IDE) de Eclipse. No es una API libre y requiere de registro en el Web de Esri [10].

De las opciones revisadas, finalmente se ha decidido utilizar **Osmdroid**. Se inició el proyecto con la API de Google, pero en ciertos aspectos de simbolización y representación cartográfica offline no se adecuaba a las necesidades. Osmdroid sí lo hacía, y además al tratarse de código libre (con licencia GNU Lesser GPL) se podían hacer modificaciones en la API para integrarla a la aplicación.

Osmdroid acepta tanto cartografía online como offline. Aún así, la cartografía offline ha de servirse de una forma muy concreta. Se han creado los mapas con el programa **Mobil Atlas Creator (MOBAC)** [11]. Este programa crea un conjunto de *tiles* (imágenes) del área seleccionada y son exportados y compactados en formato ZIP. La carpeta que contiene las imágenes dentro del ZIP necesariamente ha de llamarse "Mapnik" para que Osmdroid la reconozca. Finalmente se ha de situar el archivo ZIP dentro de la carpeta Osmdroid que se crea al instalar la aplicación.

Conclusiones

Geolocalización

En los dispositivos móviles la localización geográfica puede basarse en dos sistemas (y su combinación). Por un lado se pueden posicionar a través de la red; es decir, a través del proveedor de servicios. El problema es que requiere proximidad al proveedor para tener precisión (comúnmente llamado localización por red o WiFi). Por otro lado, los dispositivos con receptor de señales y el software adecuado pueden posicionarse a través de las señales externas. De estos, el sistema más utilizado es el GPS. El problema de estos últimos se da en espacios cerrados o estrechos (como calles) donde la señal de los satélites no puede llegar o llega distorsionada.

Como la aplicación se define para su uso en campo, y las zonas de alud se encuentran en la montaña, donde posicionarse a través de redes sería difícil, la aplicación está preparada sólo para usar el posicionamiento por **GPS**.

Realidad Aumentada

La realidad aumentada es una tecnología donde la visión de la realidad a través de un dispositivo se ve ampliada con elementos virtuales que son añadidos o superpuestos a esta visión real. Este proceso se basa en tecnologías derivadas de la visualización y/o reconocimiento de la posición para crear un sistema que reconozca la información real que hay a su alrededor y crear así una nueva capa de información virtual. Esta información se mezcla con el mundo real de forma que para el usuario coexisten objetos virtuales y reales en el mismo espacio.

Para la simbolización de las zonas de alud se ha recurrido al sistema de realidad aumentada basada en la posición y la orientación. Es la utilizada en la mayoría de las aplicaciones actuales de realidad aumentada y no funciona en el reconocimiento en las imágenes de la cámara. Por tanto requiere de un sistema de localización y de sistemas que reconozcan la orientación del dispositivo, como son las brújulas digitales, acelerómetros, etc.

Hoy en día hay muchas aplicaciones y APIs que trabajan con la realidad aumentada:

- *Wikitude SDK*: el uso de herramientas estándar de la industria del desarrollo hace que sea fácil de integrar en una aplicación. Wikitude SDK está disponible gratis cuando se utiliza en proyectos no comerciales, pero también requiere del registro y permiso de su uso. Además, la gestión de los datos para mostrar en la realidad aumentada depende de Wikitude (y sus servidores) o de servidores que trabajan con esta plataforma [12].
- *Layar*: es una biblioteca estática que implementa la funcionalidad básica de la app de Layar en una aplicación, como la carga de una capa y la presentación de la visión de realidad aumentada. Igual que con Wikitude, la gestión de datos depende de sus servidores [13].
- *Mixare*: es un visor de realidad aumentada de código abierto, publicado bajo la licencia GPLv3. Funciona como una aplicación autónoma o se puede integrar en el código de una aplicación a través de su API. Es relativamente antigua (en términos de programación) y el código puede ser confuso debido a su extensión y complejidad [14].
- *Appunta*: es una API que permite no sólo mostrar toda la información geosposicional para el usuario, sino que también puede crear nuevas formas de mostrar la información. Es una API con código OS con licencia Apache 2.0. Esto permite que pueda ser modificada para la buena integración en aplicaciones propias [15].
- *Junaio*: ofrece un servicio gratuito y se constituye como navegador de realidad aumentada basado tanto en la geolocalización como en el reconocimiento de imágenes. Como las posibilidades más comerciales, Junaio requiere de conexión a internet para gestionar los datos a mostrar en la realidad aumentada [16].

