GPSLoc. Localización y Relaciones Sociales en el ámbito de los Teléfonos Inteligentes.

Pablo Sáez Sáez

Director: Vicente Pelechano Ferragud
Co-Director: Ismael Torres Boigues

2011

UNIVERSIDAD POLITECNICA DE VALENCIA

DSIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

ÍNDICE

AGRADECIMIENTOS

Aprovecho este apartado para enviar un fuerte abrazo y agradecimientos especiales a Pablo Muñoz, porque gracias a él descubrí la existencia de esta proposición de tesina, a Vicente Pelechano e Ismael Torres, directores de mi tesina, por toda la ayuda y apoyo que me han dado y las dudas que me han resuelto, y finalmente a mis padres Fermín y Mª Ángeles, a mi novia Estefanía y a mis amigos, por soportar todas esas incontables horas en las que he estado encerrado en mi cuarto sin dar señales de vida.

A todos ellos, muchas gracias.

1. PRESENTACION DE LA TESINA

Con el paso de los años las aplicaciones software han ido evolucionando a lo largo del tiempo, y junto a ellas también el hardware y las tecnologías que sustentan estas aplicaciones. En la actualidad estamos viviendo el boom de las aplicaciones para dispositivos móviles, gracias al auge que están teniendo estos dispositivos, los cuales reúnen cada vez más características de los equipos de sobremesa, aportando además ciertas ventajas que estos no poseen, como son la movilidad y el tamaño.

El auge de las redes sociales y la tecnología móvil han revolucionado las vías tradicionales de comunicación, tanto a nivel personal como a nivel empresarial. La tecnología móvil es una de las herramientas fundamentales del cambio que está sufriendo el panorama software actual.

Con la llegada de los smartphones las posibilidades de los teléfonos móviles han crecido de manera abrumadora [1], hasta llegar a lo que son hoy en día: herramientas fundamentales de trabajo. Esto está derivando en aplicaciones cada vez más complejas, seguras, optimizadas y funcionales en los dispositivos móviles actuales. Este es el caso de las aplicaciones de geolocalización, las cuales están experimentando un crecimiento importante en todas las plataformas móviles.

En la memoria de la tesina “GPSLoc. Localización y Relaciones Sociales en el ámbito de los Teléfonos Inteligentes”, se va a seguir y detallar el proceso de creación de la aplicación GPSLoc, una aplicación capaz de ofrecer a todos los usuarios de Android la posibilidad de mantener localizadas a otras personas, según diferentes criterios, los cuales serán explicados a lo largo del documento. El motivo de la elección de Android como plataforma para la cual desarrollar la aplicación GPSLoc es la rápida evolución y expansión que está teniendo este sistema operativo, el cual está desbancando uno tras otro a todos sus competidores en características y cuota de mercado.

2. PROPUESTA

Durante este capítulo será explicada la especificación de requisitos del proyecto siguiendo el estándar IEEE Std. 830-1998 mediante el cual serán definidos los requisitos y funcionalidades que tendrá la aplicación GPSLoc.

2.1. INTRODUCCION

En este punto serán descritos los requisitos y funcionalidades de GPSLoc.

2.1.1. PROPOSITO, MOTIVACION Y RETOS

Con el presente documento se persigue describir los requisitos de GPSLoc para poder desarrollar cada una de las funcionalidades deseadas.

El crecimiento de la demanda de aplicaciones de geolocalizacion experimentado en todas las plataformas móviles existentes y la excesiva semejanza entre las aplicaciones existentes en el sector son las principales motivaciones de este proyecto. A pesar de existir diversas opciones en el mercado, el reto que se persigue es ofrecer una aplicación diferente, que aporte algo más, ya que las aplicaciones existentes en la actualidad son muy similares entre sí, y ven reducida su utilidad al ámbito de la familia y los amigos. Este proyecto busca ofrecer geolocalización no solo para los usuarios comunes, sino también para, por ejemplo, empresas con sus empleados o para padres con sus hijos. Para ello será necesaria la implementación de diferentes tipos de permisos de visibilidad (total, por horas y por días), algo innovador hasta el momento. Estudios recientes han demostrado la importancia de la privacidad en la geolocalizacion [18], motivo por el cual es necesario tener constancia de quién nos hace una petición y qué nos está pidiendo [16]. En muchas ocasiones los usuarios no son conscientes de la información que publican a través de la red ni de los riesgos que esto conlleva en el caso de la geolocalización [17], lo cual hace aún más necesaria la gestión de la privacidad en este tipo de aplicaciones. Para respetar estos principios serán implementados los citados permisos de visibilidad y un sistema de invitaciones que permitirá gestionar la privacidad eficientemente. Además, las aplicaciones actuales solo ofrecen la posibilidad de ver la última posición de cada usuario, mientras

que con GPSLoc, se buscará dar un nuevo enfoque más amplio, ofreciendo la posibilidad de ver no solo la última, sino las últimas posiciones de un usuario, para poder observar donde ha estado recientemente.

2.1.2. ALCANCE

Definición y desarrollo completo de una novedosa aplicación de localización para la plataforma Android, cuyo objetivo es ofrecer a los usuarios la posibilidad de localizar a otros usuarios, según varios tipos diferentes de permisos. La aplicación requerirá un servidor remoto para centralizar el almacenamiento de la información de los usuarios.

2.1.3. DEFINICIONES, SIGLAS Y ABREVIATURAS

- *Usuario:* Persona física dada de alta en el sistema
- *Amigo:* Es la persona sobre la cual se posee un permiso.
- *Persona externa:* Persona física externa al programa, la cual no se encuentra dada de alta en el mismo
- *GPS:* Global Positioning System o sistema de posicionamiento global. Es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de una persona
- *BD Local:* Base de datos local (interna) de la aplicación GPSLoc
- *BD Remota:* Base de datos remota, alojada en el servidor remoto.
- *Permiso:* Es la relación de visibilidad de un usuario sobre las posiciones de otro.
    - *Permiso Total:* Un usuario tiene visibilidad sobre otro usuario a cualquier hora del día y en cualquier día.
    - *Permiso Por Horas:* Un usuario tiene visibilidad sobre otro usuario solo durante un determinado rango de horas, por ejemplo, desde las 8:00 AM hasta las 8:00 PM.
    - *Permiso Por Días:* Un usuario tiene visibilidad sobre otro usuario solo durante un determinado rango de días, por ejemplo, desde el día 1 de febrero hasta el día 10 de febrero.

- *Invitación:* Un usuario tiene una invitación cuando otro usuario ha solicitado algún tipo de permiso sobre él.
- *Posición:* Es la posición GPS de un usuario en un determinado momento
- *Posición Válida:* Es la posición para la cual se cumple un permiso.
- *UPV:* Universidad Politécnica de Valencia
- *APK:* Aplicación para Android

2.1.4. ESTRUCTURA DEL DOCUMENTO

El primer capítulo de la memoria está dedicado a la presentación de la misma, mientras que el capítulo 2 es una descripción completa del comportamiento del sistema que se va a desarrollar. Durante este capítulo y siguiendo este orden, serán detalladas una descripción global del proyecto incidiendo en la perspectiva, las funciones, las características del usuario, las restricciones y las dependencias, y una completa descripción de los requisitos del proyecto, tanto funcionales como no funcionales.

En el siguiente capítulo serán descritos los aspectos tecnológicos empleados durante el desarrollo de la tesina, incluyendo la arquitectura, el entorno de desarrollo, el servidor remoto, la base de datos remota, los diferentes lenguajes de programación utilizados y la plataforma Android.

El capítulo 4 de la memoria ofrecerá una demostración de las funcionalidades de la aplicación mediante el estudio de un escenario real en el que utilizar GPSLoc.

A lo largo de la extensión del capítulo 5 de la memoria serán explicados los problemas encontrados durante el desarrollo de la aplicación, tanto debidos a las limitaciones de Android, como a cualquier otro tipo de restricción, como por ejemplo el hecho de que se trate de una aplicación para dispositivos móviles, lo cual delimita una serie de aspectos que son críticamente diferentes en las aplicaciones desarrolladas para equipos de escritorio. Uno de los aspectos más importantes en este ámbito es el hecho de que la conexión a Internet de los dispositivos móviles es muy limitada y dependiente del lugar en el que se encuentre el dispositivo, por lo que es necesario mantener el tráfico en red lo más reducido posible.

En el capítulo 6 se tratarán las posibles mejoras y futuros avances en GPSLoc, y, finalmente, en el capítulo 7, se detallarán las conclusiones obtenidas durante el desarrollo de la tesina y la elaboración de la aplicación GPSLoc, para cerrar con un capítulo 8 dedicado a la bibliografía. Además, como anexo, se incluye un manual de la aplicación.

2.2. DESCRIPCION GLOBAL

En los siguientes puntos serán descritas otras alternativas existentes en el mercado y se dará a conocer una visión global de las principales funcionalidades de GPSLoc.

2.2.1. ALTERNATIVAS EXISTENTES EN EL MERCADO

GPSLoc será una aplicación para Android, que permitirá mantener localizadas a otras personas, ya sean amigos, familiares o empleados de una empresa, según varios tipos de permisos de visibilidad disponibles. Existen otras aplicaciones similares en el mercado que utilizan diferentes tecnologías, como son los servicios web y las redes sociales. Encontramos aplicaciones situadas en el ámbito de la geolocalización en diferentes plataformas: Android, Iphone, Facebock, etc. Algunos de los exponentes más significativos dentro del ámbito de la geolocalización son los siguientes: Google Latitude, (el cual se integra con el servicio Google Maps y muestra de forma sencilla la posición de otros usuarios), Life360 (aplicación orientada a mantener localizados a los miembros de la familia), GPS Tracking (una sencilla aplicación de geolocalización, que tiene la ventaja de ser multiplataforma) y Foursquare (una red social de lugares, cuya característica principal es que no se centra en ofrecer la posición exacta en tiempo real de los usuarios, sino el lugar en el que se encuentran, lugar que previamente debe de haber sido dado de alta en el sistema). A continuación serán explicadas de forma más concisa dichas aplicaciones.

GOOGLE LATITUDE:

Google Latitude es un servicio capaz de representar la posición geográfica actual del usuario y la de sus amigos en Google Maps con la intención de facilitar la localización de personas [2] [3].

Los recursos básicos que utiliza Google Latitude para implementar el servicio son dos: las coordenadas calculadas por el GPS y una conexión de datos mediante servicios web con el servidor remoto.

Google Latitude interpreta las coordenadas, las empaqueta y las envía, a través de la conexión a Internet del teléfono móvil, al servidor encargado de compartirlas con el grupo de amigos previamente configurado.

Figura 1 – Google Latitude

Las funcionalidades principales que ofrece Google Latitude son: ver las posiciones de los amigos, compartir la ubicación, gestionar la privacidad de la ubicación (quién puede verla y a qué nivel de detalle), visitar sitios y compartir la ubicación del usuario en dichos sitios. La desventaja principal de esta aplicación es su excesiva sencillez y escasez de funcionalidades extras.

LIFE360:

Life360 es una aplicación multiplataforma, disponible para Iphone y Android, que brinda la seguridad de saber dónde se encuentra la familia del usuario en todo momento. Ubica en un mapa a los miembros de la familia, y permite comunicarse con ellos aunque se produzcan fallos en infraestructuras principales [4]. Es una aplicación particularmente valiosa en situaciones de emergencia, como desastres naturales o de otro tipo. La aplicación Life360 automáticamente notifica a las familias sobre eventos de emergencia y puede enviar mensajes a determinados miembros de la familia [5].

Figura 2 – Life360

A diferencia de otros sistemas de notificación, las alertas de Life360 son personalizadas para cada usuario, por ejemplo, pueden transmitir la ubicación del refugio más cercano al usuario utilizando el GPS del teléfono. El sistema GPS también permite a los usuarios rastrear la ubicación de los miembros de su familia a tiempo real.

Con la red comunitaria Life360, los usuarios pueden pedir ayuda, así como ayudar a otras personas que utilicen el sistema. Por ejemplo, si un usuario pulsa el botón de pánico en su teléfono, todos los usuarios cercanos quedan notificados sobre el evento y reciben flujo de audio y vídeo de la cámara de la víctima.

Debido a que la red comunitaria utiliza tecnología GPS, todos los mensajes tienen una ubicación física en el mapa, y los usuarios tienen la opción de hacer visibles las alertas cercanas. Si bien la utilización de teléfonos móviles para comunicarse en situaciones de emergencia no es un concepto nuevo, Life360 aprovecha la tecnología GPS para proporcionar lo que la empresa cree que representa el futuro en situaciones de emergencia.

La principal desventaja de esta aplicación, es el hecho de que está situada principalmente dentro del ámbito familiar, de forma que resulta ineficiente para otro tipo de ámbitos como son la amistad o el trabajo.

GPS TRACKING:

Una sencilla aplicación de pago que permite localizar a otras personas mediante el envío de SMS. Sus principales funcionalidades son: localizar usuarios en tiempo real, integración con Facebook, envío de mensajes de texto, localización de grupos, subir fotos y gestión de la privacidad (quien puede o no puede ver al usuario) [6].

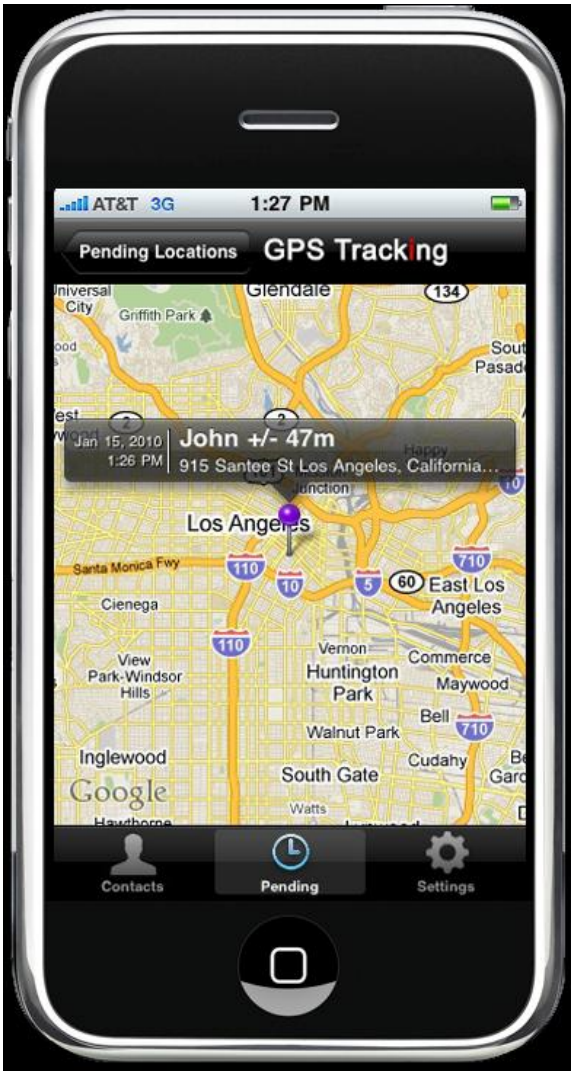

Figura 3 – GPS Tracking

Para sus desarrolladores, su principal ventaja respecto a los competidores es el hecho de que es capaz de ofrecer una ruta para llegar a la posicion en la que esté un amigo del usuario. Esto no es ciertamente una verdadera ventaja, ya que el cálculo de rutas

de un punto a otro es una funcionalidad que ya ofrece de forma gratuita Android mediante GoogleMaps, mientras que esta aplicación tiene cargos al ofrecer algunos de sus servicios.

GPS Tracking tiene en cambio la ventaja de ser multiplataforma, ya que está disponible tanto para Android como para Iphone, de forma que usuarios de Iphone pueden localizar a usuarios de Android y viceversa. La principal desventaja de esta aplicación, tal y como se ha comentado anteriormente, es el hecho de que es una aplicación que depende de servicios que requieren coste, al contrario que la gran mayoría de la competencia, que ofrece sus funcionalidades de forma gratuita.

FOURSQUARE:

Foursquare es una aplicación multiplataforma para teléfonos móviles Android, BlackBerry e iPhone, que aprovecha las capacidades de geolocalización de los dispositivos móviles actuales para compartir ubicaciones [7].

Figura 4 - Foursquare

La aplicación sigue una dinámica simple: se inicia Foursquare, se muestra un listado de lugares públicos que corresponden a la ubicación actual del usuario (de no existir da la posibilidad de crear el lugar donde se encuentre) y luego permite al usuario

registrarse en un lugar. Si se desea, la ubicación o registro es luego compartida con la lista de contactos de Foursquare.

Foursquare ofrece además compatibilidad con Facebook y Twitter, permitiendo compartir ubicaciones con usuarios de estas redes sociales.

Foursquare es una red social geolocalizada pero con una capa de entretenimiento, ya que a medida que el usuario se va registrando en distintos lugares, va ganando puntos de acuerdo a su número diario de registros, si es su primera vez en ese lugar, o si fue el primero en crearlo. Estos puntos asimismo se complementan con badges o placas, que van desde algunas referentes a un lugar en particular, hasta algunas que involucran a quienes se han registrado en el mismo lugar que el usuario, como por ejemplo BFF (Best friends for ever). Otro ejemplo es la categoría Mayor o Alcalde, que se le asigna al usuario cuando es quien más visita un lugar en particular, lo cual le agrega una jerarquía virtual a esta suma de factores.

Foursquare se sustenta gracias a su simple pero efectivo sistema de publicidad. Muestra publicidad relacionada con la ubicación del usuario, como por ejemplo: "Aprovechando que estás cerca del Fourbux de Plaza Central, los Frapuchinos están con 20% de descuento".

La principal desventaja de esta aplicación, es el hecho de que no ofrece en todo momento la posición real de la persona, sino que la sitúa en un lugar determinado y dado de alta en el sistema cuando ésta así lo solicite, de forma que no se está obteniendo información real de la verdadera posición de un usuario.

Todas las propuestas comentadas en este apartado tienen sus ventajas y sus defectos, pero sin duda se puede llegar a la conclusión de que hay hueco para una propuesta mas, ya que todas presentan algún tipo de carencia, ya sea por ser de pago, por ofrecer una funcionalidad pobre, o por carecer de alguna característica imprescindible. La aplicación que será desarrollada en este proyecto, GPSLoc, será innovadora respecto a todas, puesto que no solo proporcionará funcionalidades similares, sino que además ofrecerá dos aspectos nuevos, el sistema de permisos (total, por horas y por días) y la posibilidad de ver las últimas posiciones en las que se ha situado un usuario, y no solo su última posición.

2.2.2. SOLUCION PROPUESTA

En este proyecto se presenta la aplicación GPSLoc, una aplicación capaz de ofrecer a todos los usuarios de Android la posibilidad de mantener localizadas a otras personas, según cierto tipo de criterios, ya sean amigos, familiares, empleados de una empresa o tengan cualquier otro tipo de relación con el usuario de la aplicación. Para conseguir esto, la aplicación implementará tres tipos de permisos de visibilidad: permiso total (posición del usuario visible a cualquier hora del día y cualquier día de la semana), permiso por horas (posición del usuario visible solo durante un rango de horas y si así se indica, fines de semana incluídos) y permiso por días (posición del usuario visible sólo durante un determinado rango de días).

GPSLoc ofrecerá una amplia gama de funcionalidades atractivas para el usuario, desde las más típicas como registrarse, log-in, mostrar a los amigos en un mapa, seleccionar una foto de perfil visible por los demás usuarios, llamar por teléfono a un usuario, borrar permisos… hasta nuevas funcionalidades no implementadas por ninguna otra aplicación existente en la actualidad, como poder observar las diversas últimas posiciones en las que ha estado una persona, y no sólo su última posición.

Para poder ofrecer todas las funcionalidades será necesaria la utilización de un servidor remoto en el cual será alojada toda la información necesaria para el funcionamiento de la aplicación. GPSLoc utilizará una arquitectura de tres capas, interfaz, lógica y persistencia, para lograr una correcta y eficiente separación de contenidos y dependencia entre capas. Tal y como será explicado en los puntos siguientes, en Android, para poder realizar una conexión con un servidor remoto, es necesario utilizar algún tipo de puente intermediario. La solución propuesta en este proyecto consiste en un puente basado en ficheros PHP, con los cuales la aplicación se comunicará mediante codificación JSON. Estos ficheros PHP serán los que realizarán la comunicación con la base de datos remota y enviarán la información a la aplicación.

Uno de los principales objetivos del proyecto es cuidar los tiempos de espera que debe sufrir el usuario, los cuales son inevitables en una aplicación de esta naturaleza, que necesita estar constantemente enviando y recibiendo datos de un servidor remoto. Para este objetivo se adoptará la estrategia de utilizar una base de datos local, y tres servicios en background, que optimizarán los tiempos de espera del usuario al mínimo posible, siendo inviable eliminarlos por completo debido a la naturaleza de la aplicación. Los servicios en background son clases de Android que continúan su

ejecución en segundo plano sin necesidad de interactuar con el usuario. Estos aspectos tecnológicos de la aplicación serán detalladamente explicados en el apartado 3 de la memoria.

Se ha optado por utilizar la plataforma Android, debido a que es una apuesta de futuro, siendo actualmente la plataforma que experimenta un crecimiento más acelerado y progresivo de todas.

En los siguientes sub-apartados serán descritas las funcionalidades más importantes de GPSLoc, las cuales serán más detalladas y ampliadas en el apartado 2.3.2 de la memoria. Las funcionalidades explicadas a continuación irán acompañadas de imágenes extraídas del prototipo de la aplicación y serán las siguientes: la localización de amigos, la sección de invitaciones, y la sección de configuración.

2.2.2.1. Localización de amigos

La interfaz principal de la aplicación es el mapa de amigos, donde aparecerán sobre un mapa tipo GoogleMaps, todos los usuarios sobre los que se tiene permiso de visibilidad, para los cuales se cumpla el permiso en alguna de sus posiciones almacenadas en el servidor remoto. También será mostrado un punto rojo en el mapa, señalando la posición actual del usuario, si es que así lo ha indicado en la configuración el usuario logueado. Hay que destacar que, tal y como se puede observar en la Figura 5, encima del mapa se muestran cinco pestañas, cada una de las cuales muestra una interfaz diferente para visualizar a los amigos según los criterios explicados a continuación, diferenciados por el icono de la pestaña:

-  Muestra el mapa con todos los amigos para los cuales alguna posición cumple el permiso que se posee sobre ellos.
-  Muestra la lista de todos los amigos
-  Muestra la lista de los amigos sobre los que se posee permiso total
-  Muestra la lista de los amigos sobre los que se posee permiso por horas
-  Muestra la lista de los amigos sobre los que se posee permiso por días

Figura 5 – Mapa de amigos

Figura 6 – Vista detallada de un amigo

Al pulsar sobre un amigo en el mapa, será mostrado un cuadro de diálogo con su nombre, su imagen de perfil, su correo y el permiso que se tiene sobre él, y además, pinchando sobre el botón Show Details, se accederá a una nueva interfaz que mostrará la información detallada de cada amigo, tal y como se puede observar en la Figura 6. Esta interfaz ofrece la posibilidad de actualizar la posición de ese usuario, ver sus últimas posiciones, llamarle por teléfono, o borrar el permiso que se tiene sobre él. También es posible acceder a los detalles de un amigo a través de las cuatro pestañas que listan los amigos según el tipo de permiso, para ello se debe pulsar sobre el nombre del amigo.

2.2.2.2. Configuración

En la sección de configuración, tal y como se observa en la Figura 7, el usuario tendrá la posibilidad de cambiar su imagen de perfil, seleccionando una imagen que ocupe un máximo de 100KB (es recomendable seleccionar imágenes de tamaño muy reducido debido a la naturaleza de la aplicación y la velocidad de acceso a Internet de las conexiones de dispositivos móviles actuales), modificar su nombre completo, su número de contacto, cambiar su contraseña, e indicar si desea mostrar su posición actual en los mapas o no.

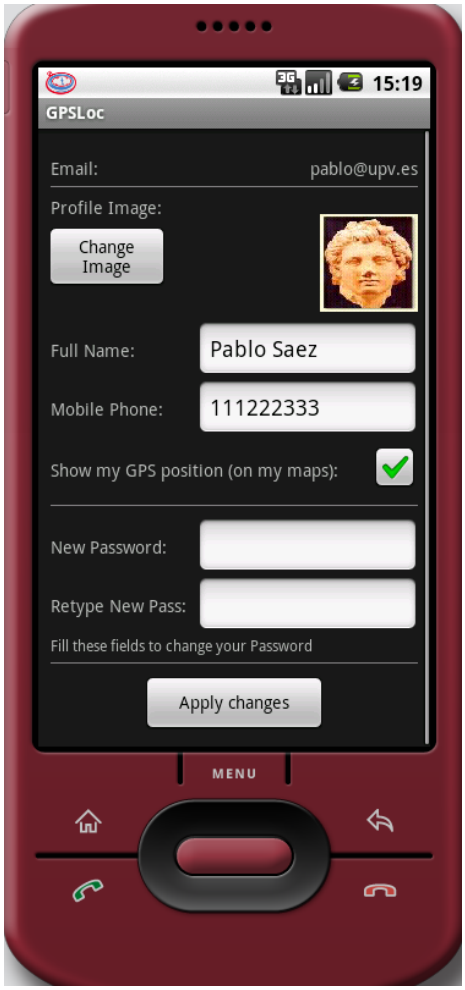

Figura 7 - Configuración

2.2.2.3. Invitaciones

En esta sección, tal y como podemos observar en la Figura 8, el usuario tendrá 4 opciones:

- Buscar usuarios de GPSLoc para enviarles una invitación.
- Invitar a un amigo externo a GPSLoc mediante el envío de un correo electrónico.
- Gestionar las invitaciones pendientes.
- Eliminar los permisos concedidos a otros usuarios.

Estos puntos serán explicados a continuación.

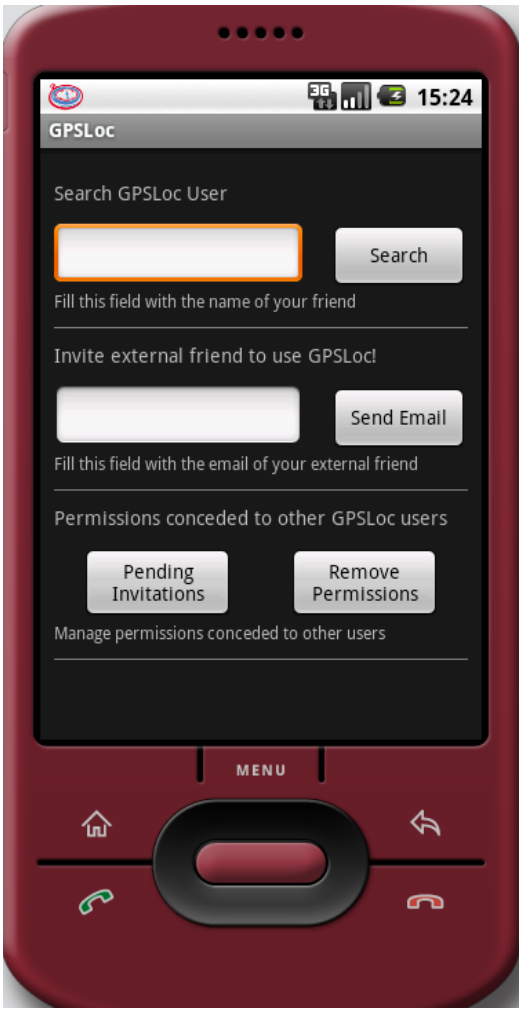

Figura 8 - Invitaciones

2.2.2.3.1. Buscar y enviar invitaciones a usuarios de GPSLoc

Introduciendo un nombre en el cuadro de texto inferior a "Search GPSLoc User" y pulsando el botón Search, GPSLoc buscará todos los usuarios cuyo nombre contenga el texto introducido. Es decir, si se introduce la cadena de texto "bl", aparecerán todos los nombres que contengan la secuencia de letras "bl", como por ejemplo: Pablo Sáez, Pablo Ruiz, etc. Si no existe ningún usuario registrado cuyo nombre completo contenga la secuencia de texto introducida, no aparecerá ningún resultado.

Una vez el usuario ha obtenido la lista de usuarios que cumplen con el criterio de búsqueda, pinchando sobre uno de ellos, podrá acceder a su información e imagen de perfil, y seleccionar el permiso que desea tener sobre él, indicando para cada tipo de permiso los parámetros necesarios, tal y como se observa en las imágenes 9 para permiso total, 10 para permiso por horas y 11 para permiso por días (las imágenes 10 y 11 son ligeramente más extensas debido a que es necesario desplazar la pantalla del móvil para visualizar toda la interfaz). Para el caso del permiso total, no será necesario completar ningún campo extra. Si es permiso por horas, es necesario indicar de qué hora a qué hora, además de si se desea obtener visibilidad en fin de semana. Si es permiso por días, habrá que indicar el rango de días en los que se desea tener visibilidad sobre la posición del usuario al que se le va a enviar la invitación.

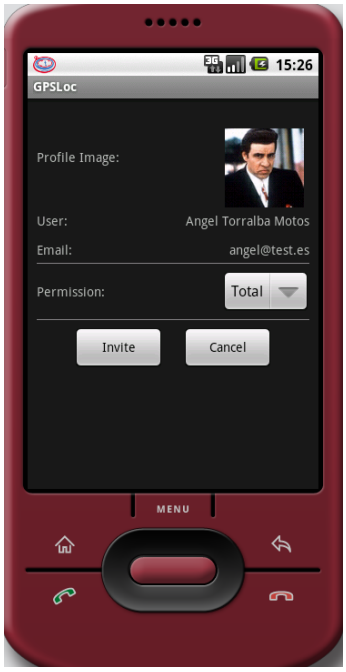

Figura 9 – Invitación per. Total

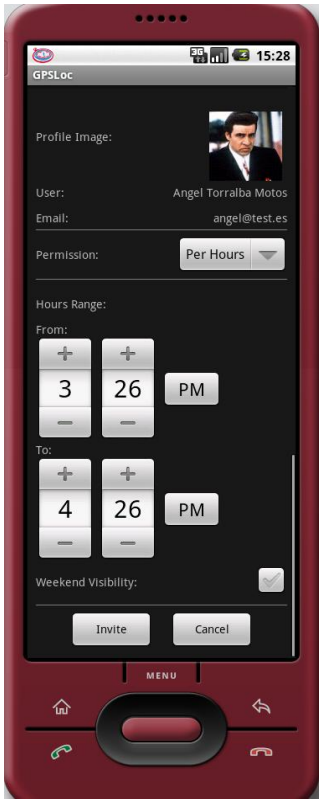

Figura 10 – Inv. per. Por Horas

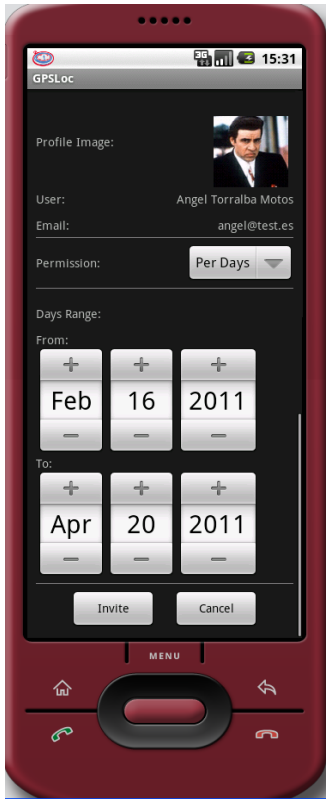

Figura 11 - Inv. per. Por Días

2.2.2.3.2. Invitar a un usuario externo a GPSLoc mediante email

Rellenando con el email de un amigo el segundo campo que se observa en la Figura 8, inferior al texto *"Invite external friend to use GPSLoc"*, y pulsando el botón *Send email*, la aplicación procederá a enviar un correo electrónico a la dirección indicada utilizando el gestor de correo electrónico de Android, permitiendo así al usuario modificar y personalizar el texto de la invitación, si así lo deseara, antes de confirmar el envío del email.

2.2.2.3.3. Gestionar las invitaciones pendientes

Pulsando en el botón Pending Invitations que se observa en la Figura 8, la aplicación mostrará una lista de los usuarios que han enviado al usuario logueado una invitación pendiente de aceptar o rechazar, y si no los hubiera, mostrará un mensaje avisando de que no existen invitaciones pendientes. En el caso de que sí existan, pulsando sobre una de ellas, será mostrada una nueva interfaz con la información del usuario que ha enviado la invitación, incluyendo su foto y las características del permiso que solicita con el usuario logueado, tal y como se observa en la Figura 12. El usuario logueado puede aceptar o rechazar la invitación mediante los dos botones de la parte inferior de la pantalla.

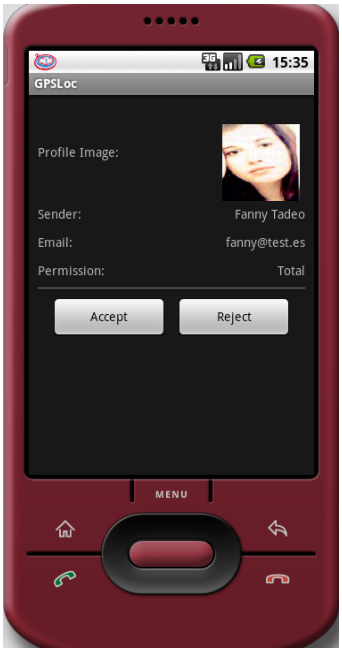

Figura 12 – Invitación pendiente

2.2.2.3.4. Eliminar permisos concedidos a otros usuarios

Pulsando en el botón Remove Permissions que se observa en la Figura 8, la aplicación mostrará una lista de los usuarios a los cuales el usuario logueado ha concedido un permiso de visibilidad, y si no los hubiera, la aplicación mostrará un aviso de que no ha sido concedido ningún permiso. En el caso de que sí los hubiera, pulsando sobre uno de ellos, será mostrada una nueva interfaz con la información del usuario al que se le ha concedido el permiso, incluyendo su foto y las características del permiso concedido, permitiendo al usuario logueado eliminarlo si así lo desea, tal y como se observa en la Figura 13.

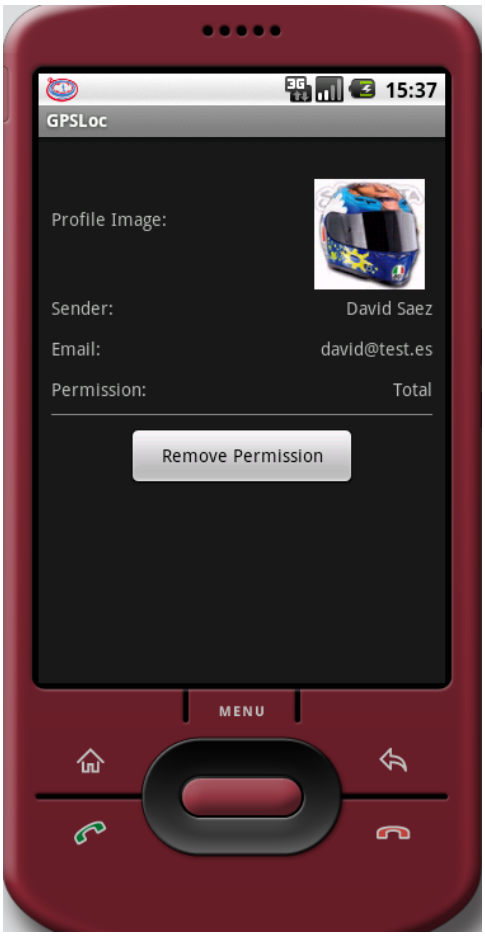

Figura 13 – Permiso concedido

2.2.3. DIAGRAMAS DE CASOS DE USO

Como complemento del punto anterior, en las siguientes imágenes de esta sección de la memoria se tratará de clarificar las funcionalidades que tendrá la aplicación mediante cuatro diagramas de casos de uso, acompañados de una descripción de cada uno de ellos. Cada caso de uso puede interpretarse como un requisito funcional de la aplicación y estos requisitos serán detalladamente explicados en el punto 2.3.2 de la memoria.

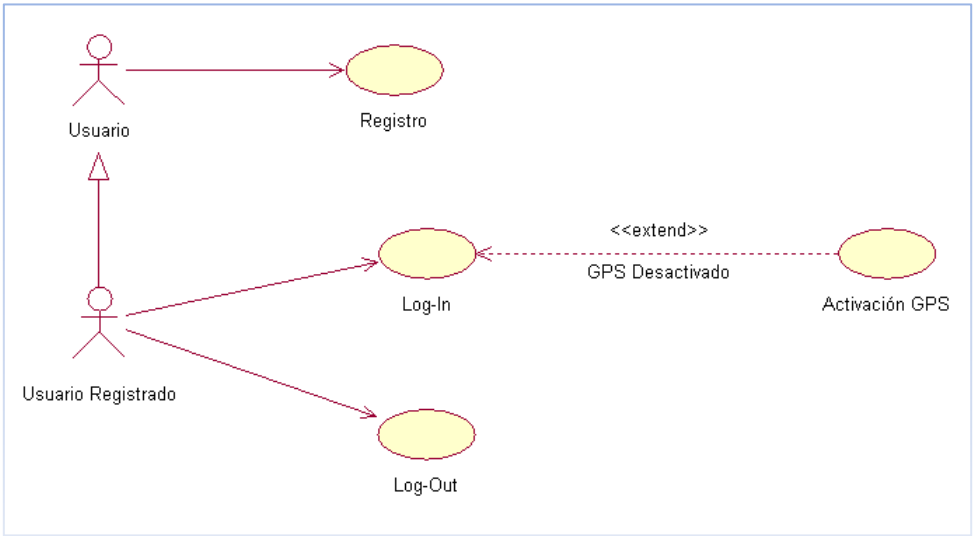

Figura 14 - Diagrama de Inicialización de la aplicación

C.U. Registro: El usuario podrá introducir sus datos personales y efectuar el registro de una nueva cuenta de usuario.

C.U. Log-In: El usuario introducirá su cuenta y su contraseña para hacer log-in y poder acceder a todas las características de la aplicación.

C.U. Activación GPS: Si el GPS está desactivado, la aplicación instará al usuario a activarlo, ya que es necesario para el funcionamiento de GPSLoc.

C.U. Log-Out: El usuario podrá hacer log-out de la aplicación en cualquier momento.

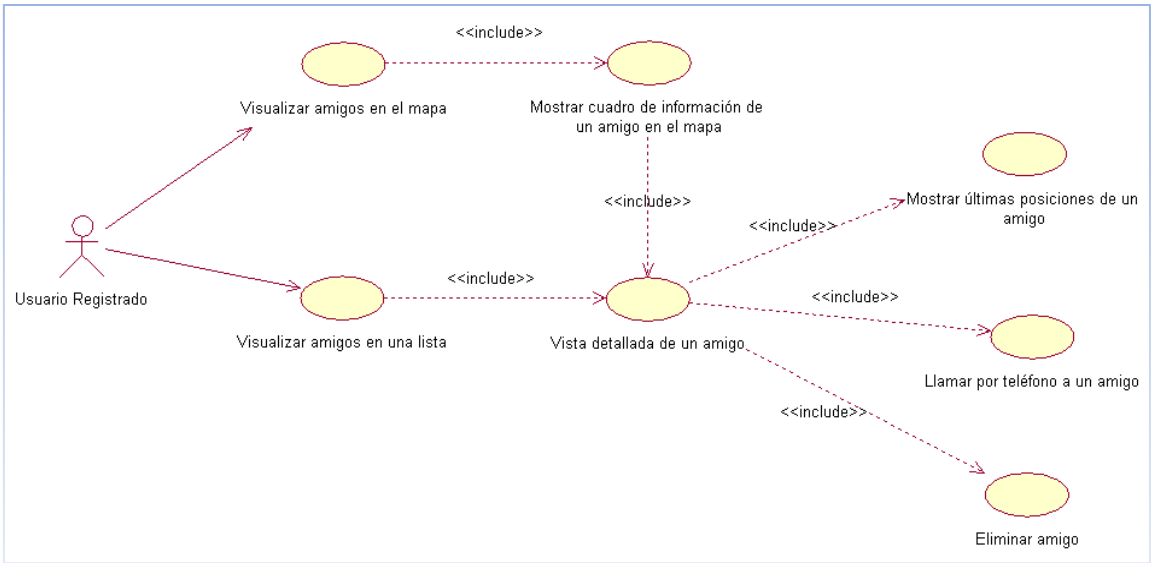

Figura 15 - Diagrama de Localización de amigos

C.U. Visualizar amigos en el mapa: Las últimas posiciones de todos los amigos del usuario serán mostradas en un mapa, de forma que el usuario podrá obtener una visión global de la posición de todos ellos.

C.U. Mostrar cuadro de información de un amigo en el mapa: Cuando el usuario pulse sobre un amigo, será visualizado un pequeño cuadro de información con la foto y los datos del amigo.

C.U. Visualizar amigos en una lista: El usuario podrá visualizar sus amigos mediante listas según el permiso que posea sobre ellos.

C.U. Vista detallada de un amigo: Con esta vista detallada serán visualizados los datos del amigo, los datos del permiso, la fecha de la última posición, y su posición en el mapa.

C.U. Mostrar últimas posiciones de un amigo: Esta característica permitirá mostrar en el mapa las últimas posiciones en las que ha estado un amigo, siempre que estas cumplan las condiciones que especifique el permiso que se posea sobre él.

C.U. Llamar por teléfono a un amigo: Pulsando sobre un botón será posible realizar una llamada telefónica al amigo.

C.U. Eliminar amigo: La aplicación permitirá eliminar amigos, mostrando un cuadro de advertencia antes de realizar la acción.

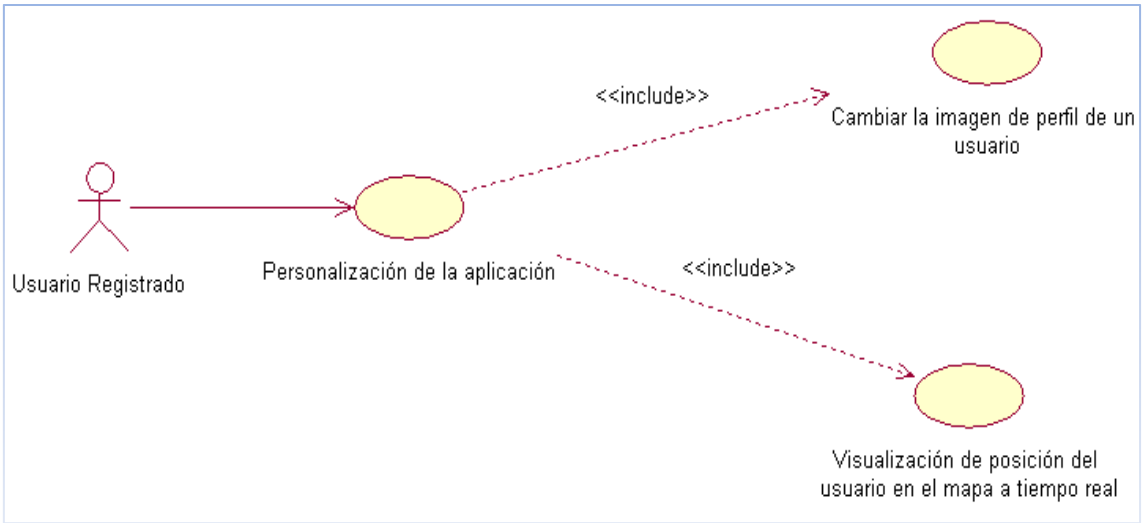

Figura 16 - Diagrama de Configuración

C.U. Personalización de la aplicación: La aplicación ofrecerá al usuario la posibilidad de cambiar su imagen de perfil, modificar sus datos, cambiar su contraseña y mostrar o no mostrar su posición en los mapas.

C.U. Cambiar la imagen de perfil de un usuario: Se le permitirá al usuario elegir, como foto de perfil, una de las imágenes almacenadas en su teléfono, siempre que ésta ocupe un máximo de 100KB.

C.U. Visualización de la posición del usuario en el mapa a tiempo real: El usuario podrá indicar si desea que su posición en tiempo real quede reflejada en los mapas, mediante una marca de validación en la ventana de configuración.

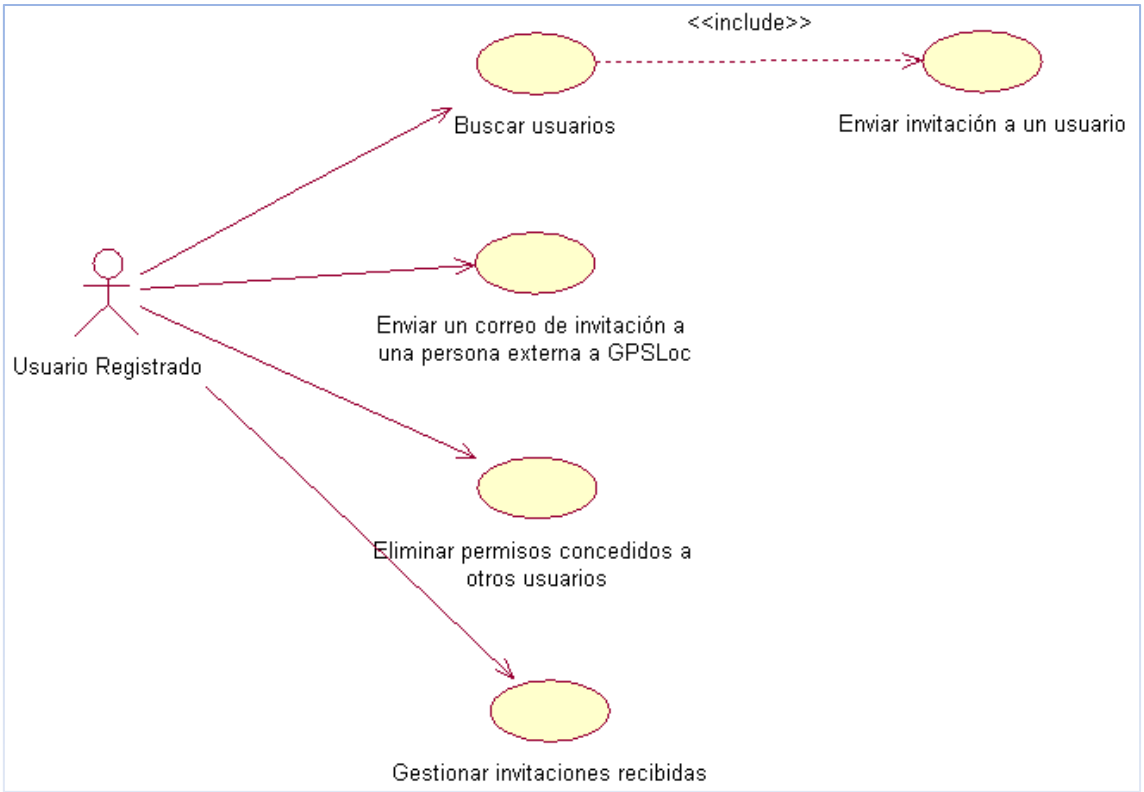

Figura 17 - Diagrama de Invitaciones

C.U. Buscar usuarios: La aplicación ofrecerá la posibilidad de buscar usuarios mediante la introducción de secuencias de texto y el filtrado entre todos los usuarios del sistema.

C.U. Enviar invitación a un usuario: El usuario podrá enviar invitaciones a otros usuarios, indicando el tipo de permiso que desea y las características del mismo.

C.U. Enviar un correo de invitación a una persona externa a GPSLoc: La aplicación permitirá enviar correos electrónicos de invitación a cualquier persona externa a GPSLoc mediante la introducción de su email.

C.U. Eliminar permisos concedidos a otros usuarios: El usuario podrá eliminar en todo momento cualquier permiso que haya concedido a otro usuario de GPSLoc.

C.U. Gestionar invitaciones recibidas: El usuario podrá visualizar sus invitaciones pendientes recibidas, enviadas por otros usuarios, de forma que podrá aceptarlas o rechazarlas según su criterio.

2.2.4. CARACTERISTICAS DEL USUARIO

Para poder acceder a la aplicación, primero, los usuarios deberán registrarse, y una vez registrados, podrán acceder a la misma mediante su nombre de usuario, representado por un correo electrónico, y su contraseña. Los usuarios registrados tendrán a su disposición todas las funcionalidades de la aplicación, mientras que los usuarios no registrados tan sólo podrán utilizar la funcionalidad de registro.

La utilización de GPSLoc no requiere ningún tipo de conocimiento avanzado de informática para los usuarios. El único requisito es estar familiarizado con el manejo de aplicaciones para dispositivos móviles.

2.2.5. RESTRICCIONES

GPSLoc depende de un servidor remoto, en el que almacena los usuarios, los permisos entre usuarios, y las posiciones de los mismos. Es recomendable que el servidor remoto sea potente, para ofrecer una navegabilidad fluida al usuario de la aplicación, ya que, a pesar de que GPSLoc dispondrá de una BD Local que agilizará la velocidad de la aplicación, muchas de las tareas requerirán conexión directa con el servidor y por ello depende de la velocidad del mismo que la aplicación sea fluida para el usuario.

La aplicación GPSLoc será implementada para ofrecer compatibilidad con el mayor numero de dispositivos Android posible, para ello se ha optado por maximizar la compatibilidad con versiones antiguas del sistema operativo. GPSLoc funcionará en dispositivos Android 1.5 o superior, es decir en prácticamente la totalidad de dispositivos móviles Android existentes.

Es requisito indispensable que el usuario active el GPS del dispositivo móvil para utilizar la aplicación. Para ello, si el usuario no lo ha activado, la aplicación le instará a hacerlo, siendo imposible activarlo de forma interna por la aplicación, debido a limitaciones del sistema operativo Android.

Gracias a la optimización del código de la aplicación, GPSLoc no tendrá requisitos de hardware, debido a que cualquier dispositivo capaz de soportar un sistema operativo Android será capaz de ejecutar el programa.

Es un requisito indispensable que el dispositivo móvil que va a ejecutar la aplicación posea acceso a Internet, preferiblemente 3G. Debido a la naturaleza de la aplicación, GPSLoc necesitará constantemente acceder a la red para obtener datos o enviar posiciones.

2.2.6. ATENCION Y DEPENDENCIAS

La aplicación ofrece compatibilidad total con la gran mayoría de versiones del sistema operativo Android existentes, 1.5, 1.6, 2.0, 2.1, 2.2 y 2.3. Sin embargo, cabe la posibilidad de que en versiones futuras del sistema operativo modifiquen la API que se proporciona a los desarrolladores, con lo que la aplicación podría requerir algún tipo de modificación en caso que esto ocurriese.

2.3. REQUISITOS ESPECIFICOS

En los siguientes puntos se describirán de forma detallada los requisitos de GPSLoc.

2.3.1. REQUISITOS DE INTERFACES EXTERNAS

2.3.1.1. INTERFAZ DE USUARIO

La interfaz de la aplicación cumple con los estándares más importantes de las interfaces para aplicaciones de dispositivos móviles [8]. Tal y como observaremos más adelante, la navegabilidad entre las principales áreas de la aplicación queda agrupada en el menú desplegable con la tecla "menú" del móvil. Cada interfaz muestra el mayor número de información posible sin excederse hasta límites molestos para el usuario, teniendo en cuenta el tamaño reducido de la pantalla de los dispositivos móviles. Además, para cierto tipo de interfaces, se utiliza el modelo maestro-detalle según sea necesario como se observará más adelante.

En la Figura 18 podemos observar la pantalla inicial de la aplicación, y en la Figura 19, se muestra la pantalla principal de la misma, a la que se accede una vez está realizado el log-in, y en la cual podemos observar el menú de navegabilidad desplegado, a través del cual se puede acceder a la sección de amigos, a la sección de configuración, a la sección de invitaciones, y realizar log-out de la aplicación.

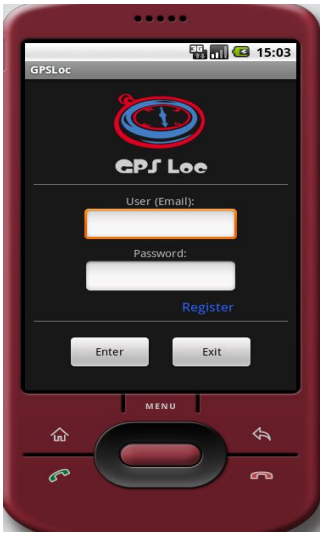

Figura 18 – Menú inicial

Figura 19 – Menú

A continuación, en la Figura 20, se puede observar un ejemplo de asociación maestro-detalle. En la parte izquierda se muestra la lista de amigos que tiene el usuario logueado. Seleccionando a "Carlos", se accede a la interfaz mostrada en la parte derecha, el detalle del amigo, en el cual pueden observarse los detalles del usuario, las características del permiso que se tiene sobre él y su posición en el mapa. Además, esta interfaz ofrece la posibilidad de actualizar la posición del usuario, ver sus últimas posiciones, llamarle por teléfono, o eliminar el amigo en cuestión.

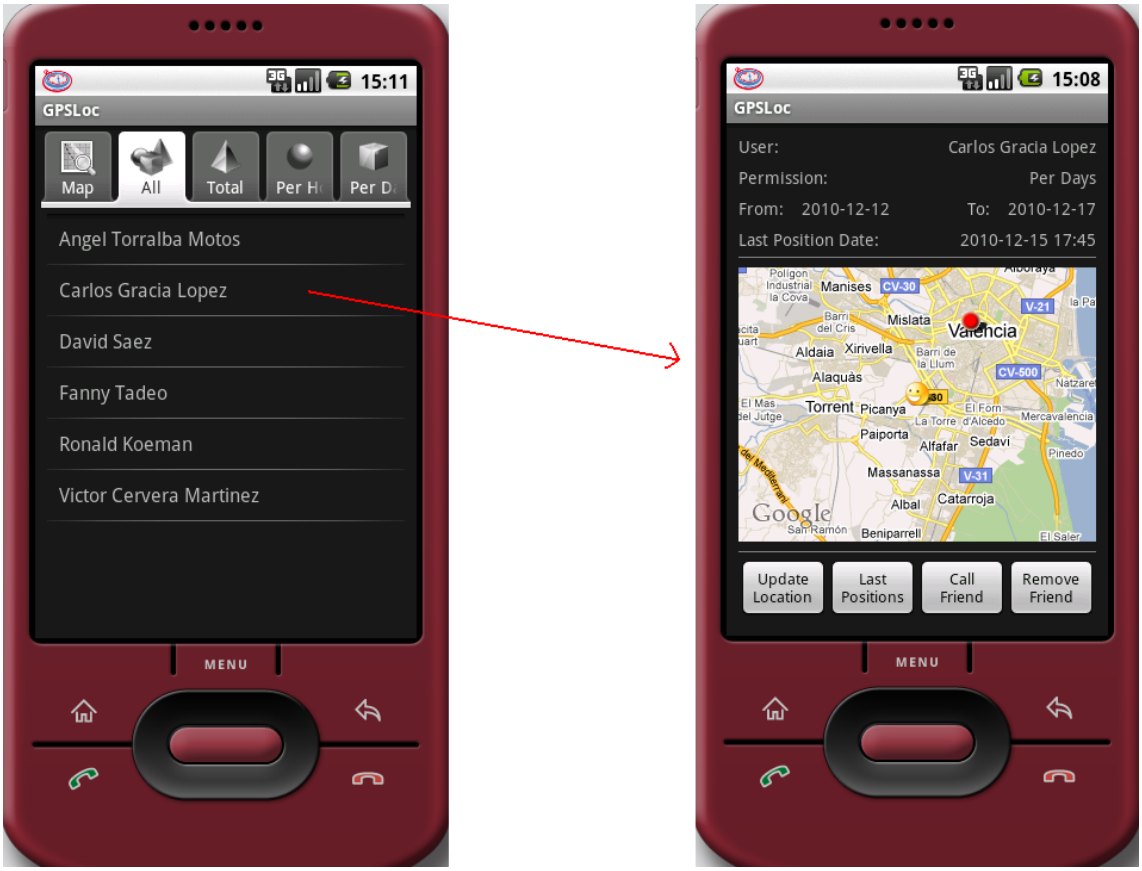

Figura 20 – Ejemplo Maestro-Detalle

2.3.1.2. INTERFAZ HARDWARE

GPSLoc depende de un servidor remoto, en el que se alojan la BD Remota y una capa ADO de acceso a los datos implementada mediante JSON y ficheros PHP, que hacen de puente entre GPSLoc y la BD Remota. A continuación se describen los detalles del servidor remoto:

Tipo de Procesador: Quad Core Virtualizado
Memoria RAM: 6 gb Virtualizados
Velocidad de conexión: Desconocida
Almacenamiento: 300 GB

Debido a que la velocidad de la aplicación depende directamente de la potencia del servidor remoto y del ancho de banda del mismo, la configuración del servidor actual no sería suficiente para ofrecer una escalabilidad óptima. A continuación se describe una posible configuración óptima para el servidor remoto:

Tipo de Procesador: Quad Core Dedicado
Memoria RAM: 8 gb Dedicados
Velocidad de conexión: 10 MB simétricos dedicados
Almacenamiento: 20TB*

*Hay que hacer inciso en que una gran capacidad de almacenamiento sería necesaria para poder asegurar la escalabilidad de la aplicación.