En un principio se quiso prescindir de integrar y programar una solución propia para la realidad aumentada y utilizar una aplicación externa (Wikitude o Layar) que tendría que ser instalada en el dispositivo junto a la aplicación en desarrollo. Pero este hecho implicaba tener que subir los datos a servidores de terceros para su visualización y disponer de conexión a Internet. Al ser requisito gestionar los datos directamente desde el propio IGC y tener que trabajar sin conexión a la red, se decidió utilizar una APIs integrada en la aplicación y así poder trabajar offline. Debido a la complejidad del código de Mixare, se optó por utilizar **Appunta**. Al tratarse de una API sencilla y con código simplificado, fue fácil integrar la realidad aumentada en la aplicación IdAllau.

ENTORNO DE PROGRAMACIÓN

Las aplicaciones para Android se programan en lenguaje Java y son ejecutadas en una máquina virtual especialmente diseñada para esta plataforma llamada Dalvik. El núcleo de Android está basado en Linux 2.6. Hay más de una opción para desarrollar aplicaciones en Android, y en este caso se ha escogido Eclipse. Son necesarias las siguientes herramientas para completar el entorno de programación:

- *JDK de Java*: se trata de un conjunto de herramientas (programas y librerías) que permiten desarrollar programas en lenguaje Java [17].
- *Eclipse*: es un entorno de desarrollo integrado (IDE) de código abierto y libre multiplataforma. Es extensible mediante módulos y *plug-ins*. Está orientado inicialmente a la creación de aplicaciones Java. Los propios creadores de Android recomiendan como IDE para el desarrollo en su SO a Eclipse; y por eso

es el único que dispone oficialmente de un *plug-in* que facilita todo el proceso de desarrollo, el ADT [18].

- *SDK Android*: es un kit de desarrollo de software que contiene el conjunto de herramientas básicas que permiten compilar y depurar aplicaciones escritas para Android; así como empaquetar y firmar las aplicaciones finalizadas [19].
- *ADT*: de sus iniciales *Android Development Tools*, es un *plug-in* diseñado específicamente para Eclipse. Esta herramienta incorpora a Eclipse los menús y opciones que facilitan los trabajos habituales en el desarrollo de aplicaciones [20].

Si bien es cierto que existe la máquina virtual Dalvik que permite probar y depurar las aplicaciones en diferentes escenarios, en este caso diferentes dispositivos móviles virtuales, no siempre se pueden probar todas las funciones que se pueden implementar en Android. Por esta razón se han utilizado diferentes dispositivos para depurar la aplicación.

DISEÑO

En la Figura 1 se puede ver un esquema general de la aplicación IdAllau. Está dividida en 6 bloques (de colores diferentes) que representan las diferentes funcionalidades de la aplicación.