2.3.1.3. INTERFAZ SOFTWARE

Sistema Operativo: Android
Servidor Web: Parallels Plesk
Sistema Operativo Servidor Web: Microsoft Windows 2008 R2

2.3.2. REQUISITOS FUNCIONALES

En este apartado serán descritos los requisitos funcionales de GPSLoc. Se incorporará una captura del prototipo acompañando a las descripciones de las funcionalidades de la aplicación. Esto es algo no contemplado por el estándar IEEE-830, pero facilitará la compresión de la aplicación.

2.3.2.1. Registro

Entrada: Email, Password, Full Name, Mobile Phone.
Proceso:

- Comprobar:
  - Que el campo email no está vacio
  - Que el email no esté ya dado de alta en el sistema
- Dar de alta al usuario en la BD remota

Salidas: Confirmación del registro, advertencia de que el campo email no debe estar vacío o de que ese email ya está registrado.

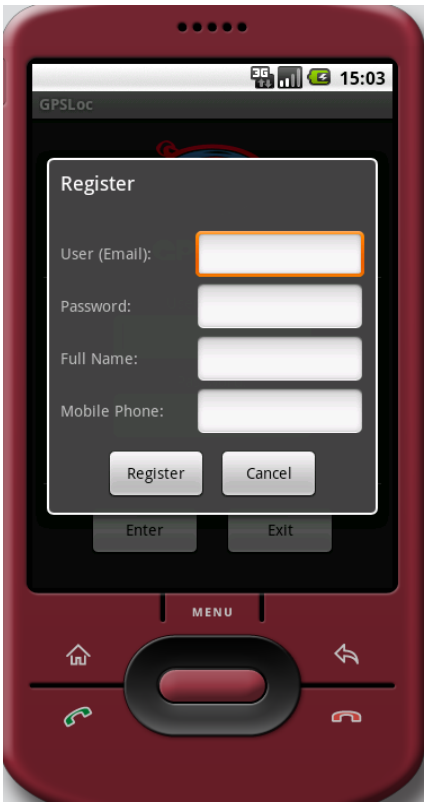

Figura 21 – Panel de registro

2.3.2.2. Log-In

Entrada: Email, Password.

Proceso:

- Cuando el usuario escribe la contraseña, se muestran asteriscos, no letras.
- Comprobar:
  - Que el GPS esté activado.
  - Que el campo Email no esté vacio.
  - Que el Email esté dado de alta en el sistema.
  - Que la contraseña es correcta
- Descargar los amigos y permisos del usuario, y almacenarlos en la BD Local.
- Situar el icono de la aplicación en la barra de estado.
- Iniciar el servicio que actualiza la barra de estado con notificaciones pendientes (explicado más adelante).
- Iniciar el servicio que lee la posición GPS actual y la envía al servidor remoto (explicado más adelante).
- Iniciar el servicio que actualiza la BD Local (explicado más adelante).
- Mostrar el mapa de amigos.

Salidas: Aviso de si el usuario no existe, o la contraseña es incorrecta. Además, si el GPS no está activado, insta a activarlo (explicado más adelante).

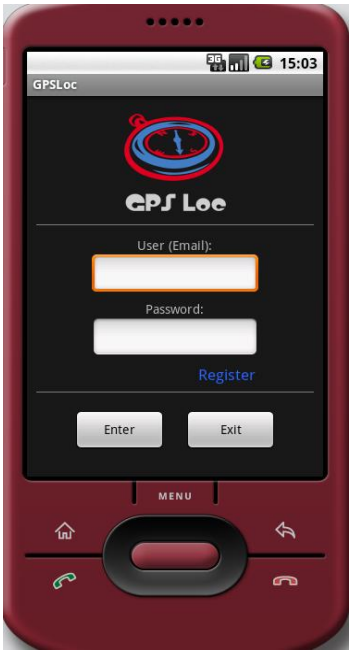

Figura 22 – Menú inicial

2.3.2.3. Activación del GPS

Entrada: Email, Password

Proceso:

- Redirige al usuario a la configuración del GPS, para poder activarlo (no es posible activarlo mediante código en Android)

Salidas: GPS Activado

Figura 23 – Mensaje GPS desactivado

2.3.2.4. Notificación de invitaciones pendientes

Entrada: Email

Proceso:

- Obtener invitaciones pendientes de la BD Remota
- Comprobar
    - Si hay invitaciones pendientes nuevas
- Mostrar el número de invitaciones pendientes en la barra de estado
- Si hay invitaciones nuevas, notificarlo mediante un mensaje
- Repetir el proceso cada 5 minutos, mediante un servicio en background.

Salidas: Aviso en la barra de estado cuando se recibe una nueva invitación pendiente.

Figura 24 – Status Bar con invitaciones pendientes

2.3.2.5. Visualizar amigos en el mapa

Entrada: Email.

Proceso:

- Obtener los amigos y permisos de la BD Local.
- Obtener las posiciones de los amigos de la BD remota.
- Filtrar las posiciones que cumplan el permiso que se tiene sobre cada amigo
- Visualizar en el mapa la última posición de los amigos, para la cual se cumpla el permiso que se tiene sobre cada amigo.
- Obtener la posición GPS actual del usuario logueado y mostrarla en el mapa mediante un punto rojo, si así está indicado en la configuración de la aplicación.
- Centrar el mapa en la posición actual del usuario.

Salidas: Posiciones de los amigos y posición actual del usuario.

Figura 25 – Mapa de amigos

2.3.2.6. Visualizar todos los amigos en una lista

Entrada: Email.

Proceso:

- Obtener los amigos de la BD Local.

Salidas: Listado de amigos.

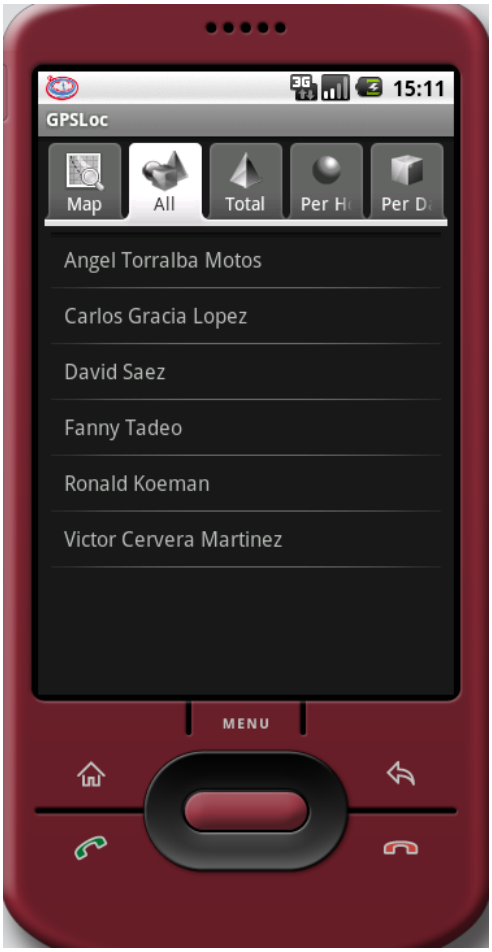

Figura 26 – Listado de todos los amigos

2.3.2.7. Visualizar amigos con permiso total en una lista

Entrada: Email.

Proceso:

- Obtener los amigos de la BD Local.
- Filtrar los amigos por permiso Total.

Salidas: Listado de amigos con permiso Total.

Figura 27 – Listado de amigos con permiso total

2.3.2.8. Visualizar amigos con permiso por horas en una lista

Entrada: Email.

Proceso:

- Obtener los amigos de la BD Local.
- Filtrar los amigos por permiso Por Horas.

Salidas: Listado de amigos con permiso Por Horas.

Figura 28 – Listado de amigos con permiso por horas

2.3.2.9. Visualizar amigos con permiso por días en una lista

Entrada: Email.

Proceso:

- Obtener los amigos de la BD Local.
- Filtrar los amigos por permiso Por Días.

Salidas: Listado de amigos con permiso Por Días.

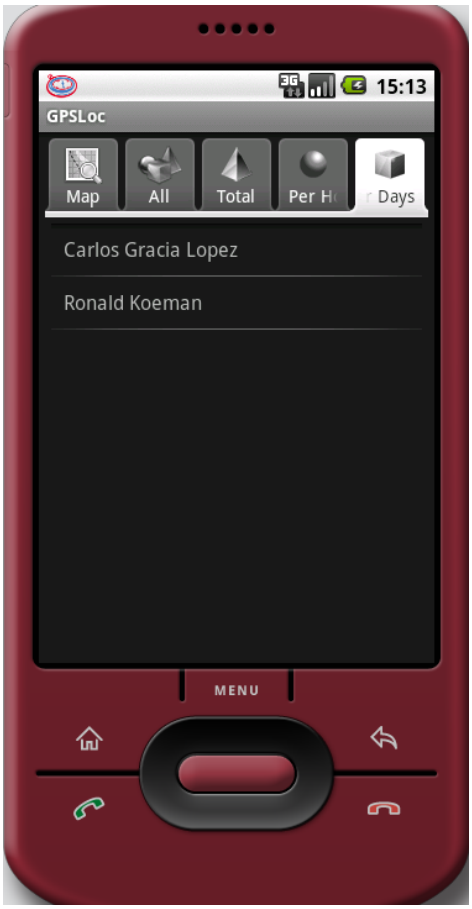

Figura 29 – Listado de amigos con permiso por días

2.3.2.10. Mostrar cuadro de información de un amigo en el mapa

Entrada: Email del amigo, Nombre completo del amigo, Permiso del amigo.
Proceso:

- Comprobar:
  - Si la foto del amigo está almacenada temporalmente en local.
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas.

Salidas: Cuadro de información con el nombre, el email, el permiso y la foto de perfil del amigo, además de un botón para acceder a su información detallada.

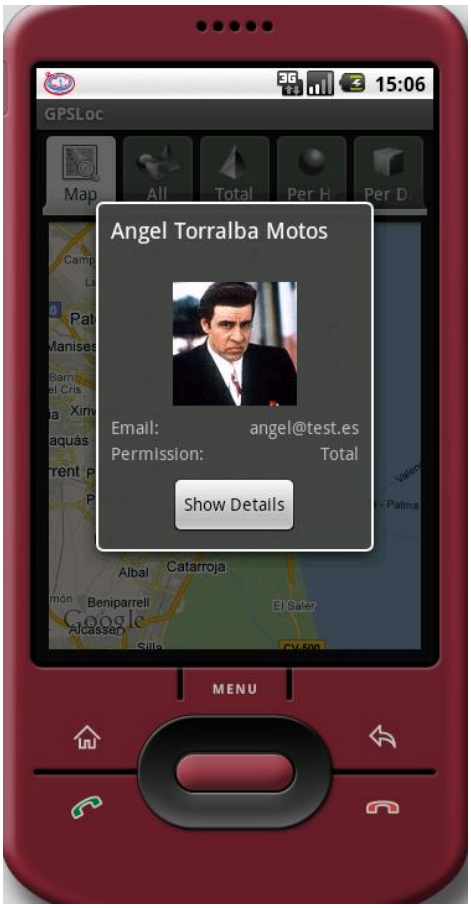

Figura 30 – Cuadro de información del amigo

2.3.2.11. Vista detallada de un amigo con permiso total

Entrada: Email, Nombre completo, Permiso.
Proceso:

- Obtener última posición del amigo de la BD Remota.
- Obtener última posición GPS del usuario logueado y mostrarla en el mapa si así está indicado en la configuración.

Salidas: Vista detallada del amigo, incluyendo el tipo de permiso, el nombre completo, la fecha de la última posición y su posición visualizada en el mapa.

Figura 31 – Vista detallada de amigo (per. total)

2.3.2.12. Vista detallada de un amigo con permiso por horas

Entrada: Email, Nombre completo, Permiso.
Proceso:

- Obtener las posiciones del amigo de la BD Remota.
- Obtener el Permiso de la BD Local.
- Para cada posición, comprobar:
    - Si la hora de la posición está entre el rango de horas del permiso
- Mostrar la última posición que cumpla el permiso.
- Obtener última posición GPS del usuario logueado y mostrarla en el mapa si así está indicado en la configuración.

Salidas: Vista detallada del amigo, incluyendo el tipo de permiso, las características del permiso (desde qué hora y hasta qué hora es válido), el nombre completo, la fecha de la última posición válida, y esa misma posición visualizada en el mapa.

Figura 32 - Vista detallada de amigo (per. por horas)

2.3.2.13. Vista detallada de un amigo con permiso por días

Entrada: Email, Nombre completo, Permiso.
Proceso:

- Obtener las posiciones del amigo de la BD Remota.
- Obtener el Permiso de la BD Local.
- Para cada posición, comprobar:
    - Si la fecha de la posición está entre el rango de días del permiso
- Mostrar la última posición que cumpla el permiso
- Obtener la última posición GPS del usuario logueado y mostrarla en el mapa si así está indicado en la configuración.

Salidas: Vista detallada del amigo, incluyendo el tipo de permiso, las características del permiso (desde que día y hasta que día es válido), el nombre completo, la fecha de la última posición válida, y esa misma posición visualizada en el mapa.

Figura 33 - Vista detallada de amigo (per. por días)

2.3.2.14. Mostrar últimas posiciones de un amigo

Entrada: Email.

Proceso: (Una vez pulsado el botón Last Positions en la ventana Detalles de amigo)

- Obtener las posiciones del amigo de la BD Remota.
- Obtener el permiso de la BD Local.
- Filtrar las posiciones que cumplan con el permiso.

Salidas: Sitúa en el mapa las últimas posiciones válidas del amigo. Además, pulsando en cada una de ellas, se puede observar la fecha de la posición.

Figura 34 – Últimas posiciones de un amigo

2.3.2.15. Llamar por teléfono a un amigo

Entrada: Email amigo.

Proceso: (una vez pulsado el botón Call Friend en la ventana Detalles de amigo)

- Lanzar el proceso de Android para realizar una llamada telefónica.

Salidas: Muestra la llamada telefónica en curso.

Figura 35 – Vista detallada de un amigo

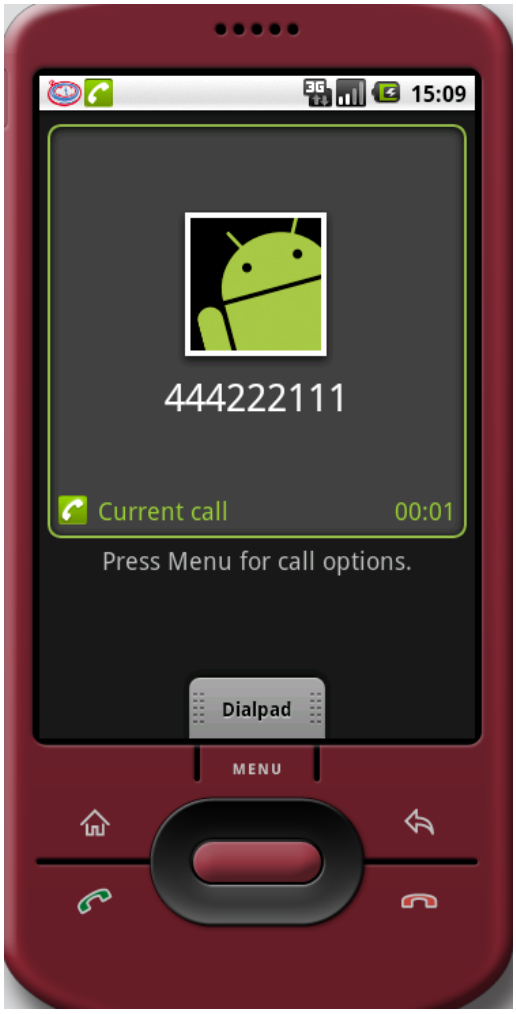

Figura 36 – Llamada telefónica en curso

2.3.2.16. Eliminar amigo

Entrada: Email amigo.
Proceso: (Una vez pulsado el botón Remove Friend en la ventana detalles de amigo)

- Eliminar el amigo de la BD Remota y la BD Local
- Eliminar el permiso de la BD Remota y la BD Local

Salidas: Advertencia al usuario de si está seguro de realizar la operación.

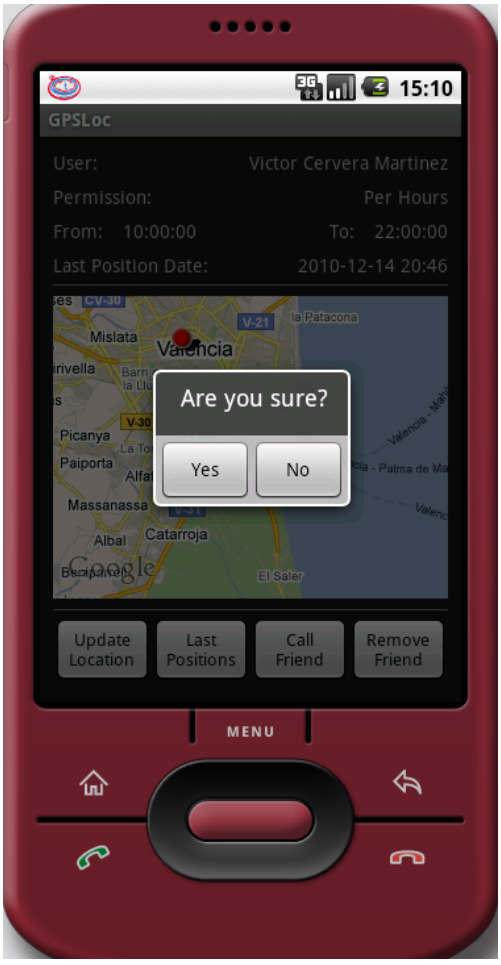

Figura 37 – Mensaje de confirmación para borrar amigo

2.3.2.17. Configuración de la aplicación

Entrada: Email.

Proceso:

- Obtener usuario de la BD Local
- Comprobar:
  - Si la foto del usuario está almacenada en local.
  - Si el usuario ha configurado la aplicación para mostrarse en el mapa
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas cercanas.
- Si el usuario ha configurado la aplicación para mostrarse en el mapa, marcar la casilla "Show my GPS position".
- Se le permite al usuario cambiar su nombre completo, número de teléfono, contraseña, elegir si mostrarse en el mapa a sí mismo y seleccionar una imagen de perfil.

Salidas: Se muestra la información del usuario logueado, incluyendo Email, nombre completo, número de teléfono, y si ha configurado o no que su posición se muestre en los mapas. Además se muestra la imagen de perfil del usuario.

Figura 38 - Configuración

2.3.2.18. Cambiar la imagen de perfil de un usuario

Entrada: Email amigo.

Proceso: (Una vez pulsado el botón Change Image en la ventana de configuración)

- Visualizar las imágenes almacenadas en el dispositivo móvil y permitir al usuario seleccionar una mediante el visualizador de imágenes de Android.
- Comprobar:
  - Que el tamaño de la foto sea igual o menor que 100KB
- Una vez ha seleccionada una foto con éxito, guardar la foto en local y en la BD Remota.

Salidas: Muestra la nueva foto en el panel de configuración.

Figura 39 - Configuración

Figura 40 – Selector de imágenes

2.3.2.19. Mostrar posición del usuario en el mapa en tiempo real

Entrada: Email usuario.

Proceso:

- Mediante un servicio en background, se lee la posición GPS actual cada 10 segundos y se guarda en local mediante SharedPreferences, una variable de Android compartida entre todas las clases de la aplicación.
- Cada minuto, se envía la posición a la BD Remota.
- En los mapas de la aplicación, cada 15 segundos, se obtiene la posición GPS de SharedPreferences y se muestra en el mapa mediante un punto rojo, si el usuario ha activado esta opción en la configuración de la aplicación.

Salidas: Visualización en los mapas de un punto rojo indicando la posición actual del usuario.

Figura 41 – Mapa de amigos

2.3.2.20. Buscar usuarios

Entrada: Secuencia de texto.
Proceso: (Una vez introducida una secuencia de texto para buscar y pulsando el botón Search en la ventana invitaciones).

- Buscar todos los usuarios cuyo nombre contenga el texto introducido. Es decir, si se introduce la cadena de texto "*bl*", aparecerán todos los nombres que contengan la secuencia de letras "bl", como por ejemplo: Pablo Sáez, Pablo Ruiz, etc.

Salidas: Lista con los nombres de usuario encontrados.

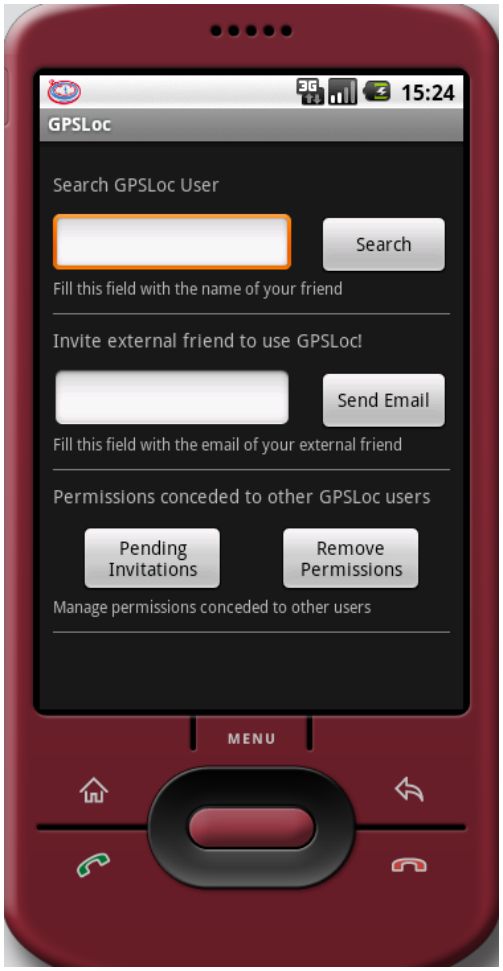

Figura 42 - Invitaciones

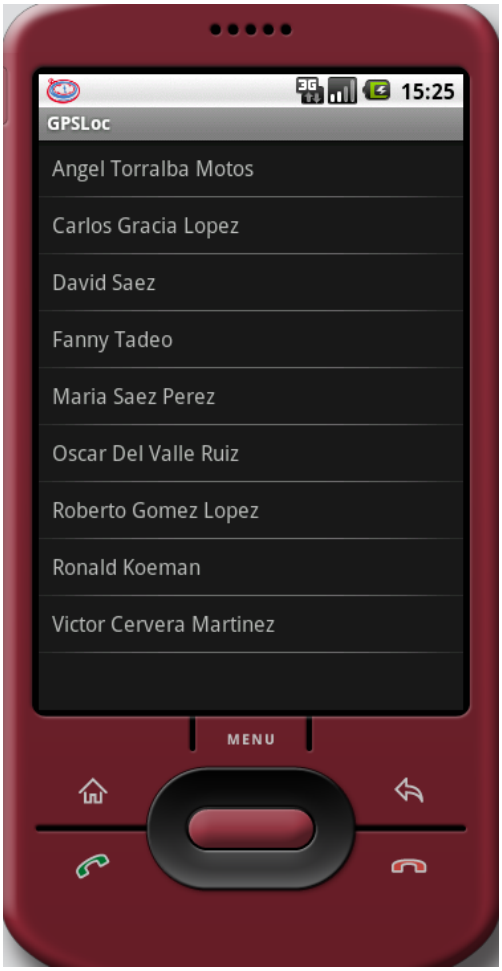

Figura 43 – Resultados de búsqueda de usuario

2.3.2.21. Enviar invitación a un usuario (permiso total)

Entrada: Email, Nombre completo.
Proceso: (Una vez seleccionado un usuario en los resultados de la búsqueda de usuarios)

- Comprobar:
  - Si la foto del usuario está almacenada en local
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas.
- Insertar el permiso en la BD con el campo "validated" a cero. De esta forma queda como invitación pendiente de aceptar por el usuario receptor.

Salidas: Se muestra una confirmación de que la invitación ha sido realizada con éxito.

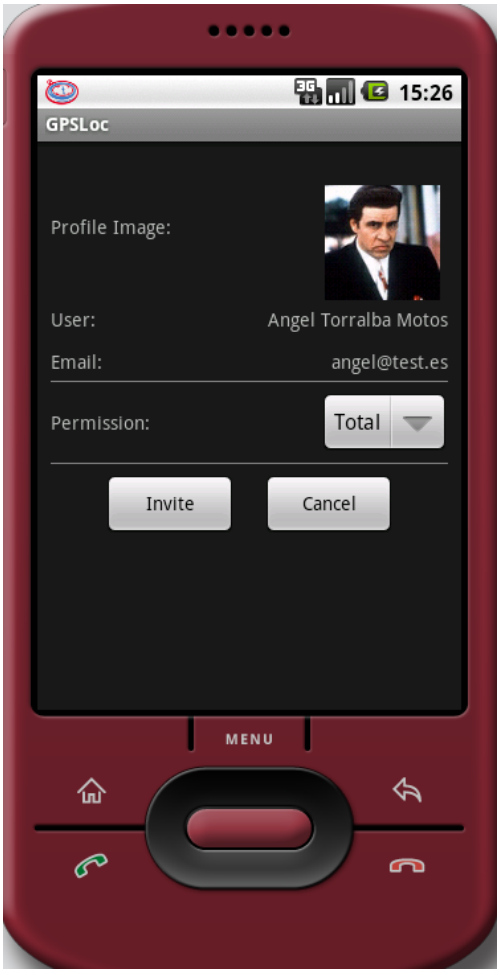

Figura 44 – Invitación permiso total

2.3.2.22. Enviar invitación a un usuario (permiso por horas)

Entrada: Email, Nombre completo.

Proceso: (Una vez seleccionado un usuario en los resultados de la búsqueda de usuarios).

- El usuario logueado selecciona un usuario de los resultados de búsqueda.
- Comprobar:
  - Si la foto del usuario está almacenada en local
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas.
- Comprobar:
  - Que la hora inicial es inferior a la hora final.
- Insertar el permiso en la BD con el campo "validated" a cero. De esta forma queda como invitación pendiente de aceptar por el usuario receptor.

Salidas: Se muestra una confirmación de que la invitación ha sido realizada con éxito.