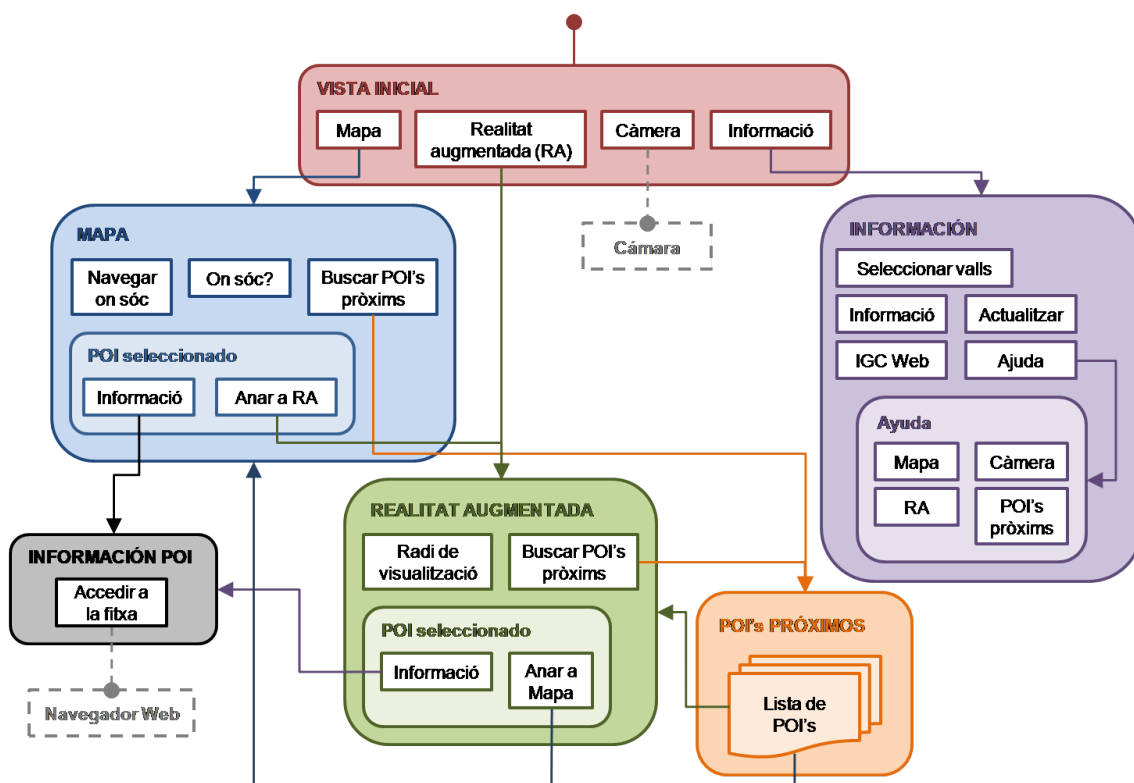


Figura 1: Diseño de la aplicación IdAllau.

Vista inicial

Al iniciarse la aplicación, ésta pregunta al usuario que zonas de alud quiere visualizar. Como hay una gran cantidad de datos, para mejorar el rendimiento se

representan sólo aquellas zonas de alud necesarias, al estar codificadas a partir del valle al cual pertenecen, el usuario selecciona aquellos que le interesan.

En el caso que sea la primera vez que se abre la aplicación, se ha de crear la base de datos con todas las zonas de alud. Mientras dura el proceso (que puede llevar varios minutos dependiendo de las características del dispositivo) aparece un aviso indicando que se está produciendo la creación de la base de datos.

Finalmente, si el posicionamiento por GPS no está activado, aparece un aviso indicándolo y da la posibilidad de ir directamente a las opciones del dispositivo para activarlo.

Una vez seleccionados los valles a representar, aparece el menú con las diferentes opciones; para ir al mapa, a la realidad aumentada, a la cámara o a la información de la aplicación. La realidad aumentada no será accesible si no hay posicionamiento por parte del GPS o cámara trasera en el dispositivo.

Mapa

Al iniciarse el mapa, la aplicación crea la simbolización de las zonas de alud como puntos de interés (POI's), que son los centroides de los polígonos que representan las zonas de alud, en función de los valles seleccionados.