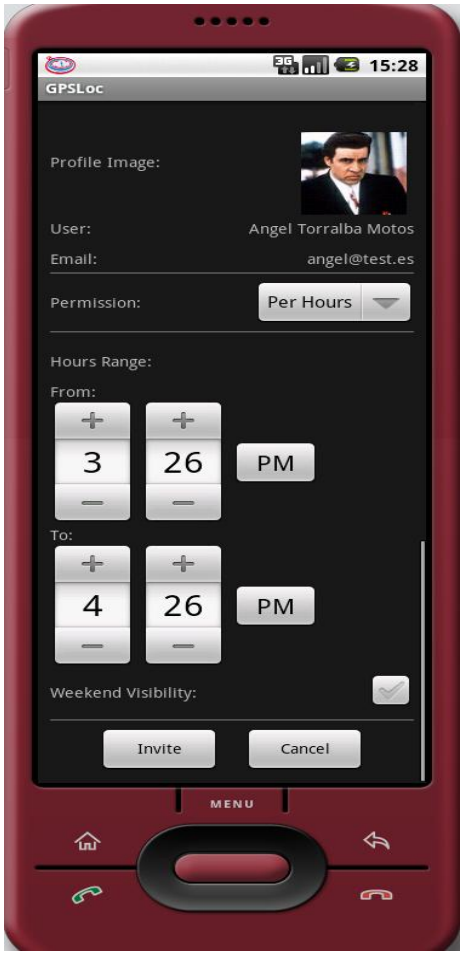

Figura 45 – Invitación permiso por horas

2.3.2.23. Enviar invitación a un usuario (permiso por días)

Entrada: Email, Nombre completo.

Proceso: (Una vez seleccionado un usuario en los resultados de la búsqueda de usuarios).

- El usuario logueado selecciona un usuario de los resultados de búsqueda.
- Comprobar:
  - Si la foto del usuario está almacenada en local
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas.
- Comprobar:
  - Que la fecha inicial es inferior a la fecha final.
- Insertar el permiso en la BD con el campo "validated" a cero. De esta forma queda como invitación pendiente de aceptar por el usuario receptor.

Salidas: Se muestra una confirmación de que la invitación ha sido realizada con éxito.

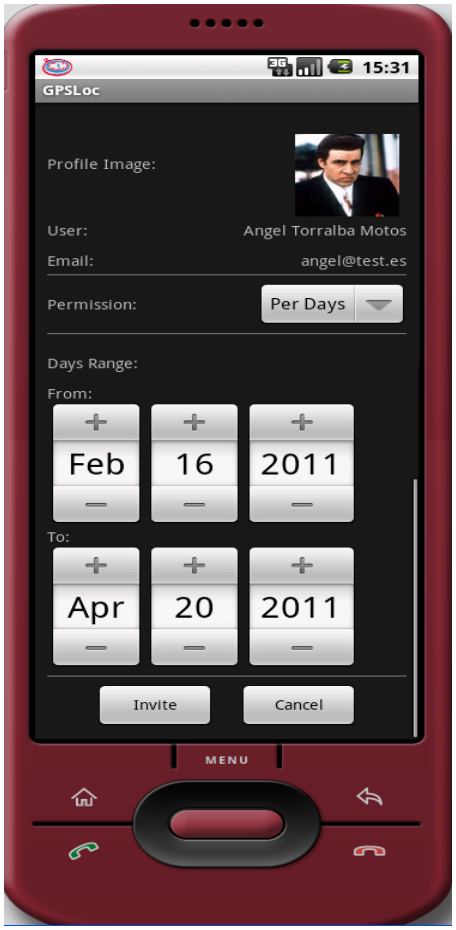

Figura 46 – Invitación permiso por días

2.3.2.24. Enviar un correo de invitación a una persona externa a GPSLoc

Entrada: Email.

Proceso: (Una vez introducido un email y pulsando el botón Send email en la ventana invitaciones).

- Se lanza el gestor de correo por defecto de Android con el correo receptor indicado y con un título y cuerpo de mensaje por defecto, de invitación para usar GPSLoc. El usuario puede editar tanto el título como el cuerpo, para personalizarlo si así lo desea, antes de confirmar el envío del email.

Salidas: El usuario receptor recibe el email de invitación.

Figura 47 – Envío de invitación mediante email

2.3.2.25. Aceptar o rechazar una invitación recibida

Entrada: Email, Nombre completo.
Proceso: (Una vez pulsado el botón Pending Invitations en la ventana Invitaciones, y seleccionada una de las invitaciones pendientes).

- Comprobar:
  - Si la foto del usuario está almacenada en local.
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas.
- Comprobar el tipo de permiso:
  - Si es total, simplemente mostrar el tipo de permiso.
  - Si es por horas, mostrar hora inicial, hora final, fin de semana.
  - Si es por días, mostrar fecha inicial, fecha final.
- Si el usuario la acepta, el campo Validated del permiso queda a 1.
- Si el usuario la rechaza, el permiso queda borrado de la BD Remota.

Salidas: La invitación queda validada, o en caso de ser rechazada, es borrada.

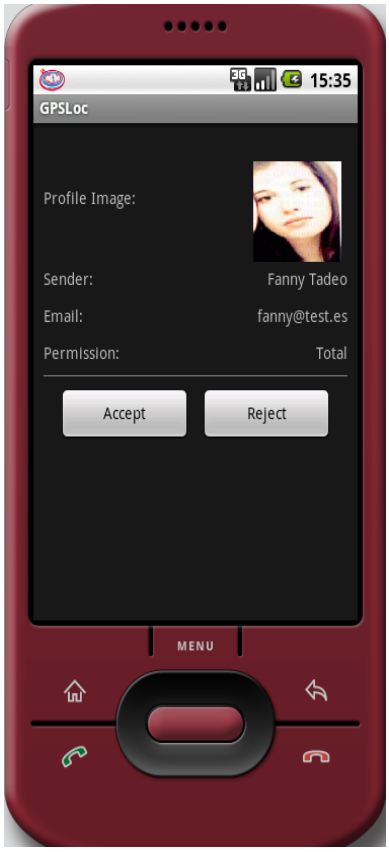

Figura 48 – Invitación pendiente

2.3.2.26. Eliminar permisos concedidos a otros usuarios

Entrada: Email, Nombre completo.
Proceso: (una vez pulsado el botón Remove Permissions en la ventana Invitaciones, y seleccionado uno de los permisos concedidos a algún usuario).

- Comprobar:
  - Si la foto del usuario está almacenada en local
- Si tenemos la foto en local, se carga, si no, se descarga de la BD Remota y se almacena en local para futuras lecturas.
- Comprobar el tipo de permiso:
  - Si es total, simplemente mostrar el tipo de permiso.
  - Si es por horas, mostrar hora inicial, hora final, fín de semana.
  - Si es por días, mostrar fecha inicial, fecha final.
- Eliminar el permiso de la BD Remota y de la BD Local.

Salidas: El permiso es eliminado.

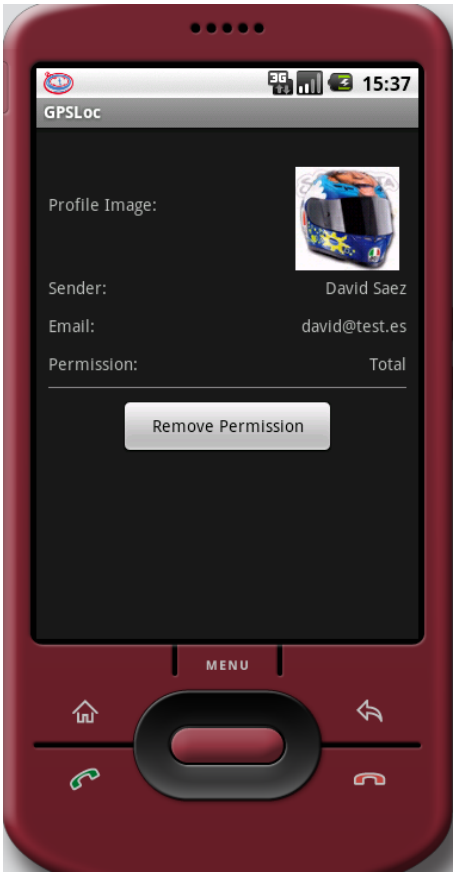

Figura 49 – Permiso concedido

2.3.2.27. Log-Out

Entrada: Ninguna.

Proceso:

- Detener el servicio que actualiza la BD Local.
- Detener el servicio que actualiza la barra de estado con notificaciones nuevas.
- Detener el servicio que lee y almacena la posición GPS.
- Eliminar el icono de la barra de estado.
- Mostrar la pantalla de log-in del programa.

Salidas: Ninguna.

Figura 50 - Menú

2.3.3. REQUISITOS NO FUNCIONALES

En este apartado serán descritos los requisitos no funcionales de GPSLoc.

2.3.3.1. REQUISITOS DEL DESARROLLO

En las aplicaciones móviles, es importante mantener la fluidez de la aplicación, por lo tanto uno de los aspectos más importantes en GPSLoc, es minimizar el máximo posible los tiempos de espera. Para ello, teniendo en cuenta que la aplicación requiere conexión constante a Internet, es recomendable que los usuarios dispongan de conexiones a Internet móvil rápidas, siendo recomendable una conexión 3G. De todas formas, tal y como se ha comentado anteriormente, el mayor handicap encontrado durante el desarrollo de esta tesina, ha sido el tiempo de respuesta del servidor remoto, el cual oscila entre 200 milisegundos y 2 segundos, ralentizando notablemente la aplicación.

Durante el mes de Febrero de 2011, fue realizado un estudio de tráfico de datos entre el servidor remoto y la aplicación, testeando la aplicación en un escenario consistente en un grupo de cinco amigos que manejaron la aplicación para mantenerse localizados entre sí, utilizando GPSLoc una media de 30 minutos diarios. En la siguiente figura puede observarse el mapa de amigos donde el usuario pudo visualizar la posición de sus compañeros (sólo cuatro de ellos participaron en el escenario de prueba).

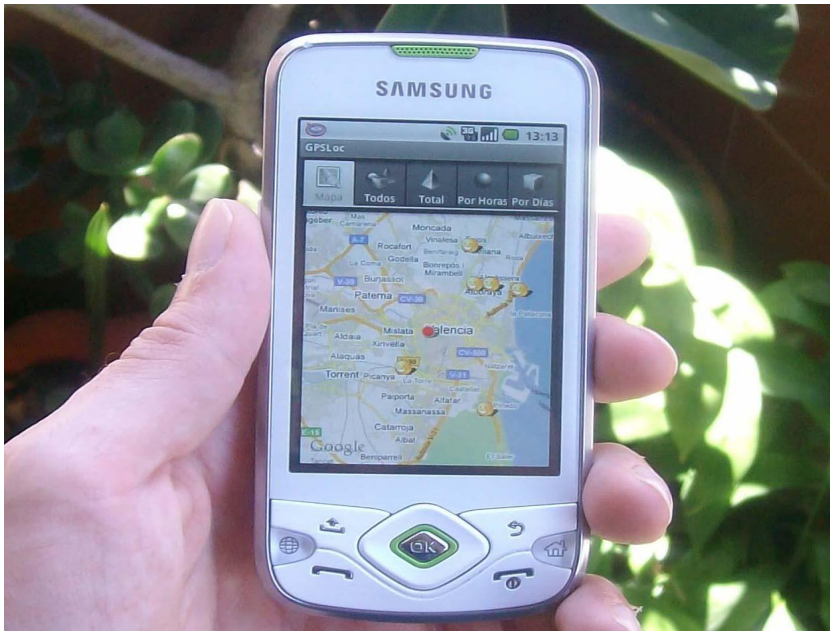

Figura 51 – Mapa de amigos, escenario de prueba

En la siguiente gráfica se puede observar los MB transferidos al día durante ese mes.

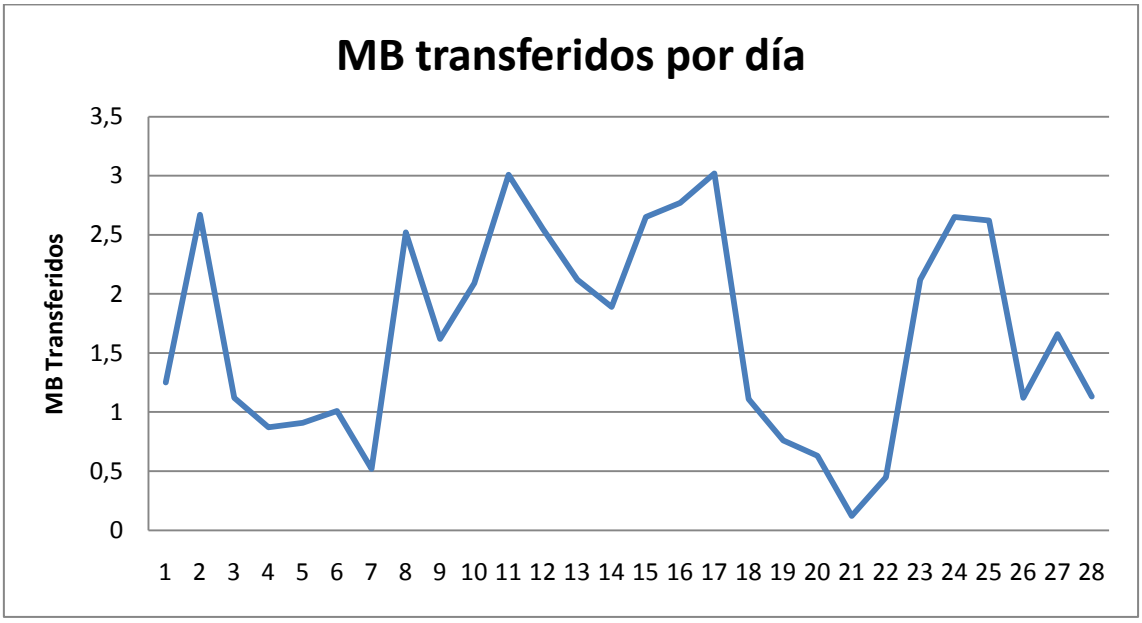

Figura 52 – Estadísticas de MB transferidos por día

El total de KB transferidos entre la aplicación y el servidor remoto por estos cinco usuarios durante el mes de febrero ha sido de 46,96 MBs, lo cual hace una media de 1,67 MBs diarios. En la siguiente figura, se puede observar la proporción entre el tráfico saliente y el tráfico entrante en el servidor. Evidentemente el tráfico saliente es muy superior, ya que el usuario recibe mucha más información de la que envía, a causa principalmente del tamaño de la imagen de perfil de los demás usuarios.

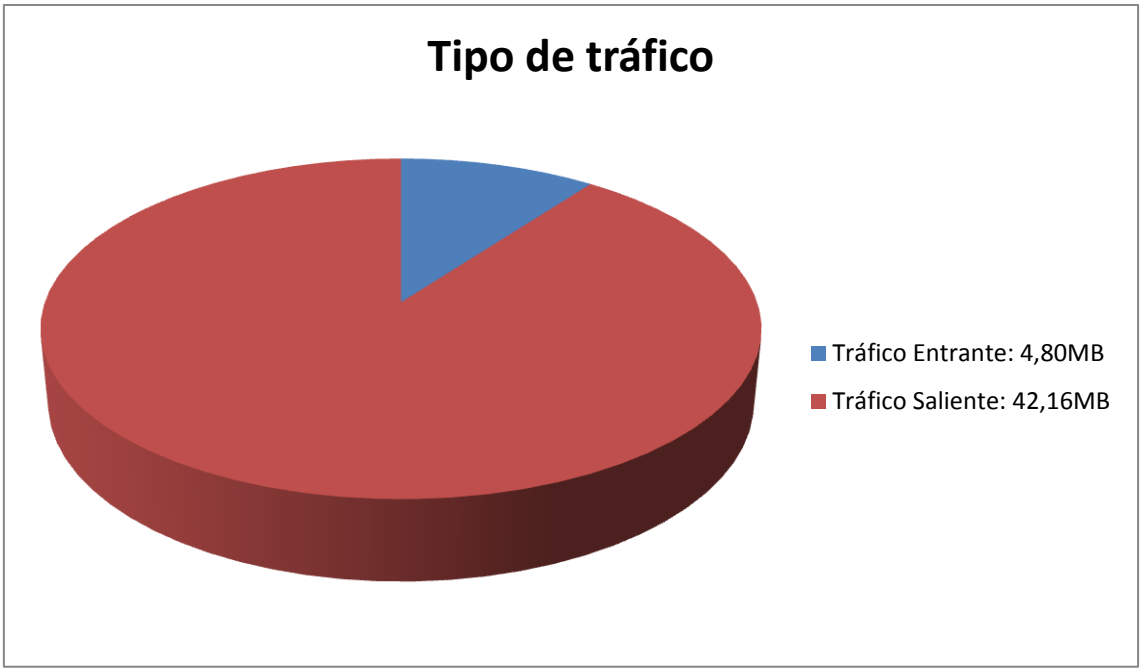

Figura 53 – Estadísticas de tráfico entrante/saliente

Teniendo en cuenta estos datos, es totalmente necesario que en un futuro, para asegurar la escalabilidad de la aplicación, el servidor remoto sea migrado a otro servidor de mayor potencia y velocidad, con el que puedan manejarse grandes volúmenes de información de forma fluida.

2.3.3.2. REQUISITOS DE DISEÑO

La aplicación GPSLoc está diseñada para ofrecer la máxima compatibilidad posible con el mayor número de dispositivos Android del mercado, para ello, se ha decidido que sea compatible con Android 1.5, lo que asegura su compatibilidad con las versiones superiores existentes del sistema, 1.6, 2.0, 2.1, 2.2 y 2.3, debido a que éstas ofrecen retro-compatibilidad completa.

2.3.4. ATRIBUTOS

GPSLoc deberá cumplir, además de los requisitos detallados anteriormente, los siguientes requisitos de carácter no funcional, los cuales son considerados como atributos de la aplicación, es decir, características del proyecto que quedarán como normas para su desarrollo.

- La utilización del sistema y el acceso a los datos y la información, quedarán restringidos mediante mecanismos que permitan la identificación y la autentificación de los usuarios.
- La aplicación estará disponible en cinco idiomas: español, inglés, francés, italiano y alemán.
- Los ficheros PHP deberán estar protegidos contra ataques del tipo SQL Injection y contra el acceso de forma externa a la aplicación, mediante una clave.
- La aplicación deberá ser compatible con terminales que incorporen el sistema operativo Android 1.5 o superior
- El hardware mínimo necesario deberá ser similar al requerido por el sistema operativo Android 1.5.

3. ARQUITECTURA Y TECNOLOGIAS

Durante los apartados anteriores, han quedado detalladas las funcionalidades que ofrece GPSLoc. Durante el capítulo actual y sus apartados, van a ser explicadas la arquitectura y las tecnologías empleadas para construirla y desarrollar sus funcionalidades.

Serán detalladas las capas utilizadas en GPSLoc, analizándolas y situándolas dentro de la arquitectura, además de detallar las tecnologías utilizadas para su elaboración.

3.1. ARQUITECTURA DEL PROYECTO

GPSLoc utiliza una arquitectura de tres capas, interfaz, lógica y persistencia, para lograr una correcta y eficiente separación de contenidos y dependencia entre capas.

Tal y como será explicado en los puntos siguientes, en la capa de interfaz, se encuentran los layouts, ficheros de Android que representan todas y cada una de las interfaces visuales del proyecto, sin funcionalidad. En la capa de lógica se encuentra todo el entramado de funciones y clases que hacen posible la funcionalidad de la aplicación. Finalmente, en la capa de persistencia, están tanto la base de datos local, como la base de datos remota, a la cual es posible conectar mediante un puente PHP+JSON que será explicado más adelante.

A continuación puede observarse de manera gráfica la estructuración de las tres capas de la aplicación.

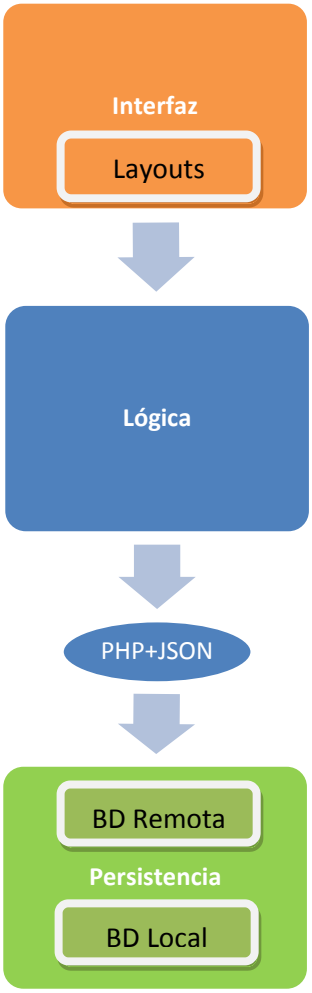

Figura 54 – Arquitectura de 3 capas

3.1.1. INTERFAZ

Mediante la interfaz, GPSLoc muestra toda la información al usuario y permite a éste interactuar con las funcionalidades de la aplicación. Como en todas las aplicaciones para dispositivos móviles, la interfaz está basada en su mayoría en el patrón maestro-detalle. En la figura 55 se puede observar el menú inicial de log-in de la aplicación.

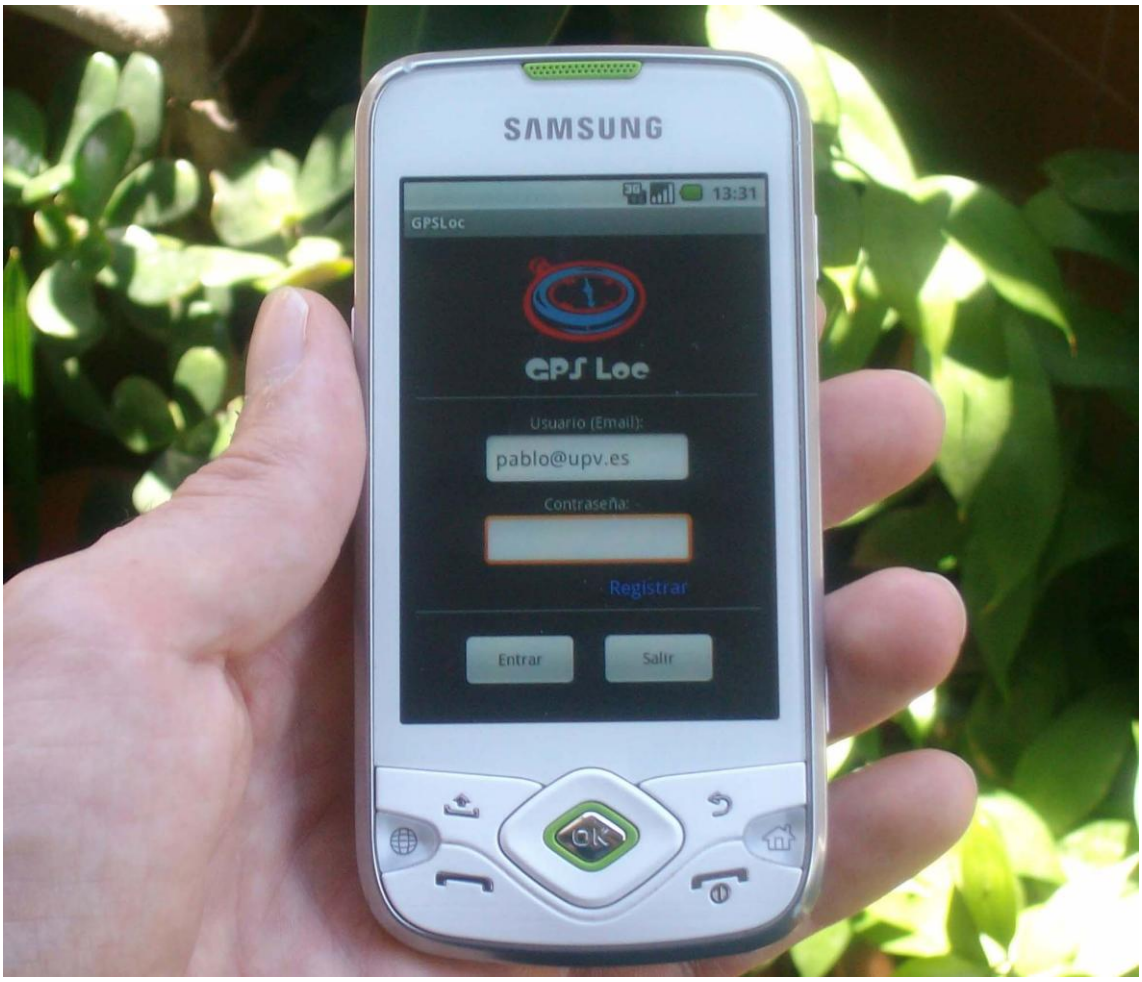

Figura 55 – Menú de log-in

A continuación, en la Figura 56 se observa otro ejemplo de la interfaz, el mapa de amigos, en el cual se pueden visualizar los puntos en los que se encuentran amigos, mediante un icono de una sala sonriente ( ), el punto en el que se encuentra el dispositivo móvil del usuario, mediante un punto rojo ( ), y las pestañas superiores de navegabilidad.

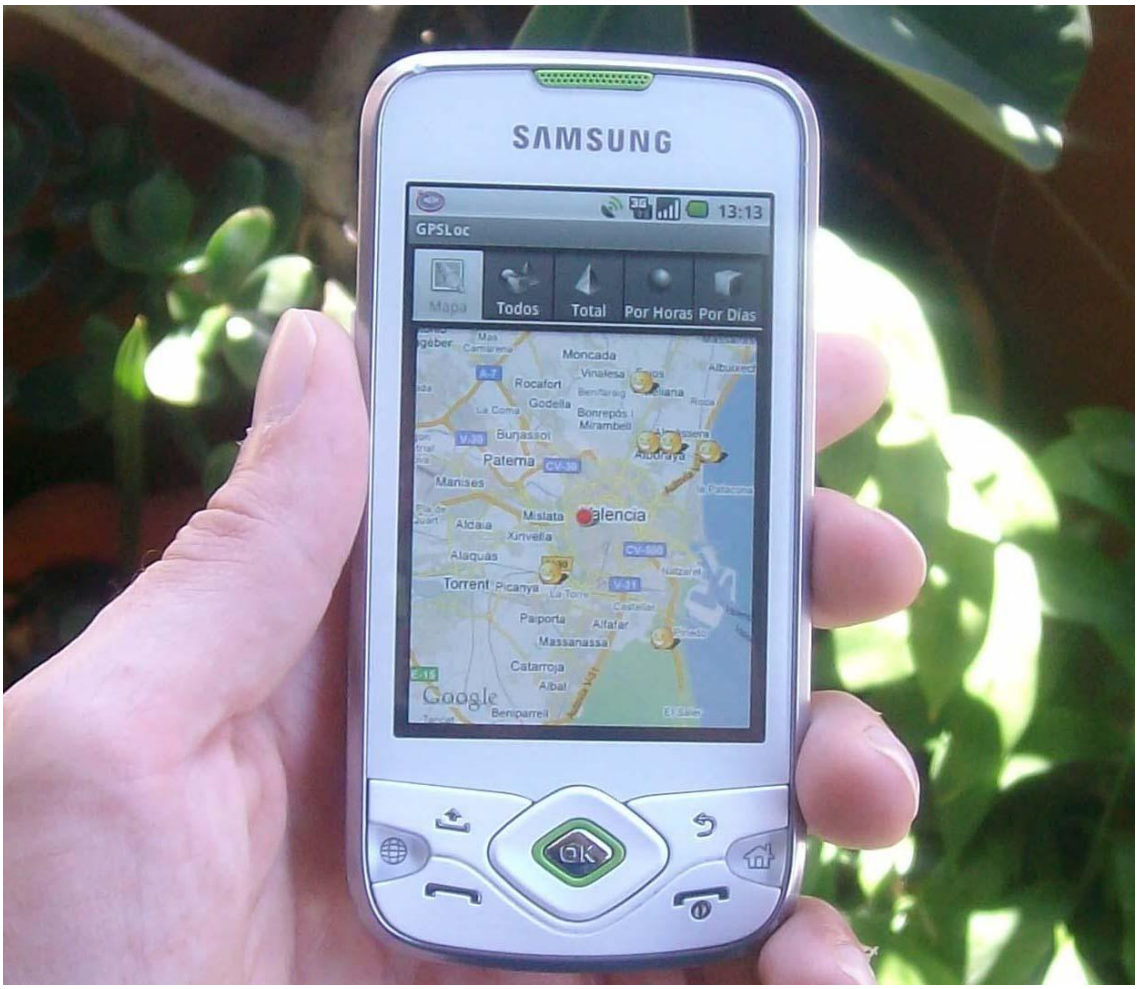

Figura 56 – Mapa de amigos

Las pestañas dan acceso a las siguientes secciones:

-  Muestra el mapa con todos los amigos para los cuales alguna posición cumple el permiso que se posee sobre ellos.
-  Muestra la lista de todos los amigos.
-  Muestra la lista de los amigos sobre los que se posee permiso total.
-  Muestra la lista de los amigos sobre los que se posee permiso por horas.
-  Muestra la lista de los amigos sobre los que se posee permiso por días.

En la Figura 57 puede observarse un ejemplo de asociación maestro-detalle. En la parte izquierda se muestra la lista de amigos que tiene el usuario logueado, seleccionando a "Carlos", se accede a la interfaz mostrada en la parte derecha, el detalle del amigo, en el cual puede observarse los detalles del usuario, las características del permiso que tenemos sobre él y su posición en el mapa. Además, esta interfaz ofrece la posibilidad de actualizar la posición del usuario, ver sus últimas posiciones, llamarle por teléfono, o eliminar el amigo en cuestión.

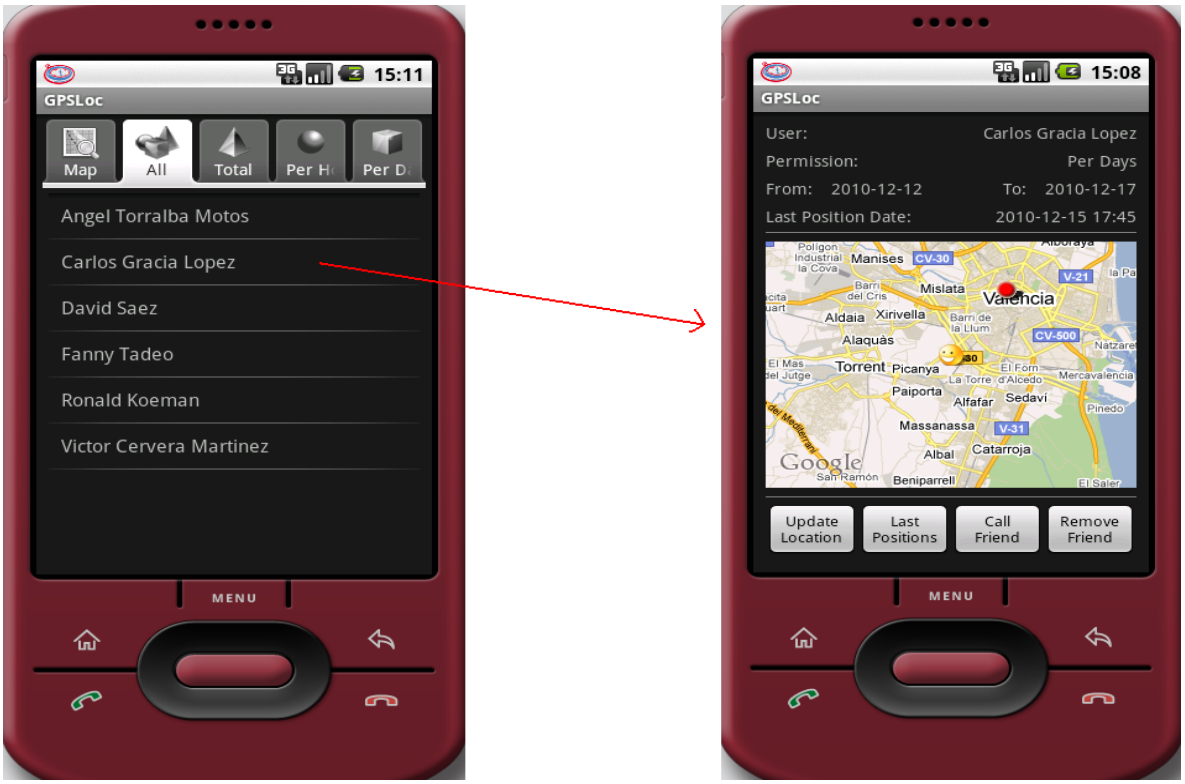

Figura 57 – Ejemplo Maestro-Detalle

La interfaz es la capa más alta de la arquitectura, y ésta tiene por debajo la capa de lógica, la cual se procede a explicar en el siguiente punto.

3.1.2. LOGICA

Esta capa contiene las entidades y el negocio de la aplicación, se encarga de toda la lógica del proyecto y de ella depende el funcionamiento del mismo. Va a ser la encargada de suministrar la información y tratar los datos introducidos en los formularios o producidos por las interacciones del usuario con la aplicación.

Las entidades, contenidas en esta capa, conforman todas las clases, sus constructores, funciones, métodos y procesos. Todas estas clases implementan las funcionalidades de la aplicación explicadas en el punto 2.3.2 de la memoria, por lo que en lugar de repetir la misma información, se ha optado por mostrar el diagrama de clases de la aplicación, el cual se puede observar en la siguiente figura.

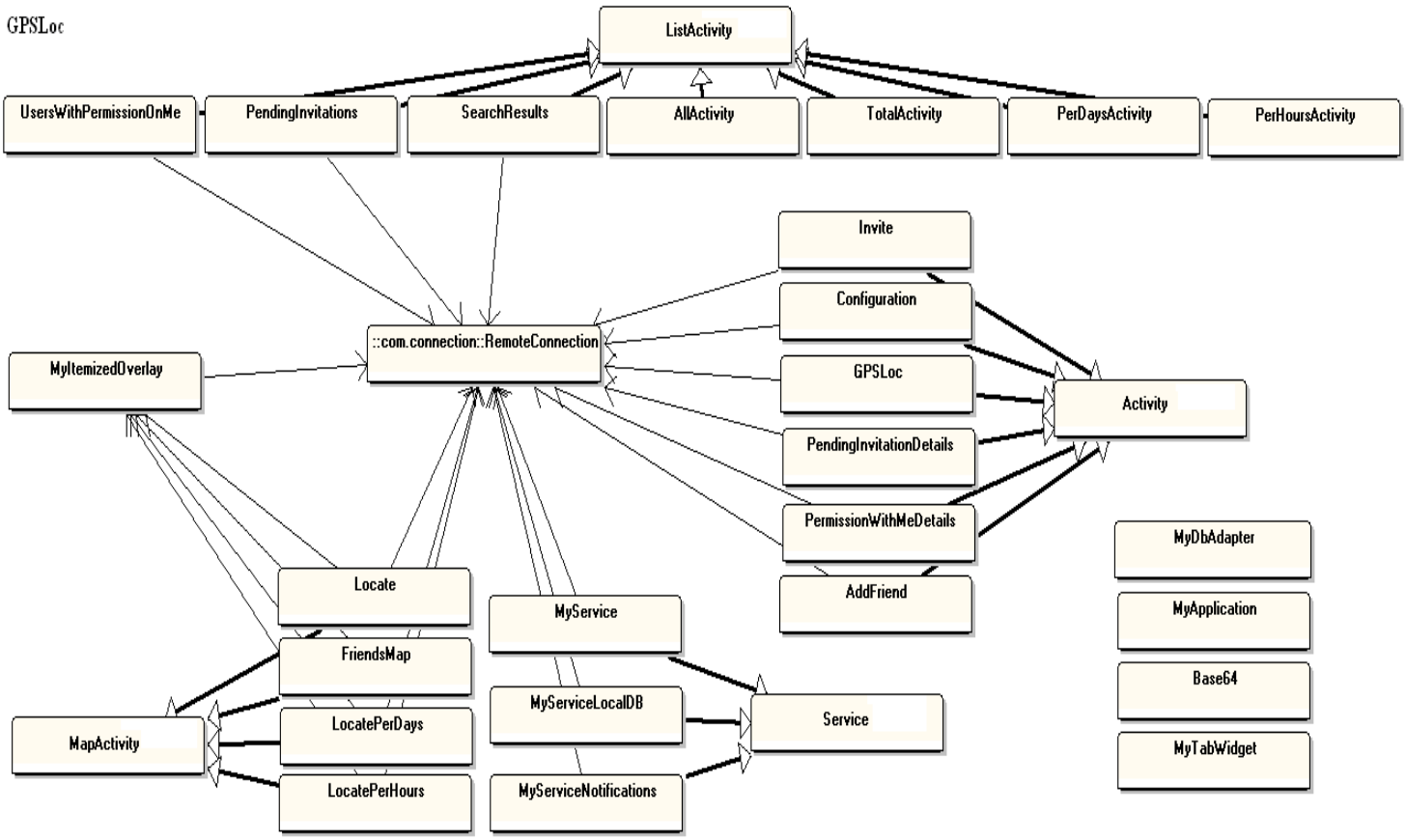

Figura 58 – Diagrama de clases

3.1.3. PERSISTENCIA

Es la última capa de la arquitectura de la aplicación y se encarga de transmitir la información entre la aplicación y la base de datos, guardando, obteniendo y actualizando los registros de la misma.

En Android no existe una forma nativa para acceder a bases de datos remotas. En GPSLoc, para conectar entre la capa de lógica y la capa de persistencia, se ha decidido utilizar un puente que utiliza las tecnologías PHP y JSON para la comunicación [9]. Partiendo de una base de datos MySQL en el servidor remoto, para enviar datos de una tabla a la aplicación, tendremos dos secciones: la parte ejecutada en el servidor remoto y la parte Android. La parte del servidor remoto consiste en un fichero PHP que se conecta a la base de datos, realiza una consulta y devuelve los resultados codificados en formato JSON (la codificación a este formato permite obtener y enviar los datos de forma fluida y rápida). En la parte Android se utiliza un método HttpPost para obtener los datos, se transforma la respuesta a String JSON, y finalmente se transforman los datos JSON en el tipo de datos necesario (String, Int, etc). A continuación, en las siguientes imágenes, podemos observar el puente PHP/JSON utilizado y el esquema entidad-relación de la base de datos remota.

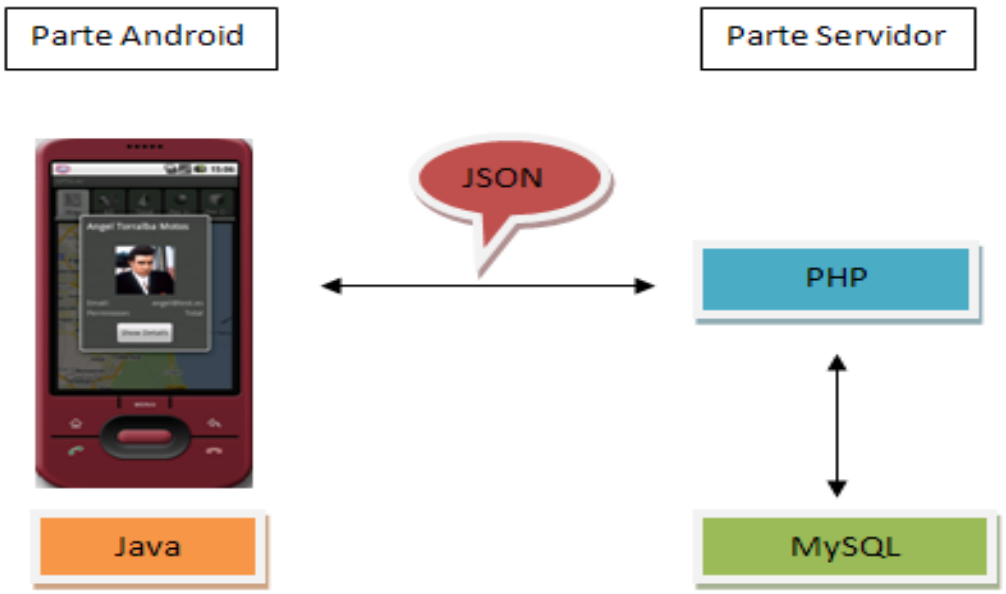

Figura 59 - Puente PHP/JSON entre Android y Servidor remoto

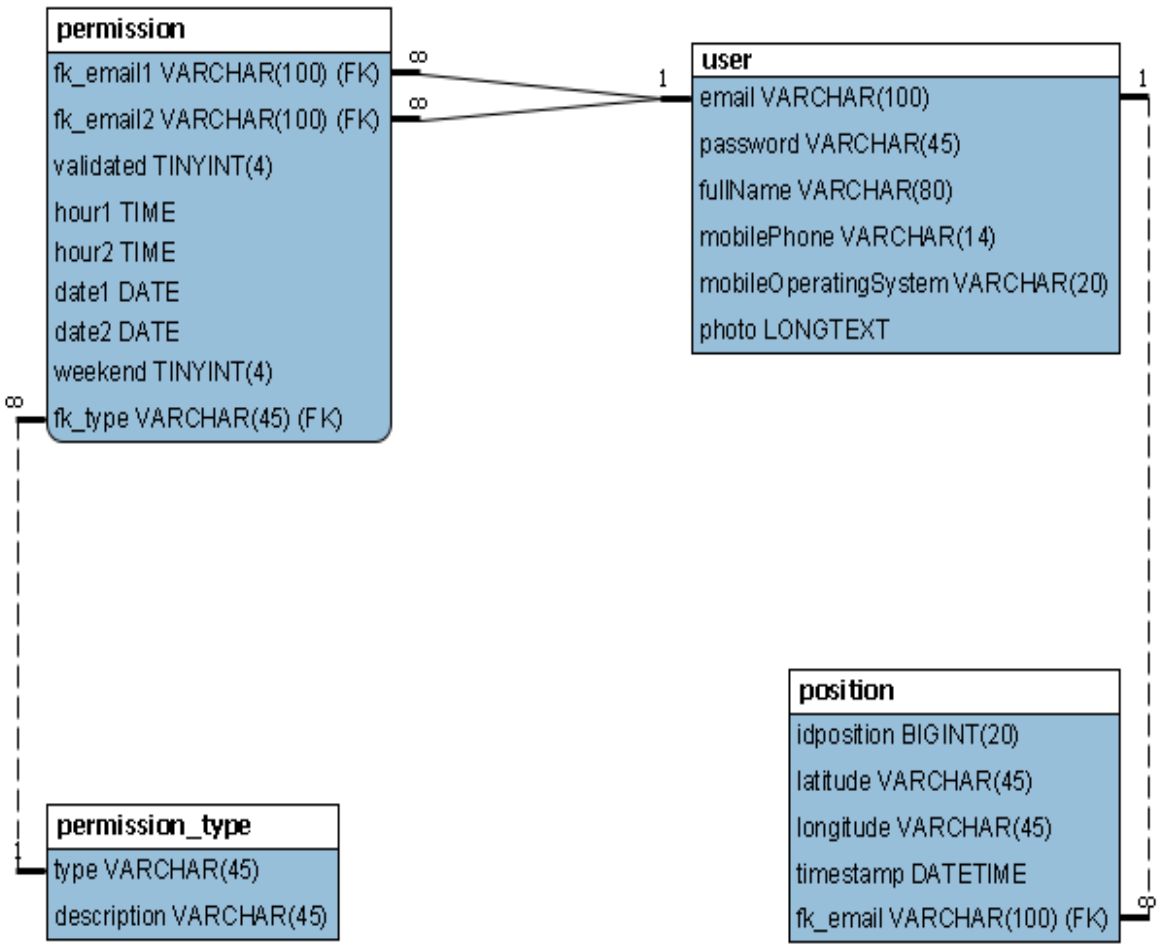

Figura 60 – Diagrama Entidad-Relación

Además, para optimizar la velocidad de la aplicación se ha optado por incorporar una pequeña BD Local SQLite, la cual almacena los amigos sobre los que tiene permiso el usuario y el propio permiso que tiene sobre los mismos. No se ha optado por almacenar en local las posiciones ya que la naturaleza de esta aplicación conlleva poder visualizar la posición de nuestros amigos en tiempo real, y para ello lo mejor es descargar las posiciones cada vez que se muestre al amigo o amigos en un mapa y no cargar posiciones antiguas almacenadas en local.

Para poder tener una base de datos SQLite compartida en toda la aplicación, se ha optado por utilizar una clase Application (MyApplication), la cual inicializa MyDbAdapter, a través del cual se puede acceder a la BD local desde cualquier activity de la aplicación. Las activities de Android son clases con código ejecutable que permiten la interacción con el usuario mediante una interfaz gráfica, de modo que cada activity se podría interpretar como una ventana de la aplicación. Las clases Application tienen la ventaja de ofrecer acceso global (para todas las activities de la App) a las variables que estén almacenando.

Inicialmente se pensó que, para que el usuario, al iniciar la aplicación, no tuviera que esperar demasiado en tiempo de carga, la mejor opción era que en segundo plano, o en un hilo, se descargaran los amigos y los permisos del usuario, para después obtener la posición. Esta estrategia fue descartada debido a que, para poder descargar la posición de los usuarios, es requisito indispensable haber obtenido ya la lista de amigos y sus permisos, por lo tanto, aunque éstos se obtuvieran en segundo plano, no sería posible comenzar a descargar las posiciones mientras la tarea de descargar amigos y permisos no hubiera terminado, con lo que colocar esa tarea en segundo plano seria redundante e inservible.

Finalmente, la estrategia adoptada ha sido la comentada a continuación, la cual da comienzo a partir de que el usuario inicie la aplicación. Se siguen los siguientes pasos:

1. Se vacía la BD local (para evitar redundancias o datos obsoletos).
2. Se descargan las listas actualizadas de amigos y permisos en la BD local.
3. Para cada amigo, se descarga la lista de posiciones.
4. Para cada amigo, se obtiene, según el permiso que se posee sobre él, la última posición que cumple los requisitos del permiso.
5. Se muestra el mapa de amigos con la última posición que cumpla los requisitos del permiso de cada amigo.

A partir de este punto, cada vez que se necesite leer los permisos o los amigos, se hará en local, aumentando considerablemente el rendimiento.

Para mantener actualizada la BD local, cada vez que se borra a un amigo, se borra paralelamente de la BD remota y de la BD local, así como cada vez que se envía una invitación a un usuario que ya es amigo (por ejemplo, para cambiar de tipo de permiso Por Horas a Total), debido a que el permiso sobre ese usuario pasa a ser "no validado", también se procede a actualizar la lista de amigos y permisos local.

Actualizar la BD cuando se envía una invitación a otro usuario quedó descartado, ya que ésta puede ser aceptada pasadas varias horas o días, o incluso nunca, con lo que en ocasiones, la aplicación estaría consumiendo recursos y tiempo del usuario para actualizar la BD sin realizar ningún cambio en la misma. En su lugar, para mantener la BD local actualizada, se ha seguido la siguiente estrategia: en el servicio en background "MyServiceLocalDB", cada cinco minutos, en un hilo se descargará la nueva información actualizada de la BD Remota, para a continuación, mediante un Handler, insertarla en la BD Local. De esta forma, se consigue que el trabajo se realice de forma transparente para el usuario, sin tiempos de espera, ya que la carga pesada que es la obtención de los datos remotos se realiza en el hilo, y la carga suave, que es actualizar la BD Local con la nueva información, se realiza en un Handler, para evitar que otras secciones de la APP puedan tratar de leer de la BD Local mientras ésta se esté actualizando (el Handler aísla las variables de forma que todo lo que quiera acceder a ellas tendrá que esperar a que éste termine).

Otra estrategia que quedó descartada fue la de tener un service en segundo plano que fuera comprobando, cada cierto tiempo, si se habían producido variaciones entre la BD Local y la BD Remota. Esta estrategia consumía prácticamente los mismos recursos que la estrategia anterior, aunque de una forma más compleja, por lo que se optó por descartar esta opción.

3.2. TECNOLOGIAS

En los siguientes apartados de la memoria, serán explicadas las diferentes tecnologías utilizadas durante el desarrollo de la tesina. El entorno de desarrollo elegido es Eclipse, junto con el plug-in ADT para Android. El servidor remoto elegido es Internet Information Server 7.5 y la BD utilizada es MySQL.

3.2.1. ENTORNO DE DESARROLLO: ECLIPSE

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma. Es una plataforma comúnmente usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). La definición que ofrece el proyecto Eclipse acerca de su software es: "un IDE abierto y extensible para todo y nada en particular".

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, abarcando casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse esta desarrollado en la actualidad por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La base para Eclipse es su plataforma de cliente enriquecido (del Inglés Rich Client Platform RCP) [10]. Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plug-ins
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto.
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes.

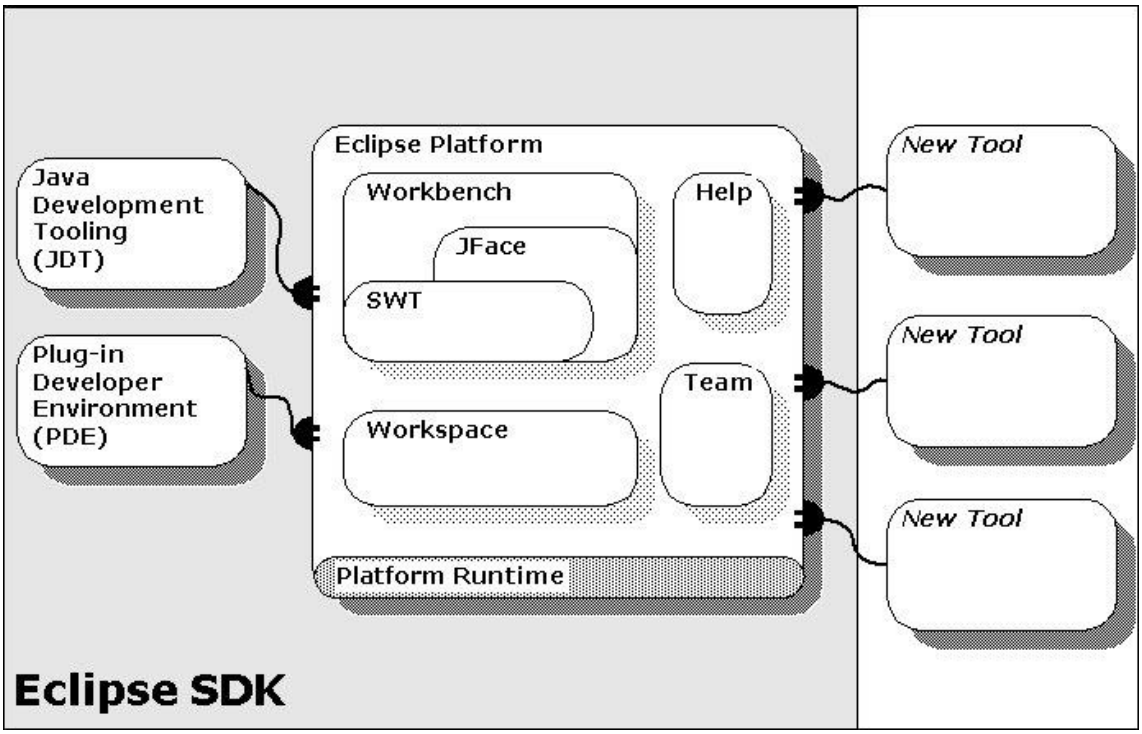

Figura 61 – Arquitectura Eclipse

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. A pesar de estar centrado en Java, Eclipse puede extenderse usando otros lenguajes de programación como son C/C++ y Python, además de trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. Asimismo, a través de plug-ins libremente disponibles, es posible añadir control de versiones con Subversion e integración con Hibernate.

En cuanto a las aplicaciones clientes, eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Por ejemplo, GEF (Graphic Editing Framework - Framework para la edición gráfica) es un plug-in de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto “wysiwyg” hasta editores de diagramas UML, interfaces gráficas para el usuario (GUI), etc. Dado que los editores realizados con GEF "viven" dentro de Eclipse, además de poder ser

usados conjuntamente con otros plugins, hacen uso de su interfaz gráfica personalizable y profesional.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadata en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

En la actualidad existen varias versiones de Eclipse, explicadas a continuación:

- Callisto - En 2006 la fundación Eclipse coordinó sus 10 proyectos de código abierto, incluyendo la Plataforma 3.2, para que sean liberados el mismo día. Esta liberación simultánea fue conocida como la liberación Callisto.
- Europa - La versión consecutiva a Callisto es Europa, que corresponde a la versión 3.3 de Eclipse. Salió el 29 de junio del 2007.
- Ganymede - La versión consecutiva a Europa es Ganymede, que corresponde a la versión 3.4 de Eclipse. Salió el 25 de junio del 2008.
- Galileo - La versión consecutiva a Ganymede es Galileo, que corresponde a la versión 3.5 de Eclipse, con fecha del 24 de junio del 2009.
- Helios - Corresponde a la versión 3.6 de Eclipse y se lanzó el 23 de junio de 2010.

La versión usada para desarrollar GPSLoc es Galileo, junto con el plug-in ADT. Android Development Tools (ADT) es un plug-in para Eclipse diseñado para ofrecer un entorno potente e integrado en el que desarrollar aplicaciones Android.

ADT amplía las capacidades de Eclipse para que pueda configurar rápidamente nuevos proyectos de Android, crear una aplicación de interfaz de usuario, añadir componentes basados en la API Android Framework, depurar sus aplicaciones utilizando las herramientas SDK de Android, e incluso exportar APKs (con o sin firma) para distribuir la aplicación.

El desarrollo en Eclipse con ADT es muy recomendable y es la manera más rápida para empezar. Con la configuración guiada del proyecto que proporciona, así como la integración de herramientas, editores de XML personalizados, y paneles de depuración, ADT ofrece un potente impulso en el desarrollo de aplicaciones para Android.

3.2.2. SERVIDOR REMOTO

El servidor remoto escogido utiliza la tecnología Internet Information Server 7.5 (ISS 7.5). Es un servidor web que fue inicialmente concebido como parte opcional de Windows NT. Posteriormente se integró en otros sistemas operativos destinados a ofrecer servicios, como Windows 2000 y Windows Server 2003. Ofrece diferentes tipos de servicios, tales como FTP, SMTP, MNTP, HTTP y HTTPS.

Internet Information Server proporciona las herramientas y funciones necesarias para administrar un servidor web de forma segura y gestionar las tareas de administración de sites y acceso a maquinas. ISS es además un servidor web basado en módulos que ofrecen la capacidad de procesar diferentes tipos de páginas, como por ejemplo ASP, ASP.NET, PHP, Perl, etc…

3.2.3. SISTEMA GESTOR DE BASE DE DATOS REMOTA

El sistema gestor de base de datos utilizado en la tesina es MySQL. Este es un sistema de gestión de base de datos libre, relacional, multi-hilo y multiusuario utilizado en decenas de millones de proyectos de todo el mundo. Por un lado, se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero aquellas empresas que quieran incorporarlo en productos privados deben comprar una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privadas, la compañía ofrece soporte y servicios.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad

en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y, en cambio, el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Su afinidad con PHP, junto con sus otras ventajas, ha sido lo que ha llevado a tomar la decisión de utilizarlo en el proyecto.

La utilidad seleccionada para manejar la base de datos ha sido PhpMyAdmin, debido a que ofrece potentes herramientas para manejar una base de datos MySQL. PhpMyAdmin es una utilidad escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y se puede utilizar en 62 idiomas. Se encuentra disponible bajo la licencia GPL. Este proyecto está vigente desde el año 1998, siendo de los mejores evaluados en la comunidad de descargas de SourceForge.net.