Se puede navegar libremente por el mapa, aunque no haya datos de posición del usuario. Si el posicionamiento por GPS está desactivado, aparece un símbolo en la parte superior derecha que así lo indica. Tampoco estarán disponibles las opciones del menú, que son las siguientes:

- *"Navegar on sóc" (Navegar donde estoy): posiciona al usuario en el centro del mapa.*
- *"On sóc? (¿Dónde estoy?):" muestra las coordenadas del usuario.*
- *"Buscar POI's pròxims" (Buscar POI's próximos): abre una lista de las zonas de alud próximas al usuario después de preguntar al usuario por el radio de búsqueda.*

Finalmente, al seleccionar un POI, éste queda seleccionado (cambiando de color) y se muestra un cuadro con su información básica y dos opciones. La primera lleva a la vista de información del POI, para ver más información, y la segunda a la realidad aumentada manteniendo el POI seleccionado.

Realidad aumentada

Al iniciarse se crea la capa virtual que se superpone encima de la realidad captada a través de la cámara del dispositivo. Como en el caso del mapa, ésta se crea a partir de la consulta a la base de datos mediante el código de los valles seleccionados. Además de la capa virtual en sí, los POI's también son simbolizados en un pequeño radar situado en la parte superior derecha de la vista. Esta vista tiene dos opciones en el menú:

- *"Radi de visualizació" (Radio de visualización): hace aparecer una barra deslizante (o desaparecer si ya está visible) que controla el radio máximo de visualización de los POI's. De esta forma se puede determinar en qué radio el*

usuario quiere ver las zonas de alud que tiene a su alrededor. El radio seleccionado puede verse debajo del radar.

- *"Buscar POI's pròxims" (Buscar POI's próximos): como en el caso del mapa, abre una lista de las zonas de alud próximas al usuario aunque aquí la distancia se determina a partir de la seleccionada en la barra deslizante del control de radio de visualización.*

Esta vista tiene un funcionamiento igual que en el caso del mapa con respecto a seleccionar un POI. Aparece un cuadro con su información básica y dos opciones; aunque en este caso la segunda opción servirá para ir al mapa manteniendo el POI seleccionado.

Cámara

No se ha programado un recurso propio para la cámara, por tanto la aplicación busca alguna otra aplicación que pueda hacer la foto. Una vez obtenida, IdAllau se encarga de darle nombre (la fecha y hora en que se ha realizado) y de guardarla dentro de la carpeta de la aplicación.

Información del POI

Se recibe la información del POI que estaba seleccionado (en el mapa o en la realidad aumentada) y se busca toda su información en la base de datos y se visualiza. En caso de que el usuario tenga acceso a internet, puede consultar la ficha completa correspondiente a esta zona de alud en los servidores del IGC.

Lista de POI's próximos

En forma de lista aparecen aquellas zonas de alud que están dentro del rango de indicado. Para conocer cuáles son, la aplicación entra en la base de datos y calcula la distancia que hay entre el usuario y cada una de las zonas de alud. Si la distancia es menor, se agrega la zona de alud a la lista. Al seleccionar una de las zonas de alud, la aplicación devuelve al usuario a la vista desde donde se ha hecho la consulta y selecciona la zona de alud.

Información

En forma de lista se presentan diferentes opciones:

- *"Seleccionar valls" (Seleccionar valles):* permite al usuario volver a seleccionar las zonas de alud que quiere visualizar.
- *"Informació" (Información):* se muestra información sobre la aplicación, como puede ser la versión, el desarrollador, instituciones participantes, etc.
- *"Actualitzar" (Actualizar):* borra los datos de la base de datos, descarga los nuevos datos del servidor del IGC y, a partir de éstos, rellena otra vez la base de datos.
- *"Ajuda" (Ayuda):* un recopilatorio de información de ayuda sobre la aplicación para los nuevos usuarios.
- *"IGC Web":* permite el acceso a la página web del IGC si hay conexión a Internet disponible.

CONCLUSIONES

Desarrollar la aplicación en Android y librerías o APIs de código libre a supuesto un reto, ya que no siempre hay una documentación oficial para ayudar en el desarrollo y en la implementación dentro de una aplicación propia. Sin embargo, la gran comunidad de desarrolladores ligada a los proyectos de código libre y a la constante revisión de la documentación existente han permitido llevar a cabo los objetivos planteados.

Trabajar con las librerías de Osmdroid y Appunta ha sido satisfactorio, aunque no se han podido explorar, más allá de la documentación, otras opciones que parecían igualmente interesantes.

También ha sido un reto crear una aplicación que sea funcional sin conexión a internet. En un mundo cada día más conectado a la red, muchas de las APIs y funciones requerían de esta conexión. Finalmente, se ha podido superar la dependencia a la red.

Como con cualquier aplicación para Android, su funcionamiento óptimo estará determinado por el dispositivo móvil donde se instale, tanto por parte del software (la versión de Android instalada) como por el hardware. Es prácticamente imposible probar la aplicación en todos los dispositivos Android existentes en el mercado.

Finalmente, exponer algunas limitaciones de la aplicación, algunas de las cuales son debidas al uso de las librerías elegidas:

- La simbolización de los POI's en la realidad aumentada se realiza a una única altura y no dependiendo de la altitud en que se encuentren los POIs. De las aplicaciones y APIs revisadas, ninguna parecía solucionar este problema.
- La aplicación permite seleccionar POIs que no estén visibles (porque no están en el rango de visión definido). Appunta no tenía implementado el cambio de rango de visualización, y por tanto, dentro del rango fijo se visualizaban todos los elementos existentes. La API tendría que ser modificada para tener en cuenta que no puede seleccionarse un POI que no esté visible.
- La aplicación no puede hacer capturas de pantalla (función muy interesante en la realidad aumentada para poder capturar tanto la vista real de la cámara como información virtual superpuesta). Actualmente, los nuevos dispositivos móviles (Smartphone) ya cuentan con una combinación de teclas para poder hacer capturas, así que en un futuro puede llegar a implementarse esta función.

AGRADECIMIENTOS

Agradecer tanto al MTIC como al LIGIT por darme la oportunidad de desarrollar esta aplicación para móviles, en una clara apuesta para el futuro de los sistemas de información geográficos. También agradecer a mis dos tutores, a José Quirós Jiménez (UAB) y Jordi Marturià Alavedra (IGC) por sus consejos y asesoramiento durante el desarrollo de la aplicación. Tampoco olvidar a Neus Querol Vidal (IGC) por su implicación en el proyecto.

REFERENCIAS

- [1] Institut Geològic de Catalunya: <http://www.igc.cat/web/ca/index.php>
- [2] Màster en Tecnologies de la Informació Geogràfica: <http://ligit0.uab.es/mtig/index.htm>
- [3] Gartner Smart Phone Marketshare: <http://www.gartner.com/newsroom/id/2237315>
- [4] Datos del uso de las diferentes versiones del sistema operativo Android:
<http://developer.android.com/about/dashboards/index.html>
- [5] Google APIs Add-On. Available: <https://developers.google.com/android/add-ons/google-apis/>
- [6] OsmDroid. Available: <http://code.google.com/p/osmdroid/>
- [7] MapsForge. Available: <http://code.google.com/p/mapsforge>
- [8] MGMaps Lib SDK. Available: <http://www.nutiteq.com/mobile-map-api-sdk-guides>
- [9] MapQuest Android SDK. Available: <http://developer.mapquest.com/>
- [10] ArcGIS Runtime SDK. Available:
<http://www.esri.com/software/arcgis/smartphones/develop>
- [11] Mobil Atlas Creator. Available: <http://mobac.sourceforge.net>
- [12] Wikitude. Available: <http://www.wikitude.com/developer>
- [13] Layar. Available: <http://www.layar.com/>
- [14] Mixare. Available: <http://www.mixare.org/>
- [15] Appunta. Available: <http://appunta.com/>
- [16] Junaio. Available: <http://www.junaio.com/>
- [17] JDK de Java. Available:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [18] Eclipse. Available: <http://www.eclipse.org>
- [19] SDK Android. Available: <http://developer.android.com/sdk/index.html>
- [20] ADT. Available: <https://dl-ssl.google.com/android/eclipse/>