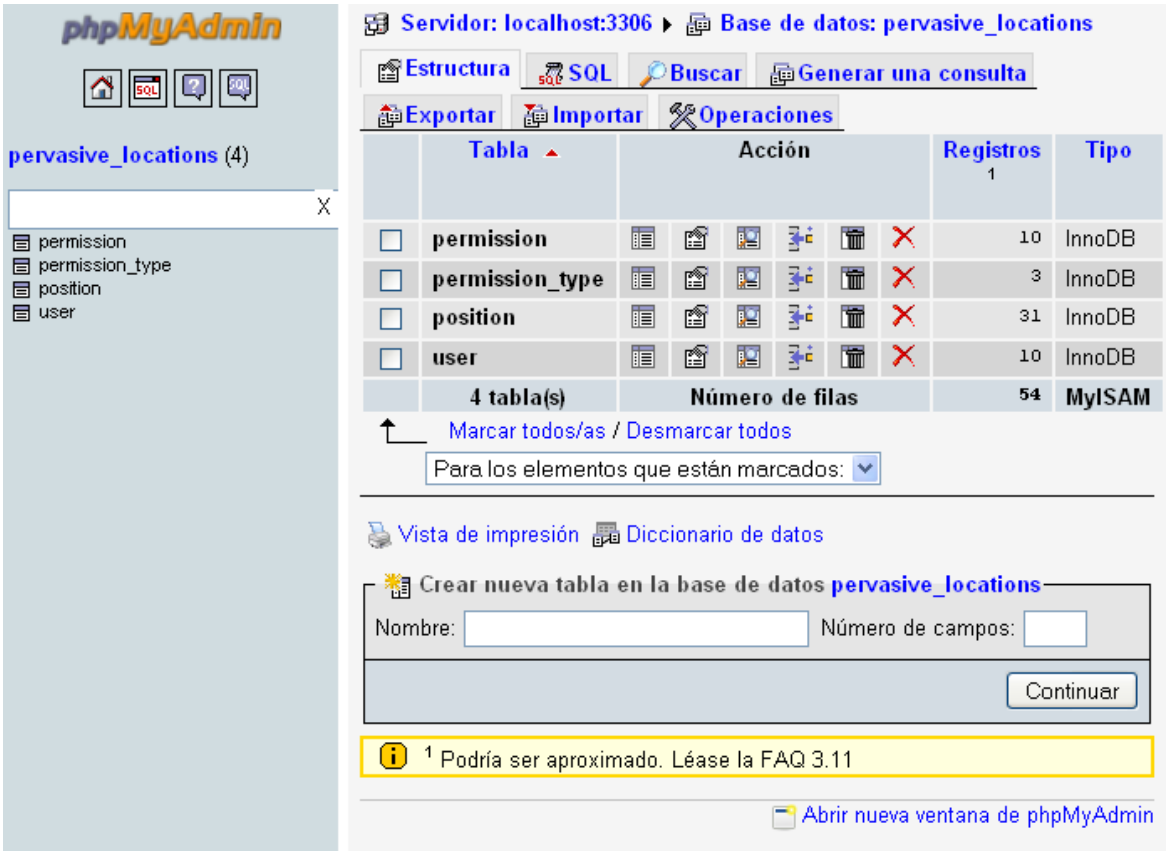

Figura 62 - phpMyAdmin

3.2.4. LENGUAJES UTILIZADOS DURANTE EL PROYECTO

Para completar el desarrollo de la tesina son necesarios cuatro lenguajes diferentes, los cuales serán citados a continuación: Java para la implementación de la aplicación Android, PHP para el puente entre Android y la BD Remota, XML para las interfaces de Android y SQL para la base de datos.

3.2.4.1. JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

El lenguaje Java se creó con cinco objetivos principales:

- Usar la metodología de la programación orientada a objetos.
- Ser independiente de la plataforma.
- Incluir por defecto soporte para trabajo en red.
- Ejecutar código en sistemas remotos de forma segura.
- Ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

La primera característica, orientado a objetos ("OO"), se refiere al método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el "comportamiento" (el código) y el "estado" (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico "cliente", por ejemplo, debería (en teoría) tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando éstos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software la construcción de proyectos de envergadura, empleando componentes ya existentes y de comprobada calidad, conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su "comportamiento" (soldar dos piezas, etc.), el objeto "aluminio" puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

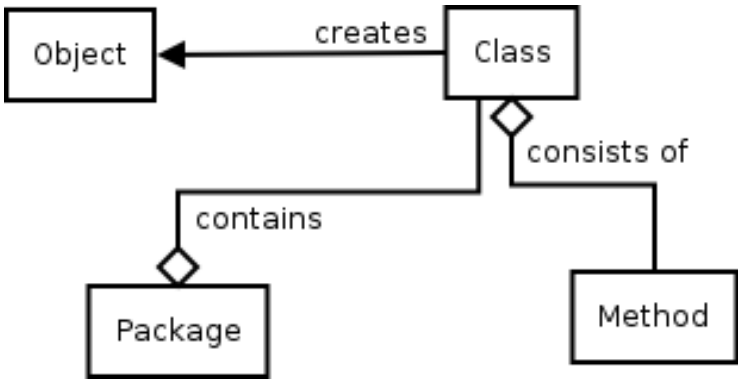

Figura 63 – Modelo Java

En Java el problema de las fugas de memoria se evita, en gran medida, gracias a la recolección de basura (o automatic garbage collector). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los mismos. El programa, u otros objetos, pueden tener localizado un objeto mediante una referencia a éste. Cuando no quedan referencias a un objeto, el recolector de basura de Java lo borra, liberando así la memoria que ocupaba, previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aun así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios. Es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y puede que más velocidad que C++.

El motivo del uso de Java en este proyecto, es el hecho de que Android requiere Java para programar la funcionalidad de las aplicaciones.

3.2.4.2. PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas [11]. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas, incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools o Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994, sin embargo, la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin coste alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. El número de sitios en PHP ha compartido algo de su preponderante posición con otros nuevos lenguajes no tan poderosos desde agosto de 2005. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web. La versión más reciente de PHP es la 5.3.5, del 6 de enero de 2011.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender un nuevo grupo de funciones. Aunque todo en su diseño está orientado a facilitar la creación de sitios webs, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Este procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo, obteniendo información de una base de datos). El resultado es enviado al cliente. Mediante extensiones también es posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos. Además, permite la conexión a diferentes tipos de servidores de bases de datos tales

como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Figura 64 – Proceso de peticiones PHP

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (como Linux o Mac OS X) y Windows, y puede interactuar con los servidores web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. Es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# VB.NET como lenguajes), a ColdFusion de la compañía Adobe (antes Macromedia), a JSP/Java de Oracle, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE (entorno de desarrollo integrado) comercial llamado Zend Studio. Recientemente, CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un entorno integrado de desarrollo para PHP, denominado Delphi for PHP. También existen, al menos, dos módulos para Eclipse, uno de los IDE más populares.

La necesidad de utilizar PHP en el proyecto viene motivada por la obligación de usar un puente entre la aplicación Android y la base de datos remota. Dada la facilidad, optimización y velocidad de PHP junto con su potente compatibilidad con MySQL se ha decidido optar por esta combinación.

3.2.4.3. XML

XML, siglas en inglés de eXtensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es, a su vez, un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. De hecho, en Android, XML se utiliza para definir las interfaces de usuario y sus componentes, motivo por el cual es necesaria su utilización en este proyecto. A continuación se puede ver en la siguiente Figura una pequeña interfaz Android que muestra dos botones, enter y exit.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:gravity="center_vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/enter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/enter"
        android:width="100px"
        android:layout_marginLeft="40dip"/>
    <Button
        android:id="@+id/exit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/exit"
        android:width="100px"
        android:layout_alignBaseline="@+id/enter"
        android:layout_alignParentRight="true"
        android:layout_marginRight="40dip"/>
</RelativeLayout>
```

Figura 65 – Ejemplo Layout Android

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades más extensas. Tiene un papel muy importante en la actualidad, ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Estas son algunas de las ventajas de XML:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- Transformación de datos en información, pues se les añade un significado concreto y se asocian a un contexto, con lo cual se tiene flexibilidad para estructurar documentos.

XML proviene de un lenguaje inventado por IBM en los años setenta, llamado GML (Generalized Markup Language), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (Standard Generalized Markup Language), capaz de adaptarse a un gran abanico de problemas. A partir de él surgieron variaciones, pasando por HTML, que no cumplían con todas las expectativas.

Se buscó entonces definir un subconjunto del SGML que permitiera:

- Mezclar elementos de diferentes lenguajes. Es decir, que los lenguajes sean extensibles.
- La creación de analizadores simples, sin ninguna lógica especial para cada lenguaje.
- Empezar de cero y hacer hincapié en que no se acepte nunca un documento con errores de sintaxis.

Para cumplir estos objetivos nace XML, dejando de lado muchas características de SGML que estaban pensadas para facilitar la escritura manual de documentos. XML está orientado a hacer las cosas más sencillas para los programas automáticos que necesiten interpretar documentos.

3.2.4.4. SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales, que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional, permitiendo efectuar consultas con el fin de recuperar (de una forma sencilla) información de interés de una base de datos, así como también hacer cambios sobre ella.

Los orígenes del SQL están ligados a las de las bases de datos relacionales. En 1970 E. F. Codd propone el modelo relacional y asociado a éste un sub-lenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje SEQUEL (Structured English QUEry Language) que más tarde sería ampliamente implementado por el sistema de gestión de bases de datos (SGBD) experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial.

SEQUEL terminaría siendo el predecesor de SQL, siendo éste una versión evolucionada del primero. SQL pasa a ser el lenguaje por excelencia de los diversos sistemas de gestión de bases de datos relacionales surgidos en los años siguientes y es, por fin, estandarizado en 1986 por el ANSI, dando lugar a la primera versión estándar de este lenguaje, el "SQL-86" o "SQL1". Al año siguiente, este estándar es también adoptado por la ISO.

En la actualidad SQL es el estándar de facto de la inmensa mayoría de los SGBD comerciales. Y, aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL es general y muy amplio.

SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales, permitiendo gran variedad de operaciones en éstos últimos. Es un lenguaje declarativo de "alto nivel" o "de no procedimiento", que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros.

Como ya se dijo anteriormente, y suele ser común en los lenguajes de acceso a bases de datos de alto nivel, SQL es un lenguaje declarativo. O sea, que especifica qué es lo que se quiere y no cómo conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución. El orden de ejecución interno de una sentencia puede afectar gravemente a la eficiencia del SGBD, por lo que se hace necesario que éste lleve a cabo una optimización antes de su ejecución. Muchas veces, el uso de índices acelera una instrucción de consulta, pero ralentiza la actualización de los datos. Dependiendo del uso de la aplicación, se priorizará el acceso indexado o una rápida actualización de la información. La optimización difiere sensiblemente en cada motor de base de datos y depende de muchos factores.

El uso de SQL en el proyecto viene motivado por la elección de MySQL como sistema gestor de base datos en combinación con PHP, para ejercer de puente entre la base de datos y la aplicación Android. Esta combinación nos asegura eficiencia, velocidad y seguridad.

3.2.5. PLATAFORMA ANDROID

Android es un sistema operativo basado en Linux para dispositivos móviles, como teléfonos inteligentes y tablets. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en el 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre.

Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. En la actualidad existen cerca de 200.000 aplicaciones disponibles para Android. Android Market es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Tal y como ya se ha explicado en la memoria, los programas están escritos en el lenguaje de programación Java.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. Actualmente

Android posee el 32,9% de cuota de mercado a escala mundial de los teléfonos inteligentes, por delante de Symbian que tiene el 30,6%.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas 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 Glibc. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Los componentes principales de la arquitectura del sistema operativo de Android son los siguientes:

- **Aplicaciones**: las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones**: los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes. Cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas**: Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android**: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma

eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida "dx".

- **Núcleo Linux**: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

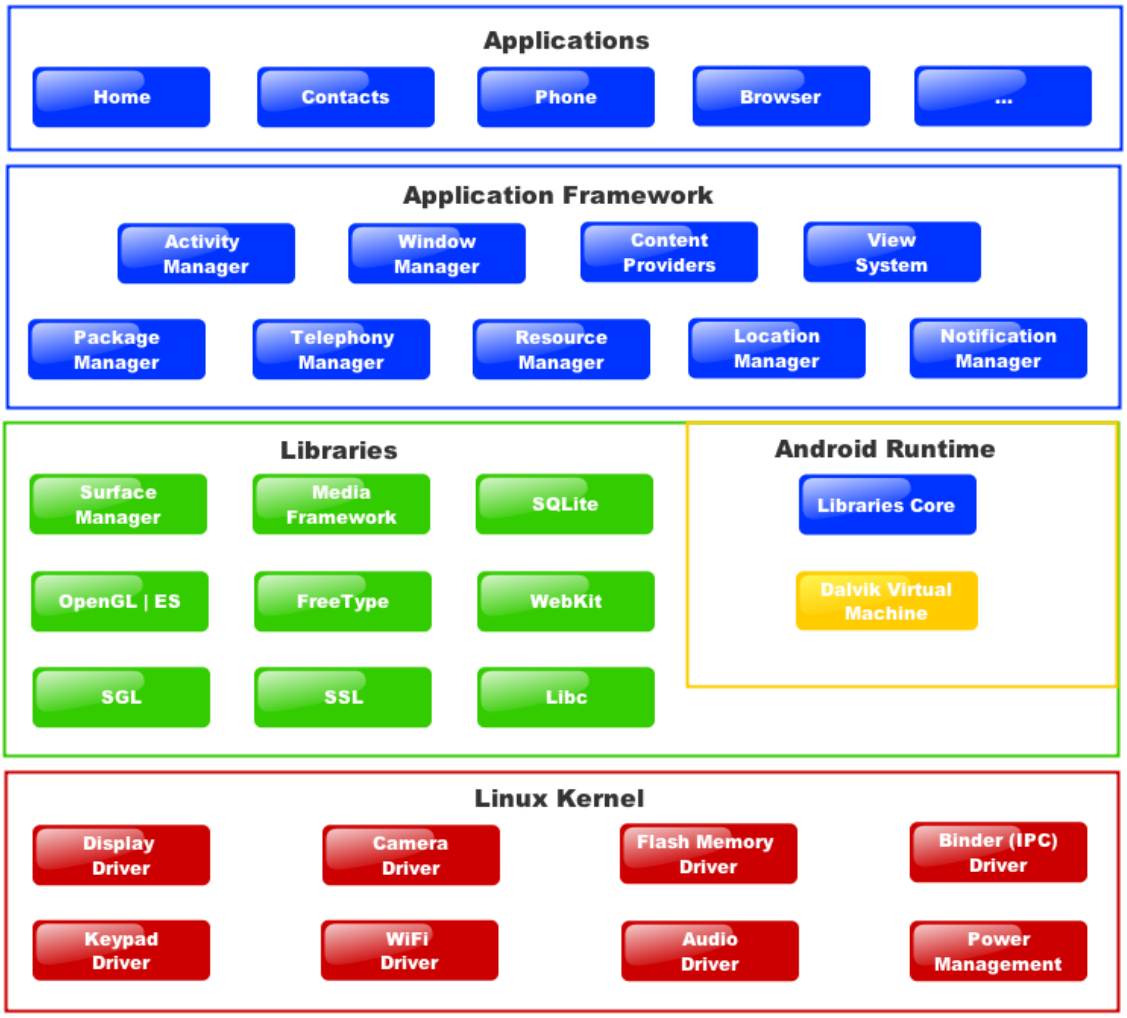

Figura 66 – Arquitectura Android

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.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en determinado móvil, porque el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un "trozo" básico del firmware para poder hacerlo funcionar en dicho terminal, además de que las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos pueden tener dificultades por razones de memoria (RAM), velocidad de procesador, etc.

Las aplicaciones en Android funcionan bajo el esquema de "Activities". Una Activity presenta una interfaz gráfica (escrita en XML) que permite al usuario interactuar con la aplicación. Cada aplicación tiene varias Activities que se van mostrando al usuario según éste las vaya necesitando. Una Activity llamará a otra cuando sea necesario, y cada una de las Activities que se vayan mostrando se almacenan en una pila; es decir, cada vez que la aplicación lo requiera, inserta una nueva Activity en la pila. Cuando las aplicaciones quedan en segundo plano, Android, de forma aleatoria y según su necesidad de obtener memoria, va cerrando Activities de aplicaciones que no estén en ese momento en uso por parte del usuario, es decir, que estén en segundo plano.

El motivo de la elección de Android como plataforma para la cual desarrollar la aplicación GPSLoc es el hecho de la rápida evolución y expansión que está teniendo este sistema operativo, tal y como se ha comentado anteriormente, el cual está desbancando, uno tras otro, a sus competidores en características y cuota de mercado. Además, otro de los factores que juega más a favor de este sistema operativo es el hecho de que no se limita a un solo dispositivo móvil de una sola empresa, como iOS, sino todo lo contrario, cada vez son más las compañías que sacan al mercado nuevos modelos que incorporan el sistema operativo Android.

4. ESCENARIO DE USO DE LA APLICACIÓN

En el siguiente apartado de la memoria será descrito un escenario de uso de GPSLoc para demostrar el correcto funcionamiento de la misma y sus aplicaciones en ese escenario.

4.1. INTRODUCCION Y OBJETIVOS

El escenario de uso de la aplicación elegido para demostrar las capacidades de la misma está ubicado dentro del ámbito de la localización entre un grupo de amigos. El objetivo de la aplicación, dentro de este escenario de uso, será ofrecer a los usuarios la posibilidad de mantener localizados a sus amigos en todo momento, evitando la necesidad de tener que contactar con ellos para acudir al mismo lugar en el que se encuentren. Los usuarios podrán visualizar la posición de sus amigos en todo momento o en un determinado rango de tiempo, según el tipo de permiso elegido (ya explicados anteriormente). El escenario estará constituido por un grupo de tres actores, quienes manejarán la aplicación para mantenerse localizados entre sí.

4.2. ACTORES

Los tres actores que participan en el escenario son los siguientes:

1. Pablo Sáez
2. Fanny Tadeo
3. Víctor Cervera

4.3. DESCRIPCION

Para describir el escenario de uso de la aplicación se partirá desde la situación en la que los siguientes permisos estarán asignados:

- Pablo: Permiso de visibilidad total sobre Fanny y permiso de visibilidad por horas sobre Víctor, concretamente entre las 10 de la mañana y las 22 de la noche.
- Fanny: Permiso de visibilidad total sobre Víctor y sobre Pablo.
- Víctor: Permiso de visibilidad por días sobre Pablo, concretamente en la franja de tiempo comprendida entre los días 1 de enero de 2011 y 15 de junio de 2011.

Los actores habrán almacenado el siguiente número de posiciones en la base de datos:

- Pablo: Tres posiciones diferentes, dos de ellas el día 15 de junio y la última el día 16 de junio.
- Fanny: Una sola posición, por lo que solo esa posición estará almacenada en la base de datos.
- Víctor: Tres posiciones diferentes a lo largo del tiempo que fue utilizada la aplicación, dos de ellas dentro del rango horario de 10 de la mañana a 22 de la noche, y una de ellas a las 23:01 de la noche.

A continuación se presenta en la siguiente figura el mapa de amigos que visualizará el actor Pablo.

Figura 67 – Mapa de amigos de Pablo, escenario de prueba

Tal y como se puede observar en el mapa, Pablo puede ver la última posición válida de Fanny y Víctor. Para profundizar más en el escenario, en las siguientes figuras se muestran las últimas posiciones de los dos amigos de Pablo, las cuales se obtienen accediendo a la vista detallada de cada amigo y pulsando sobre el botón "Last Positions". Las posiciones de los amigos aparecen representadas por puntos morados en el mapa.

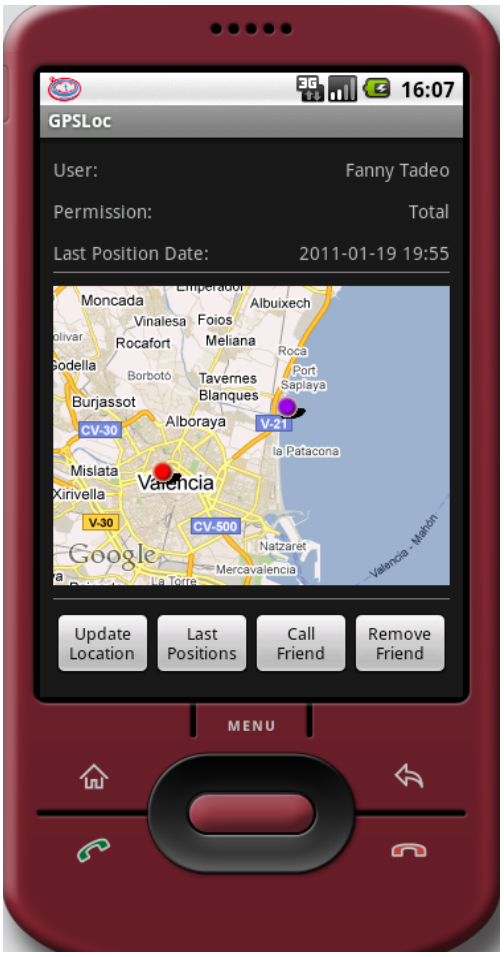

Figura 68 – Últimas posiciones de Fanny

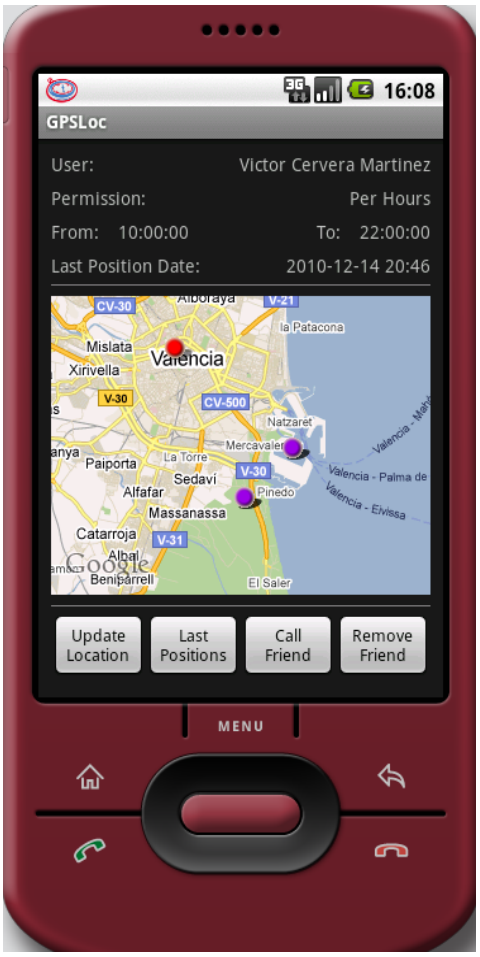

Figura 69 – Últimas posiciones de Víctor

Tal y como puede observarse, Pablo puede visualizar las últimas posiciones en las que han estado sus amigos, pero solo dos de las posiciones de Víctor, a pesar de que Víctor ha estado en tres lugares diferentes. La tercera posición de Víctor fue almacenada a las 23:01, por lo tanto no cumple el permiso que Pablo posee sobre él, y no aparece en el listado de últimas posiciones de Víctor a las que Pablo puede acceder.

A continuación, en la siguiente figura, serán mostradas las posiciones de Víctor que son visibles para Fanny (Fanny posee permiso Total sobre Víctor), es decir, todas, incluyendo la que Pablo no podía visualizar.

Figura 70 – Últimas posiciones de Victor visibles para Fanny

Para profundizar aún más en el escenario, en la siguiente figura serán mostradas las últimas posiciones de Pablo, vistas desde el teléfono móvil de Víctor, el cual posee permiso por días sobre Pablo. Es importante recordar que la última posición de Pablo fue almacenada el día 16, y el permiso de visibilidad por días que Víctor ha obtenido sobre Pablo termina el día 15, por lo tanto, esa tercera posición no aparece en el mapa.

Figura 71 – Ultimas posiciones de Pablo visibles para Víctor

A continuación, en la siguiente figura, serán mostradas las posiciones de Pablo que son visibles para Fanny (Fanny posee permiso Total sobre Pablo), es decir, todas, incluyendo la que Víctor no podía visualizar.

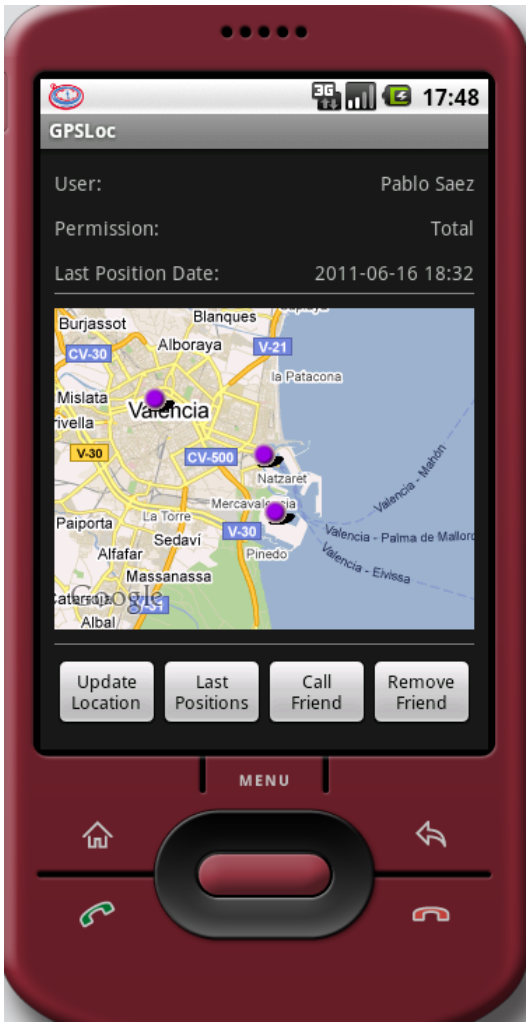

Figura 72 – Ultimas posiciones de Pablo visibles para Fanny

Durante este escenario se ha demostrado un uso práctico de la aplicación en un determinado escenario sin profundizar en todas sus características, debido a que sería redundante con el punto 2.3 de la memoria y el manual anexo a la misma. El escenario presentado podría aplicarse tanto a situaciones empresariales como familiares. Por ejemplo, el permiso por horas que el actor Pablo posee sobre Víctor podría aplicarse a una empresa de transportes, en la que el gerente desea saber dónde se encuentran sus empleados durante el horario laboral, mientras que el permiso por días que Víctor posee sobre Pablo podría aplicarse al caso de un matrimonio que realiza un viaje de pareja durante una semana y desean mantener localizados a sus hijos durante esos días.

5. PROBLEMAS Y SOLUCIONES ADOPTADAS

A lo largo del proyecto, han aparecido múltiples barreras que superar, para las cuales ha sido necesario adoptar determinadas medidas para aportar soluciones. En los siguientes puntos se detallan los problemas y las soluciones adoptadas para cada uno de ellos.

LIMITACIONES PLATAFORMA MOVIL

El hecho de que los dispositivos sobre los cuales va a ser ejecutada la aplicación son dispositivos móviles implica dos limitaciones muy importantes.

Debido al reducido tamaño de la pantalla, es necesario otro tipo de organización en las interfaces totalmente diferente a las aplicaciones de escritorio. Para ello, se ha decidido seguir los estándares más importantes y conocidos para interfaces Android, usando tonos oscuros y grisáceos para la aplicación, agrupando la navegabilidad en el menú desplegable y utilizando la interfaz más útil para el usuario como pantalla principal del programa. Estas características serán explicadas a continuación.

Los tonos grisáceos con fondo oscuro usados en la aplicación son los que más agilizan la interacción del usuario con las interfaces en los dispositivos móviles, debido a que el ojo humano tiene mayor facilidad para encontrar la información o la parte de la interfaz deseada. Además, el uso de tonos oscuros y grisáceos aumenta la duración de la batería.

La navegabilidad entre secciones de la aplicación ha sido agrupada en el menú desplegable por la tecla *Menú* del teléfono móvil, de forma que ésta no ocupe espacio en la pantalla cuando no esté desplegado. Cuando el usuario despliegue el menú, podrá elegir entre la sección *Amigos*, la sección *Invitaciones*, la sección *Configuración*, y hacer log-out.

La pantalla principal del programa es el mapa de amigos, debido a que es lo primero que el usuario siempre va a querer ver y es la pantalla que más información útil muestra. Encima del mapa, han sido agrupadas en diferentes pestañas las diversas posibilidades para la visualización de los amigos (mapa o listas, ya explicadas anteriormente).

Otra de las limitaciones más importantes en el ámbito de los dispositivos móviles, es su velocidad de acceso a Internet, la cual es muy limitada y dependiente del lugar en el que se encuentre el dispositivo, por lo que es necesario mantener el tráfico en red lo más reducido posible. Actualmente se ha extendido la velocidad 3G, considerada suficiente y recomendable para esta aplicación, tal y como se ha comentado anteriormente. Sin embargo, para disminuir el tráfico y las peticiones al mínimo posible ha sido implementada una base de datos local, descrita próximamente en la memoria. Además, se ha procurado disminuir el tráfico haciendo el menor número de llamadas posibles al servidor remoto y ajustando al mínimo necesario la cantidad de datos transmitidos, tomando como ejemplo directo la limitación a 100KB para las fotos de perfil de los usuarios.

PROBLEMA RECUPERACIÓN DE INFORMACION

Al tratar de rellenar el mapa de amigos, o cualquier otra funcionalidad para la que haya que solicitar datos a la BD para rellenar listas, inicialmente se optó por la estrategia de colocar el código para obtener y rellenar en el onCreate(), para que se realizara la tarea al cargar la ventana por primera vez, y además en el onResume(), para que se realizara la tarea al volver a la ventana desde otra interfaz. Esto generó un grave problema, ya que en Android, solo la primera vez que se carga una Activity será llamado el método onCreate(), y el resto de veces que esa Activity sea mostrada, se llamará el método onResume().

Para solventar el problema, se ha optado por poner todo el código de obtención de datos y rellenado de listas en el método onResume() de todas las Activities. El método onResume() es llamado siempre que una ventana se muestra al usuario, incluida la primera vez, por lo tanto, con colocar el código en ese método es suficiente, ya que si fuera colocado también en el onCreate(), estaría repitiéndose el trabajo de carga la primera vez que se iniciara una Activity, puesto que haría el mismo trabajo en ambos métodos.

CONFLICTO DE LAS INVITACIONES

Inicialmente se pensó que para enviar una invitación a una persona sería siempre necesario indicar los parámetros del permiso que se quiere obtener sobre él. No se tuvo en cuenta la idea de que no todo el mundo al que se desee invitar pudiera tener la aplicación instalada. Esto resultaba incongruente para las invitaciones a personas externas al programa, por tanto se opto por dividir las invitaciones en dos tipos: invitación a usuarios de GPSLoc e invitación a usuarios externos mediante Email.

Esta nueva visión de las invitaciones trajo consigo nuevos problemas, como la duda de cómo invitar a personas externas y cómo invitar a otros usuarios del programa. A continuación van a ser detalladas las soluciones adoptadas:

Para el caso de las invitaciones a personas externas al programa, primero fue prevista la inclusión de la funcionalidad de enviar mensajes de texto a los usuarios externos a los que quisiéramos invitar. Esta opción quedó descartada debido a que eso generaba un coste para el usuario y además limitaba la amplitud de versiones del sistema operativo para las cuales la aplicación sería compatible, ya que el método de enviar mensajes de texto es bastante complejo y dependiente de la versión del sistema operativo Android para la cual se implemente. En su lugar se opto por implementar el envío de invitaciones a personas externas mediante la inclusión de la capacidad de enviar correos electrónicos de invitación utilizando el gestor de correo interno de Android, para así maximizar la compatibilidad total con todas las versiones del sistema operativo Android (uno de los principales objetivos de la tesina) y, además, eliminar el coste añadido que tenía el sistema de SMS.

Para el caso de las invitaciones a personas usuarias del programa, se ha decidido presentar un sistema parecido al de las redes sociales actuales: en un campo de texto será introducida una cadena de texto correspondiente al nombre de una persona y, mediante un botón Buscar, la aplicación devolverá una lista de coincidencias con los nombres de los usuarios registrados en la aplicación. El usuario podrá seleccionar la persona que desee y, entonces, realizar la invitación indicando los parámetros correspondientes e insertando el permiso en la base de datos con el campo *validated* a cero (*false*). Una vez el otro usuario acepte la invitación, el campo *validated* pasara a ser uno (*true*) y el permiso estará activo.

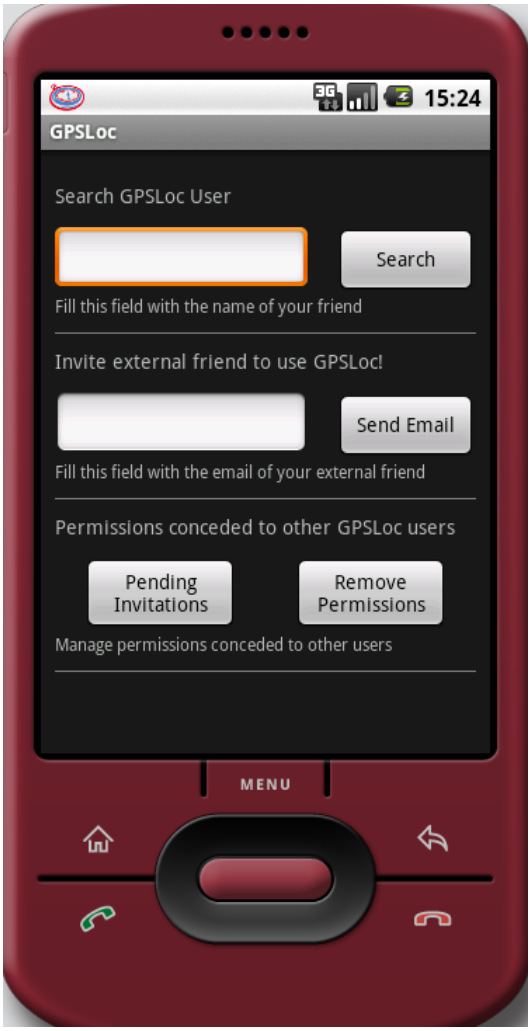

Figura 73 - Invitaciones

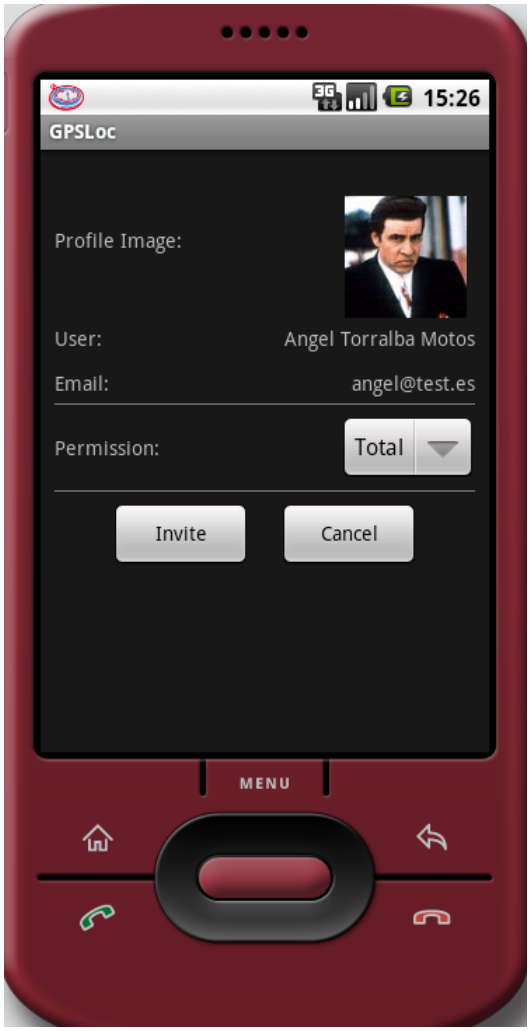

Figura 74 – Invitación permiso total

BASE DE DATOS LOCAL COMPARTIDA ENTRE TODAS LAS ACTIVITIES

Debido a la naturaleza de la aplicación, eran necesarios constantes accesos a la BD Remota para obtener datos, por lo que la velocidad de funcionamiento de la aplicación quedaba seriamente lastrada por el tiempo de respuesta del servidor remoto.

Para optimizar la velocidad de la aplicación se ha optado por incorporar una pequeña BD Local SQLite, la cual almacena los amigos sobre los que tiene permiso el usuario y el propio permiso que tiene sobre el amigo. No se ha optado por almacenar en local las posiciones, ya que la naturaleza de esta aplicación conlleva poder visualizar la posición de nuestros amigos en tiempo real, y para ello lo mejor es descargar la última posición cada vez que se muestre al amigo o amigos en un mapa y no cargar posiciones antiguas almacenadas en local.

Para poder disponer de una base de datos SQLite compartida en toda la aplicación, se ha optado por utilizar una clase Application (MyApplication), la cual inicializa MyDbAdapter, a través del cual se puede acceder a la BD local desde cualquier activity de la aplicación. Las clases Application tienen la ventaja de ofrecer acceso global a las variables que estén almacenando, para todas las activities de la App.

Inicialmente se pensó que, para que el usuario, al iniciar la aplicación, no tuviera que esperar demasiado en tiempo de carga, la mejor opción era que en segundo plano, o en un hilo, se descargaran los amigos y los permisos del usuario, para después obtener la posición. Esta estrategia fue descartada debido a que, para poder descargar la posición de los usuarios, es requisito indispensable haber obtenido ya la lista de amigos y sus permisos, por lo tanto, aunque éstos se obtuvieran en segundo plano, no se podría comenzar a descargar las posiciones mientras la tarea de descargar amigos y permisos no hubiera terminado, con lo que colocar esa tarea en segundo plano sería redundante e inservible.

Finalmente, la estrategia adoptada ha sido la siguiente, la cual se realiza a partir de que el usuario inicie la aplicación. Se siguen estos pasos:

1. Se vacía la BD local (para evitar redundancias o datos obsoletos)
2. Se descargan las listas actualizadas de amigos y permisos en la BD local.
3. Para cada amigo, se descarga la lista de posiciones.
4. Para cada amigo, se obtiene, según el permiso que se posee sobre él, la última posición que cumple los requisitos del permiso.
5. Se muestra el mapa de amigos con la última posición que cumpla los requisitos del permiso de cada amigo.

A partir de este punto, cada vez que se necesite leer los permisos o los amigos, se hará en local, aumentando considerablemente el rendimiento.

Para mantener actualizada la BD local, cada vez que borramos a un amigo, se borra paralelamente de la BD remota y de la BD local, así como cada vez que enviamos una invitación a un usuario que ya es amigo nuestro (por ejemplo, para cambiar de tipo de permiso Por Horas a Total), debido a que el permiso sobre ese usuario pasa a ser "no validado". También se procede a actualizar la lista de amigos y permisos local.

Actualizar la BD cuando enviamos una invitación a otro usuario quedó descartado, ya que ésta puede ser aceptada pasadas varias horas o días, o incluso nunca, con lo que, en ocasiones, la aplicación estaría consumiendo recursos y tiempo del usuario para

actualizar la BD sin realizar ningún cambio en la misma. En su lugar, para mantener la BD local actualizada, se ha seguido la siguiente estrategia: en un servicio en background "MyServiceLocalDB.java", cada 5 minutos, en un hilo se descargará la nueva información actualizada de la BD Remota, para, a continuación, mediante un Handler, insertarla en la BD local. De esta forma, se consigue que el trabajo se realice de forma transparente para el usuario, sin tiempos de espera, ya que la carga pesada que es la obtención de los datos remotos se realiza en el hilo, y la carga suave, que es actualizar la BD Local con la nueva información, se realiza en un Handler, para evitar que otras secciones de la APP puedan tratar de leer de la BD Local mientras ésta se esté actualizando (el Handler aísla las variables de forma que todo lo que quiera acceder a ellas tendrá que esperar a que éste termine).

Otra estrategia que quedó descartada fue la de tener un service en segundo plano que fuera comprobando, cada cierto tiempo, si se habían producido variaciones entre la BD Local y la BD Remota. Esta estrategia consumía prácticamente los mismos recursos que la estrategia anterior, solo que de una forma más compleja, por lo que se optó por descartar esta opción.

PROBLEMA DE CONEXIÓN CON EL SERVIDOR REMOTO

El sistema operativo Android no incluye librerías para realizar conexiones con base de datos remotas, algo que representa un tremendo problema para el proyecto, debido a que éste requiere de un servidor que aloje en una BD Remota a todos los usuarios, permisos y posiciones del programa.

Una de las posibles soluciones era la utilización de un servicio web para realizar la conexión entre la aplicación y el servidor remoto. Esta solución quedó descartada para optar por otra más sencilla que aseguraba velocidad y eficiencia con muy baja complejidad. La solución adoptada consiste en utilizar un puente entre Android y el servidor remoto que combina las tecnologías PHP y JSON para la comunicación. Partiendo de una base de datos MySQL en el servidor remoto, para enviar datos de una tabla a la aplicación, tendremos dos secciones: la parte ejecutada en el servidor remoto y la parte Android. La parte del servidor remoto consiste en un fichero PHP que se conecta a la base de datos, realiza una consulta y devuelve los resultados codificados en formato JSON (la codificación a este formato permite obtener y enviar los datos de forma fluida y rápida). La parte Android utiliza un método HttpPost para

obtener los datos, transforma la respuesta a String, y finalmente realiza un parse a los datos JSON para transformarlos en String, int, o el tipo de datos necesario.

Otra posibilidad era utilizar XML en lugar de JSON para la transmisión de los datos, pero quedó descartada dada la sencillez, facilidad, seguridad, optimización y velocidad que ofrece la combinación de PHP y JSON.

SEGURIDAD EN LOS PHP

Una persona externa a la aplicación, con fines maliciosos, averiguando la ruta de los ficheros PHP, podría tratar de acceder a la información de los usuarios, algo que hay que evitar a toda costa. El acceso a todos los datos remotos se realiza mediante los ficheros PHP, por lo tanto es indispensable aumentar la seguridad en los mismos.

Para proteger los ficheros PHP se han tomado dos medidas de seguridad: protegerlos de SQL Injection (un método de infiltración de código intruso que se vale de una vulnerabilidad informática de SQL para realizar consultas a una base de datos) y protegerlos de ser accesibles a través de un navegador. Para evitar la intrusión de código malicioso en la Base de Datos mediante SQL Injection, los PHP han sido protegidos mediante el método PHP "*mysql_real_escape_string*", el cual coloca barras invertidas antes de ciertos caracteres (\x00, \n, \r, \, ', " y \x1a), evitando así el código malicioso insertado.

Para asegurar que no se pueda acceder a los ficheros PHP a través de un navegador y de forma externa a la aplicación, al inicio de cada PHP se ha añadido una comprobación mediante la cual se compara un parámetro con una palabra secreta que solo conoce el código interno de la aplicación (if ($_REQUEST['app_password'] =='secret')). Además de esta comprobación realizada en cada fichero PHP, dentro del código java, en la clase que se comunica con los PHP, RemoteConnection.java, para cada función de conexión, se ha añadido un parámetro más, el parámetro "*app_password"*, con valor "*secret".* De esta forma, como condición de los PHP para realizar las Querys a la Base de Datos, comprobaremos que ha sido recibida como parámetro la palabra "secret", solo conocida por el código interno de la aplicación.

CONSUMO EXCESIVO DE BATERÍA

Inicialmente, la aplicación estaba diseñada para leer la posición GPS del usuario cada segundo y enviarla al servidor en ese intervalo. Se pretendía ofrecer en tiempo real la posición de la forma más exacta posible. Tras las primeras pruebas de la aplicación se comprobó que con esa configuración la batería del móvil se agotaba en una media de 6 horas.

La solución adoptada fue leer la posición GPS cada 10 segundos y enviarla al servidor remoto cada minuto, de forma que la duración de la batería aumentaba considerablemente, disparándose hasta las 12 horas de media.

MANTENER LA APLICACIÓN EJECUTÁNDOSE EN BACKGROUND

GPSLoc está diseñada para ser una aplicación que el usuario pueda dejar en segundo plano (background) enviando su posición al servidor remoto, hasta que el propio usuario decida cerrar la aplicación. En Android, cuando una aplicación pasa a segundo plano, sus Activities quedan paralizadas para ahorrar energía, de forma que si tuviera que transmitir o recibir datos en background, no se realizarían estas acciones. Además, Android, cada cierto tiempo, elimina las Activities paralizadas de la memoria.

La solución a este problema ha sido implementar la funcionalidad que debe permanecer en segundo plano dentro de Services. Los Services de Android son clases que continúan su ejecución en segundo plano aunque la aplicación no esté siendo visualizada por el usuario en un determinado momento. Esto soluciona el problema de las funcionalidades en segundo plano, pero no el problema de que Android cierre cada cierto tiempo Activities que estén paralizadas. Para solucionar este último problema simplemente ha sido necesario cambiar el campo “android:alwaysRetainTaskState” a *true* dentro del Manifest, con lo que se consigue que Android no cierre ninguna Activity paralizada de la aplicación.

PROBLEMA DEL NOMBRE DE LOS AMIGOS EN LOS MAPAS

Para mejorar la usabilidad de la aplicación, uno de los objetivos pensados inicialmente consistía en que, cuando el usuario estuviera visualizando el mapa de amigos, fuera mostrado el nombre de los amigos debajo del icono que representa su posición, de forma que el usuario pudiera identificar qué amigo era de manera instantánea, sin necesidad de ir a su descripción detallada. Esta posibilidad quedó descartada debido a que Android no permite implementar dicha característica en la versión actual de la API. En su lugar, ha sido implementada una pre-visualización del amigo mediante un pequeño cuadro en el que se muestra su foto, su email, su nombre y un botón para ir a los detalles del amigo. Dicho cuadro se visualiza cuando el usuario presiona sobre el icono que representa la posición del amigo.

Figura 75 – Mapa de amigos

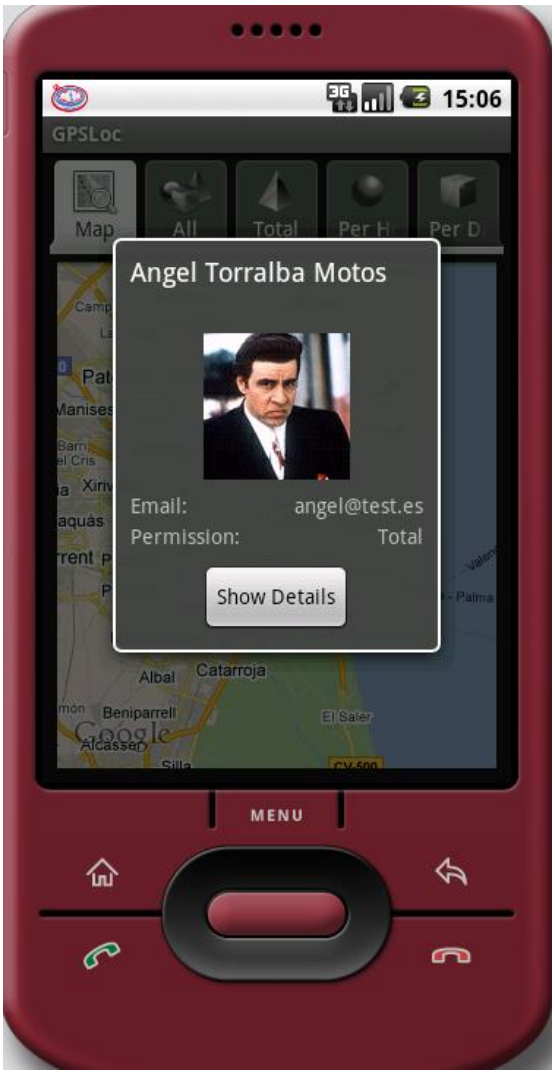

Figura 76 – Cuadro con detalles del amigo

INTERFERENCIAS ENTRE DOS MAPAS EN UNA MISMA APLICACIÓN

GPSLoc tiene dos tipos de mapas: el mapa en el que se muestran todos los amigos y el mapa en el que se muestra la posición (o últimas posiciones) de un solo amigo. Android está diseñado de forma que una misma aplicación soporta de forma óptima una sola Activity mostrando un mapa, pero si la misma aplicación tiene diferentes Activities con mapas, estos interfieren entre sí mostrando datos erróneos.

La solución a este problema ha sido ejecutar las Activities que muestran el mapa con la posición (o últimas posiciones) de un amigo en concreto, como un proceso extra. Al ejecutar cada mapa en un proceso diferente, éstos no interfieren entre sí. Para realizar la separación en procesos tan solo hay que modificar la línea del Manifest en la que se declara la Activity añadiendo *android:process=":MapViewExtra"*.

PROBLEMA DE LOS CUADROS DE PROGRESO

Durante la realización de la tesina, fue barajada la posibilidad de mostrar un ProgressDialog (cuadro de progreso) durante los tiempos de espera de la aplicación, pero esta posibilidad quedó descartada.

Uno de los motivos principales de su descarte, es el hecho de que el servidor remoto utilizado durante el desarrollo de la aplicación ofrece un tiempo de respuesta considerablemente peor que el que ofrecería un servidor potente, por lo tanto, usando un servidor remoto con más recursos, los tiempos de espera que tiene actualmente la aplicación serían prácticamente inapreciables, ya que éstos son consecuencia del retraso que tiene el servidor remoto en contestar a las peticiones de la aplicación, y no tiene nada que ver la cantidad de datos que se devuelven en las peticiones ni la potencia de proceso del dispositivo móvil. La cantidad de datos devueltos en cada petición no es influyente en los tiempos de espera de la aplicación, debido a que, por la naturaleza de la aplicación, siempre será una cantidad muy reducida.

Sin embargo, el motivo principal de haber descartado esta característica, es que sería contraproducente, debido a que en GPSLoc, estos cuadros de progreso son necesarios solo al pasar de una interfaz a otra, y en Android, introducir un cuadro de progreso en esa situación acarrea diversos inconvenientes, todo lo contrario a cuando lo introducimos en otro tipo de situaciones en las que no son necesarias transiciones de interfaces, en cuyo caso es más sencillo y no acarrea inconvenientes de ningún tipo. Para poder introducir esa característica en los cambios de interfaz a interfaz de

GPSLoc hubiera sido necesario implementar para cada caso una AsyncTask, a la cual habría de trasladar tanto la lógica de la activity que está dejando el usuario, como parte de la lógica de la activity a la que va a entrar y realizar comunicaciones entre las tres partes que ralentizarían y aumentarían la carga de la aplicación de forma totalmente contraproducente.

6. TRABAJO Y MEJORAS FUTURAS

Este apartado de la memoria está dedicado a exponer en qué puntos puede mejorar GPSLoc en un futuro mediante hipotéticas ampliaciones del proyecto.

6.1. SERVIDOR REMOTO MÁS POTENTE

Tal y como se ha comentado durante la memoria, uno de los objetivos primordiales en las aplicaciones de esta naturaleza es la fluidez de la aplicación de cara al manejo por parte del usuario.

Este es uno de los aspectos a mejorar del proyecto, ya que, tal y como se ha comentado anteriormente, el mayor hándicap encontrado durante el desarrollo de esta tesina, ha sido el tiempo de respuesta del servidor remoto, el cual oscila entre 200 milisegundos y 2 segundos, de forma variable y aleatoria.

Migrando a un servidor remoto con más recursos, los tiempos de espera que tiene actualmente la aplicación serían inapreciables, ya que estos son consecuencia del retraso que tiene el servidor remoto en contestar a las peticiones de la aplicación, y no tiene prácticamente nada que ver la cantidad de datos que devuelve en las peticiones, ni la potencia de proceso del dispositivo móvil. La cantidad de datos devueltos en cada petición no es influyente en los tiempos de espera de la aplicación, debido a que por la naturaleza de ésta, siempre será una cantidad muy reducida.

6.2. GPSLOC PARA IPHONE

El dispositivo móvil de Apple es uno de los que más cuota de mercado posee junto con los dispositivos Android, por lo que sin duda sería una gran opción migrar la aplicación GPSLoc a Iphone para así abarcar un mayor número de clientes. Además, Apple tiene una ventaja importante sobre sus rivales, y es el hecho de que todos sus dispositivos móviles tienen el mismo hardware, exceptuando pequeñas mejoras que van apareciendo periódicamente en nuevas versiones del Iphone. Este hardware “casi”

constante permite a los programadores conocer con mejor exactitud cómo optimizar y hasta qué punto exprimir el potencial de la aplicación.

Figura 77 - Iphone

El iPhone es un teléfono inteligente multimedia con conexión a Internet, pantalla táctil capacitiva (con soporte multitáctil) y una interfaz de hardware. Ya que carece de un teclado físico, integra uno en la pantalla táctil con orientaciones tanto vertical como horizontal. El iPhone 3GS dispone de una cámara de fotos de 3 mega píxeles y un reproductor de música (equivalente al del iPod), además de software para enviar y recibir mensajes de texto y mensajes de voz. También ofrece servicios de Internet como leer correo electrónico, cargar páginas web y conectividad por Wi-Fi. La primera generación de teléfonos era GSM con tecnología EDGE; la segunda generación ya incluía UMTS con HSDPA.

Para poder desarrollar aplicaciones para IPhone es necesario conocer el lenguaje Objetive C [12]. Es un lenguaje de programación basado en C, y que ha sido

modificado para poder trabajar orientado a objetos, con lo cual pueden crearse clases, objetos, variables de instancias, métodos, encapsulación, etc.

También es indispensable el conocimiento de Cocoa Touch, el Framework de desarrollo para Iphone. Cocoa Touch es un API para la creación de programas para el iPad, iPhone y iPod Touch de la compañía Apple Inc. Proporciona una capa de abstracción al sistema operativo iOS y se basa en el set de herramientas que proporciona el API de Cocoa para crear programas sobre la plataforma Mac OS X.

6.3. GPSLOC PARA WINDOWS PHONE 7

El sistema operativo Windows Phone 7 es uno de los que más crecimiento está experimentando desde su salida. Recientemente ha sido anunciado su acuerdo con Nokia para sustituir a Symbian como sistema operativo de sus dispositivos móviles, aumentando aun más el crecimiento de este sistema operativo, por lo que sin duda sería una gran opción migrar la aplicación GPSLoc a Windows Phone 7, abarcando así un mayor número de clientes.

Figura 78 – Windows Phone 7

Windows Phone 7 es un sistema operativo móvil desarrollado por Microsoft [13], como sucesor de la plataforma Windows Mobile. Está pensado para el mercado de consumo generalista en lugar del mercado empresarial, por lo que carece de muchas funcionalidades que proporciona la versión anterior. WP7 está disponible en Europa y Asia desde el 21 de octubre de 2010 y en EEUU desde el 8 de noviembre de 2010. Con WP7 Microsoft ofrece una nueva interfaz de usuario, integra varios servicios en el sistema operativo y planea un estricto control del hardware que utilizará el sistema operativo, evitando la fragmentación con la evolución del sistema.

El desarrollo de aplicaciones para Windows Phone 7 puede abarcarse empleando dos tipos de implementaciones:

- Microsoft Silverlight, entorno que permite realizar aplicaciones que contengan transiciones y efectos visuales. Silverlight permite el desarrollo de aplicaciones basadas en XAML y soporta un subconjunto de las librerías de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework. Este soporte incluye el Base Class Library, una colección de clases que soportan lectura y escritura de ficheros, manipulación XML y manejo de gráficos. Cada aplicación que es ejecutada en Windows Phone OS 7.0 CTP se ejecuta dentro de un proceso en el motor de ejecución .NET Compact Framework.
- Microsoft XNA Framework, una implementación nativa de .NET Compact Framework que incluye un amplio conjunto de bibliotecas de clases específicas para el desarrollo de juegos, por ejemplo para el manejo de dispositivos de entrada, tratamiento de sonidos y vídeos, carga de modelos y texturas, uso de ficheros de forma transparente a la plataforma en la que se ejecute, desarrollo de juegos online, etc… Permite desarrollar juegos para Windows Phone OS 7.0 CTP, Xbox 360, Zune HD y Windows 7.

El soporte, ayuda e información para el desarrollo de aplicaciones se centraliza en el Centro de Desarrollo de MSDN en español.

6.4. ANDROID 3.0 Y LAS TABLETS

La llegada al mercado de las tablets ha revolucionado el mundo de la informática. Un tablet es un pequeño dispositivo portátil con el que se puede interactuar a través de una pantalla táctil o multitáctil. El usuario puede utilizar una pluma stylus o los dedos para trabajar con el ordenador, sin necesidad de teclado físico o mouse, al igual que con los móviles de última generación. Esta modalidad de computadora portátil ha supuesto un avance significativo en la aplicación de los estudios en lingüística computacional. Existen modelos de tablet que sólo aportan la pantalla táctil a modo de pizarra, siendo así muy ligeros. También hay ordenadores portátiles con teclado y mouse, llamados convertibles, que permiten rotar la pantalla y colocarla como si de una pizarra se tratase, para su uso como tablet.

Figura 79 - Tablet

Utilizan procesadores sencillos, que consumen menos energía. El software especial que proporciona el sistema operativo permite realizar escritura manual, tomar notas a mano alzada y dibujar sobre la pantalla. Así, es útil para hacer trabajos de campo.

Los dispositivos más abundantes son los de 7' y los de 10'. Los primeros son los de más fácil transporte. Sin embargo, los segundos son más versátiles dado su mayor

tamaño de pantalla, por lo que pueden utilizarse para desarrollar un mayor número de funciones. A modo de ejemplo, ambos tamaños de pantalla permiten la lectura de un libro electrónico, pero los dispositivos de 10', además permiten visualizar comics con cierta comodidad. Así pues, dependiendo del uso que se pretenda dar al dispositivo, se optará por uno de mayor o menor tamaño.

El tablet más conocido y el que revolucionó el mercado fue el IPad, pero no por ello es el único. En la actualidad, hay decenas de modelos que incorporan otros sistemas operativos, como por ejemplo, Android.

La futura versión 3.0 de Android, incorporará una mayor compatibilidad con tablets, por lo que no sería descabellado sopesar la posibilidad de adaptar GPSLoc al mercado de las tablet.

6.5. VERSION WEB DE GPSLOC

El desarrollo de una versión funcional de la aplicación para navegadores web sería una importante ampliación de futuro, de forma que cualquier usuario pudiera consultar la posición de sus amigos a través del navegador en su ordenador personal.

Esta posibilidad eliminaría la dependencia de dispositivos móviles para utilizar la aplicación y ampliaría el espectro de potenciales usuarios de GPSLoc. Al quedar suprimidas las limitaciones de los dispositivos móviles, esta versión web podría ofrecer prácticamente el mismo contenido pero de una forma más detallada, expresiva y usable. Al tratarse de una aplicación que sería ejecutada en ordenadores sin dispositivo GPS, los usuarios no podrían emitir su posición actual. Por lo tanto, la aplicación estaría principalmente orientada a consultar las posiciones de otros usuarios. Se puede obtener la ubicación de un PC mediante la dirección IP, pero no es posible obtener una posición suficientemente ajustada como para ser tomada como referencia en este tipo de aplicaciones.

7. CONCLUSIONES

En esta tesina del Master de Ingeniería del Software, Métodos Formales y Sistemas de Información se ha presentado la aplicación GPSLoc, sus funcionalidades, su arquitectura, las tecnologías utilizadas, los problemas encontrados durante el desarrollo y las posibles mejoras de futuro.

GPSLoc es una aplicación joven pero potente, que ofrece la posibilidad de mantener localizadas a otras personas, según cierto tipo de criterios, ya sean amigos, familiares, empleados de una empresa o tengan cualquier otro tipo de relación con el usuario de la aplicación. Para conseguir esto, la aplicación implementa tres tipos de permisos de visibilidad: permiso total (posición del usuario visible a cualquier hora del día y cualquier día de la semana), permiso por horas (posición del usuario visible solo durante un rango de horas y, si así se indica, fines de semana incluidos) y permiso por días (posición del usuario visible solo durante un rango de días). GPSLoc ofrece una amplia gama de funcionalidades atractivas para el usuario, desde las más comunes como registrarse, log-in, mostrar a los amigos en un mapa, seleccionar una foto de perfil visible por los demás usuarios, llamar por teléfono a un usuario o borrar permisos, hasta una nueva funcionalidad innovadora, poder observar las últimas posiciones en las que ha estado una persona, y no solo su última posición.

La arquitectura escogida para la aplicación ha sido una arquitectura de 3 capas común, con la particularidad de que debido a limitaciones de Android, ha sido necesaria la utilización de un puente intermediario para conectar la capa de lógica y la capa de persistencia. Este puente está construido mediante las tecnologías PHP y JSON, las cuales realizan una sencilla y eficiente comunicación, consistente en dos secciones principales: la parte ejecutada en el servidor remoto y la parte Android. La parte del servidor remoto consiste en ficheros PHP que se conectan a la base de datos, realizan una consulta y devuelven los resultados codificados en formato JSON (la codificación a este formato permite obtener y enviar los datos de forma segura, fluida y rápida). La parte Android se conecta con los PHP para obtener los datos y transforma la respuesta JSON a tipos de dato estándar, para poder trabajar con ellos.

Además, para optimizar la velocidad de la aplicación se ha optado por incorporar una BD Local SQLite, la cual almacena en el dispositivo móvil los amigos sobre los que tiene permiso el usuario y el propio permiso que tiene sobre los mismos. No se ha optado por almacenar en local las posiciones, ya que la naturaleza de esta aplicación

conlleva poder visualizar la posición de los usuarios en tiempo real, y para ello, lo más adecuado es descargar las posiciones cada vez que se muestre al amigo o amigos en un mapa, y no cargar posiciones antiguas almacenadas en local.

Sin duda, una de las características más cuidadas y trabajadas en GPSLoc es la ejecución en segundo plano de las funcionalidades más importantes de la aplicación. Tal y como se ha comentado anteriormente, Android paraliza las activities de las aplicaciones que quedan en segundo plano, y las va borrando aleatoriamente si hay escasez de memoria. Para solventar este inconveniente han sido implementados tres services, clases especiales de Android que continúan su ejecución en segundo plano y no son cerradas por el sistema operativo. Un service se dedica a captar la posición GPS del usuario y enviarla al servidor remoto, otro service se encarga de mantener la base de datos local actualizada regularmente y el último service se encarga de comprobar periódicamente si se han recibido invitaciones nuevas y, si es así, avisa al usuario mediante la barra de estado de Android con una notificación.

La elección de Android como sistema operativo destino para la aplicación ha resultado ser un acierto, en mi opinión personal. Cuando tomé las riendas de la tesina, en septiembre de 2010, Android era un sistema operativo nuevo, en constante evolución, y hoy es ya una realidad, habiendo desbancado a Symbian en cuota de mercado. Además, aparecen nuevas versiones del sistema operativo constantemente, cada una de ellas con nuevas mejoras cada vez más importantes, y conservando la retro compatibilidad con las aplicaciones diseñadas para versiones anteriores del sistema operativo, lo que asegura a los desarrolladores una gran amplitud de clientes potenciales.

Finalmente cabe destacar que, gracias al desarrollo de la aplicación GPSLoc, he adquirido una valiosa auto-formación en programación para el sistema operativo Android, pero no solo eso, sino también experiencia en PHP, XML, SQL y servidores remotos. Durante todos estos meses, he ido encontrando obstáculos y barreras que inicialmente no podía superar, como la inclusión de fotos en los perfiles de usuario, o la conexión con el servidor remoto. Gracias al esfuerzo dedicado para superar las dificultades y poder implementar todas las funcionalidades de la aplicación, he adquirido una experiencia muy valiosa de cara al futuro.

8. BIBLIOGRAFIA

[1] Aplicaciones móviles en el sector financiero, http://www.muycomputerpro.com/retos-que-se-hacen-realidad-las-aplicaciones-moviles-llegan-al-sector-financiero

[2] Localización con Google Latitude, http://onsoftware.softonic.com/localiza-tus-amigos-con-google-latitude

[3] Google Latitude, http://www.google.com/intl/es_es/latitude/intro.html

[4] Life 360 Actualidadgps, http://www.actualidadgps.com/2008/09/10/la-red-movil-de-emergencia-life360-gana-premio-google/

[5] Life 360, http://lasmejoresapps.com.mx/site/?p=426

[6] GPS Tracking, http://www.locimobile.com

[7] Foursquare, http://www.fayerwayer.com/2010/01/foursquare-todo-lo-que-necesitas-saber/

[8] User Interface Guidelines, http://developer.android.com/guide/practices/ui_guidelines/index.html

[9] Connecting to MySQL database, http://www.helloandroid.com/tutorials/connecting-mysql-database

[10] Eclipse Plugins, http://www.etish.org/services/plugins.html

[11] PHP, http://www.tuttymoran.com/publico/definiciones.php?cid=006

[12] Iphone, http://www.cristalab.com/tutoriales/fundamentos-de-programacion-para-iphone-c260l/

[13] Windows Phone 7, http://www.xpertos.tv/tecnologia/Articulos/tabid/214/Article/40/conoce-mas-de-cerca-el-nuevo-windows-phone-7.aspx

[14] Base 64, Robert Harder, http://iharder.sourceforge.net/current/index.html

[15] Wikipedia, http://es.wikipedia.org/

[16] Sunny Consolvo, Ian E. Smith, Tara Matthews, Anthony LaMarca, Jason Tabert, and Pauline Powledge. Location Disclosure to Social Relations: Why, When, & What People Want to Share.

[17] Carmen Ruiz Vicente, Dario Freni, Claudio Bettini and Christian S. Jensen. Location-Related Privacy in Geo-Social Networks.

[18] Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Patrick Kelley, Madhu Prabaker and Jinghai Rao. Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application.

ANEXO: MANUAL DE USUARIO GPSLOC

A continuación se presenta, a modo de anexo de la memoria, el manual de usuario para la aplicación GPSLoc, en el que se describe el manejo de la aplicación y sus funcionalidades.

MANUAL DE USUARIO

Universidad Politécnica de Valencia

Pablo Sáez Sáez

UNIVERSIDAD POLITECNICA DE VALENCIA

DSIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

ÍNDICE

1. INICIO DEL PROGRAMA

En la pantalla inicial de la aplicación, el usuario deberá introducir su Nombre de Usuario (Email) y su Password en los campos respectivos, tal y como se observa en la Figura 1. Es un requisito para la aplicación que el GPS del teléfono móvil este activado. Si un usuario intenta realizar log-in con el GPS desactivado, la aplicación instará al usuario a activarlo.

Figura 1 – Menú Inicial

Figura 2 – Cuadro de registro

Si el usuario no está registrado, podrá crear un nuevo usuario pulsando sobre la palabra *Register*, y rellenando los campos necesarios mostrados en la Figura 2, siempre que el correo introducido no esté ya dado de alta en el sistema.

Una vez el usuario hace log-in correctamente, en la barra de estado de Android aparecerá el icono del programa, y además, si el usuario despliega la barra de estado, podrá comprobar si tiene invitaciones pendientes.

2. LOCALIZAR AMIGOS

Una vez el usuario ha realizado log-in, la siguiente pantalla mostrada será el mapa de amigos, donde aparecerán sobre un mapa todos los usuarios sobre los que tiene permiso de visibilidad, para los cuales se cumpla el permiso en alguna de sus posiciones almacenadas en el servidor remoto. También será mostrado un punto rojo en el mapa, señalando la posición actual del usuario, si es que así lo ha indicado en la configuración, detallada en el punto 3 del manual. Hay que destacar que, tal y como se puede observar en la Figura 3, encima del mapa se muestran cinco pestañas, cada una de las cuales muestra una interfaz diferente para visualizar a los amigos según los criterios explicados a continuación, diferenciados por el icono de la pestaña:

Figura 3 – Mapa de amigos

-  Muestra el mapa con todos los amigos para los cuales alguna posición cumple el permiso que se posee sobre ellos.
-  Muestra la lista de todos los amigos
-  Muestra la lista de los amigos sobre los que se posee permiso total
-  Muestra la lista de los amigos sobre los que se posee permiso por horas
-  Muestra la lista de los amigos sobre los que se posee permiso por días

Al pulsar sobre un amigo en el mapa, será mostrado un cuadro de diálogo con su nombre, su imagen de perfil, su correo y el permiso que se tiene sobre él, y además, pulsando sobre el botón *Show Details*, se accederá a una nueva interfaz que mostrará la información detallada de cada amigo, tal y como se puede observar en la Figura 4. Esta interfaz ofrece la posibilidad de actualizar la posición de ese usuario, ver sus últimas posiciones, llamarle por teléfono, o borrar el permiso que se posee sobre él. También es posible acceder a los detalles de un amigo a través de las cuatro pestañas que listan los amigos según el tipo de permiso, para ello se debe pulsar sobre el nombre del amigo.

Pulsando la tecla menú desde una de las secciones principales de la aplicación, es decir, desde el área de amigos, el área de configuración o el área de invitaciones, se accede al menú principal de la aplicación, en el cual, se le ofrece al usuario la posibilidad de navegar por dichas secciones y realizar log-out de la aplicación. A continuación se procede a explicar la sección de configuración.

Figura 4 – Vista detallada de un amigo

3. CONFIGURACION

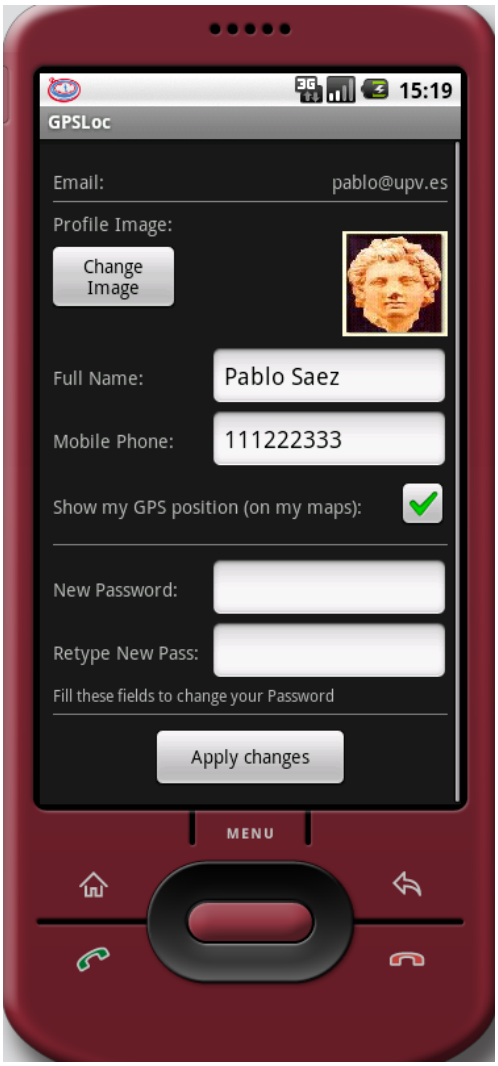

Figura 5 - Configuración

En la sección de configuración, tal y como se observa en la Figura 5, el usuario tiene la posibilidad de cambiar su imagen de perfil, seleccionando una imagen que ocupe un máximo de 100KB (es recomendable seleccionar imágenes de tamaño muy reducido debido a la naturaleza de la aplicación y la velocidad de acceso a Internet de las conexiones de dispositivos móviles actuales), modificar su nombre completo, su número de contacto, cambiar su contraseña, e indicar si desea mostrar su posición actual en los mapas o no.

Pulsando la tecla menú desde una de las secciones principales de la aplicación, es decir, desde el área de amigos, el área de configuración, o el área de invitaciones, se accede al menú principal de la aplicación, en el cual se le ofrece al usuario la posibilidad de navegar por dichas secciones y realizar log-out de la aplicación. A continuación se procede a explicar la sección de invitaciones.

4. INVITACIONES

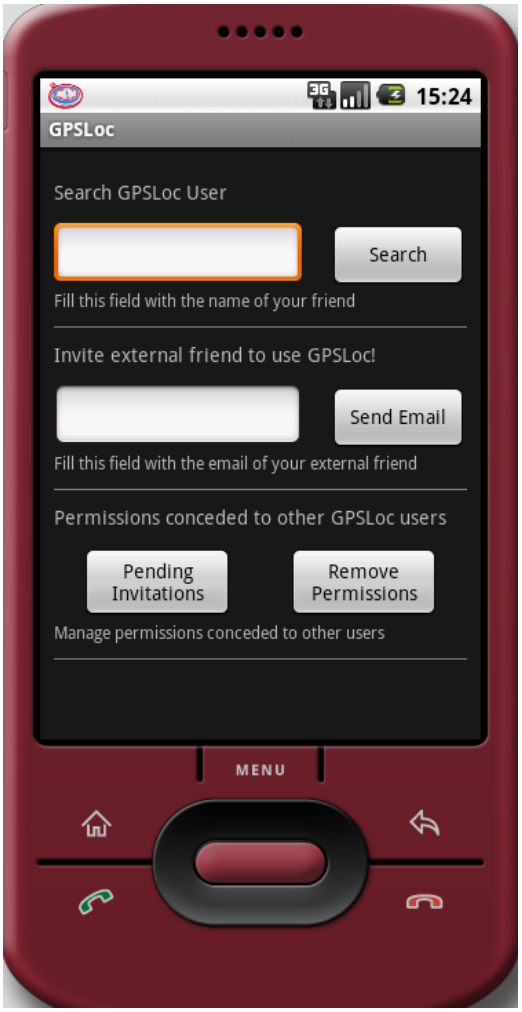

Figura 6 - Invitaciones

En esta sección, tal y como podemos observar en la Figura 6, el usuario tiene 4 opciones: buscar usuarios de GPSLoc para enviarles una invitación, invitar a un amigo externo a GPSLoc mediante el envío de un correo electrónico, gestionar las invitaciones pendientes y eliminar los permisos concedidos a otros usuarios. Estos puntos serán explicados a continuación.

4.1. BUSCAR Y ENVIAR INVITACIONES A USUARIOS DE GPSLOC

Introduciendo un nombre en el cuadro de texto inferior a "*Search GPSLoc User*" y pulsando el botón *Search*, GPSLoc buscará todos los usuarios cuyo nombre contenga el texto introducido. Es decir, si se introduce la cadena de texto "*bl*", aparecerán todos los nombres que contengan la secuencia de letras "bl", como por ejemplo: Pablo Sáez, Pablo Ruiz, etc. Si no existe ningún usuario registrado cuyo nombre completo contenga la secuencia de texto introducida, no aparecerá ningún resultado.

Una vez el usuario ha obtenido la lista de usuarios que cumplen con el criterio de búsqueda, pinchando sobre uno de ellos, podrá acceder a su información e imagen de perfil, y seleccionar el permiso que desea tener sobre él, indicando para cada tipo de permiso los parámetros necesarios, tal y como se observa en las imágenes 7 para permiso total, 8 para permiso por horas y 9 para permiso por días (las imágenes 8 y 9 son ligeramente más extensas debido a que es necesario desplazar la pantalla del móvil para visualizar toda la interfaz). Para el caso del permiso total, no será necesario completar ningún campo extra. Si es permiso por horas, es necesario indicar desde qué hora hasta qué hora, además de si se desea obtener visibilidad en fin de semana. Si es permiso por días, habrá que indicar el rango de días en los que se desea tener visibilidad sobre la posición del usuario al que se le va a enviar la invitación.

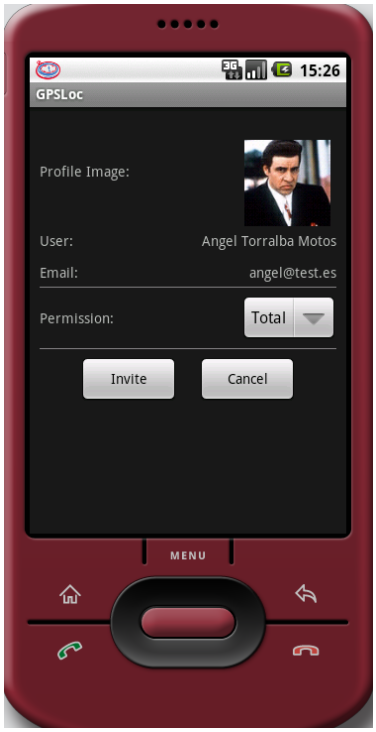

Figura 7 – Invitación permiso Total

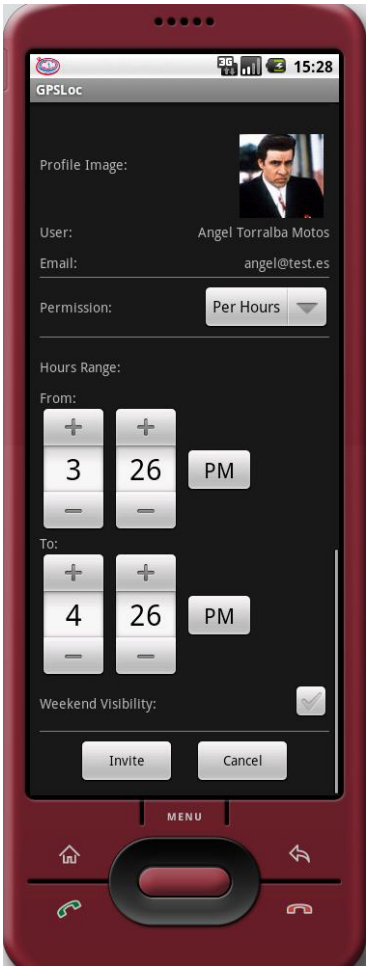

Figura 8 - Inv. permiso Por Horas

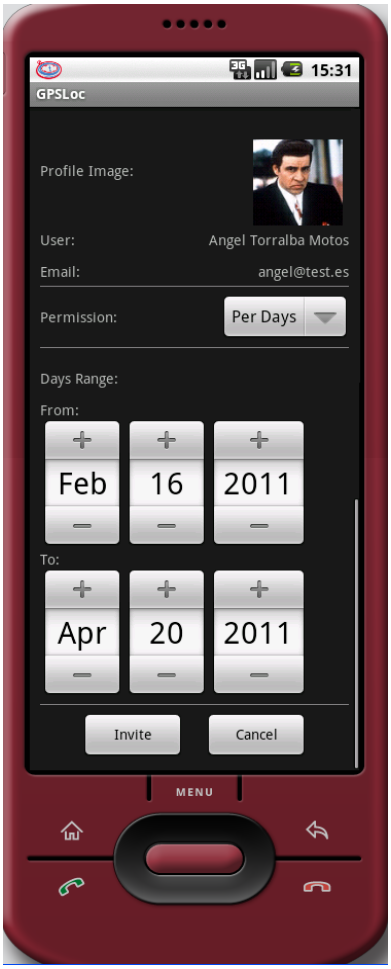

Figura 9 - Inv. permiso Por Días

4.2. INVITAR A UN USUARIO EXTERNO A GPSLOC MEDIANTE EMAIL

Rellenando con el email de un amigo el segundo campo que se observa en la Figura 6, inferior al texto *"Invite external friend to use GPSLoc"*, y pulsando el botón *Send email*, la aplicación pasará a enviar un correo electrónico a la dirección indicada utilizando el gestor de correo electrónico de Android, permitiendo al usuario modificar y personalizar el texto de la invitación, si así lo deseara, antes de confirmar el envío del email.

4.3. GESTIONAR LAS INVITACIONES PENDIENTES

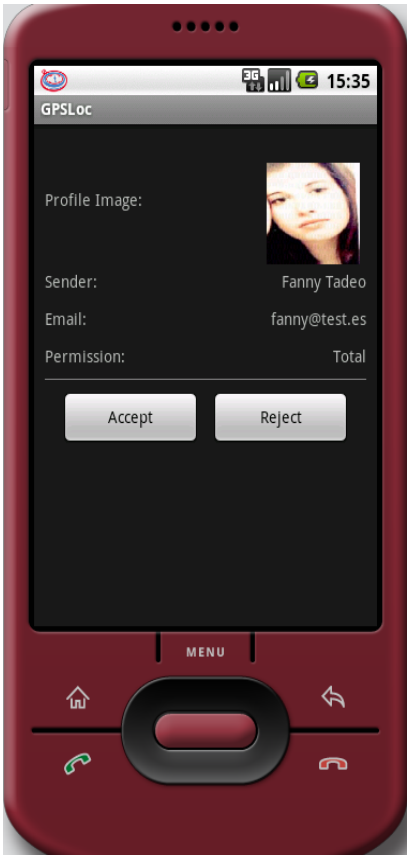

Figura 10 – Invitación pendiente

Pulsando en el botón *Pending Invitations* que se observa en la Figura 6, la aplicación mostrará una lista de los usuarios que han enviado al usuario logueado una invitación pendiente de aceptar o rechazar, y si no los hubiera, mostrará un mensaje avisando de que no existen invitaciones pendientes. En el caso de que sí existan, pulsando sobre una de ellas, será mostrada una nueva interfaz con la información del usuario que ha enviado la invitación, incluyendo su foto y las características del permiso que solicita con el usuario logueado, tal y como se observa en la Figura 10. El usuario logueado puede aceptar o rechazar la invitación mediante los dos botones de la parte inferior de la pantalla.

4.4. ELIMINAR PERMISOS CONCEDIDOS A OTROS USUARIOS

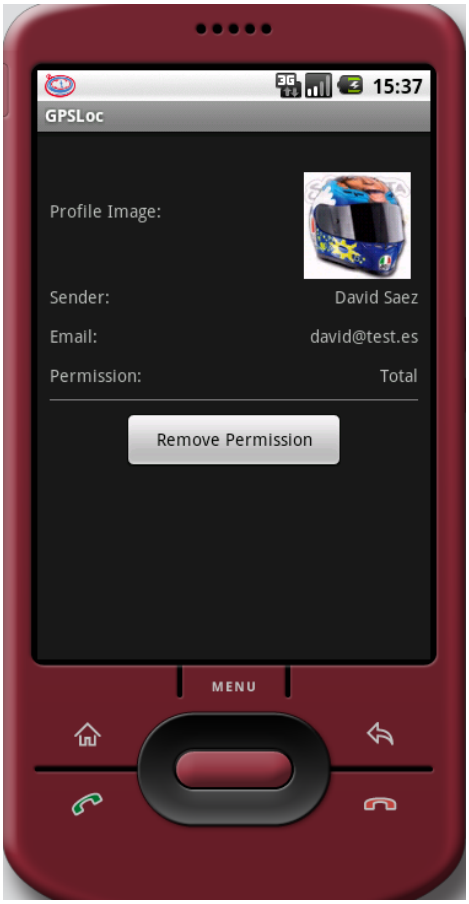

Figura 11 – Permiso concedido

Pulsando en el botón *Remove Permissions* que se observa en la Figura 6, la aplicación mostrará una lista de los usuarios a los cuales el usuario logueado ha concedido un permiso de visibilidad, y si no los hubiera, la aplicación mostrara un aviso de que no han sido concedidos permisos a ningún usuario. En el caso de que sí los hubiera, pulsando sobre uno de ellos, será mostrada una nueva interfaz con la información del usuario al que se le ha concedido el permiso, incluyendo su foto, y las características del permiso concedido, permitiendo al usuario logueado eliminarlo si así lo desea, tal y como se observa en la Figura 11.

5. LOG-OUT

Figura 12 - Menú

Pulsando la tecla menú desde una de las secciones principales de la aplicación, es decir, desde el área de visualización de amigos, el área de configuración, o el área de invitaciones, se accederá al menú principal de la aplicación, el cual, además de la posibilidad de acceder a dichas áreas, ofrece al usuario la opción de realizar log-out de la aplicación. En la Figura 12 se observa dicha opción.