



PROYECTO FINAL DE CARRERA

DISEÑO E IMPLEMENTACIÓN DE TRACKME,
APLICACIÓN MÓVIL DE GEOLOCALIZACIÓN
DE CONTACTOS

UNIVERSIDAD CARLOS III DE
MADRID

ESCUELA POLITÉCNICA
SUPERIOR

*INGENIERÍA DE
TELECOMUNICACIÓN*

Autor

Javier García Martín

Tutor

Julio Villena Román

Leganés, Octubre de 2015

Título: Diseño e implementación de Trackme, aplicación móvil de geolocalización de contactos

Autor: Javier García Martín

Director: Julio Villena Román

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

La realización de este proyecto no habría sido posible sin la ayuda, apoyo y ánimo de innumerables personas, que han estado presentes siempre de una forma u otra.

Primeramente, me gustaría dar gracias a mi tutor Julio, por haber sido tan amable y haberme ofrecido la oportunidad de trabajar con él en esta labor, estando atento y disponible cada vez que le he necesitado sin poner ningún inconveniente y dándome todas las facilidades a su alcance. Ha sido todo un placer.

Gracias también a mi familia, especialmente a mis padres, por haber sido los que me habéis aguantado todos los días con mis cosas buenas y malas, y que en los momentos bajos y duros de una carrera difícil habéis seguido empujándome, haciéndome ver que siempre hay luz al final del túnel. Gracias a mis tíos por seguir confiando en mí y ver siempre mis cosas positivas, dándome una perspectiva de la vida desde la experiencia que me ha servido de mucho, y a mis primos, especialmente al que ya no está, a quien siempre he tenido mucho cariño desde pequeño y me ha dado muchas alegrías. Gracias a mis abuelos, que ya tampoco están conmigo, porque me han transmitido todas las cosas buenas que ellos tenían y me han acompañado la mayor parte de mi vida, siendo un referente para mí entonces, ahora y en el futuro.

Agradecer también a mis amigos, que me han dado la oportunidad de compartir momentos buenos en épocas malas y han seguido ahí a pesar de las adversidades. Gracias en especial a Dani y Josu, por haber sido a los que he recurrido cuando he tenido alguna duda haciendo el proyecto y que siempre me habéis echado un cable dedicándome parte de vuestro tiempo, demostrando que al final esta etapa también me ha dado buenos amigos.

Gracias también a todos aquellos que han pasado por mi vida y me han hecho crecer como persona y a todos que los me pueda estar olvidando, porque como dice el refrán “uno puede devolver un préstamo de oro, pero está en deuda de por vida con aquellos que son amables”.

Gracias a todos desde lo más profundo del corazón.

Resumen

En los últimos tiempos, la necesidad y urgencia de las personas por estar en continuo contacto con personas de nuestro entorno, ha ido de la mano con la evolución de los dispositivos móviles, que ha dado lugar a que el tema de la geolocalización sea uno de los puntales actuales en el tema de las telecomunicaciones.

Aprovechando esto, se ha decidido desarrollar este proyecto, basado en la realización de una aplicación para dispositivos móviles para localizar en un mapa a un grupo de contactos.

Primeramente se ha estudiado el uso de tecnologías de geolocalización, profundizando más en aquellas que están inmersas en los terminales, junto con los sistemas operativos sobre los que se podría implementar.

El resultado final será la aplicación, desglosada al detalle, con las funcionalidades necesarias para poder encontrar, a través de la visualización en una proyección en el dispositivo, a aquellos familiares, amigos, compañeros, etc. que tengan instalada la misma aplicación.

Palabras clave: Geolocalización, Android, Aplicaciones móviles, Mapas

Abstract

In recent times, the need for people to know where other people are, has gone hand in hand with the evolution of the mobile devices, which has given rise to the field of the geolocation that is one of the mainstays in the current panorama of telecommunications.

Building on this, it has been decided to develop this project, based on the realization of an application for mobile devices to locate on a map to a group of contacts.

Firstly the use of geolocation technology has been studied, deeping in those that are immersed in the terminals, together with the operating systems on which could be deployed.

The end result will be the implementation, broken down to detail, with the functionality needed to be able to find, through the display on a projection on the device, to those colleagues who have installed the same application.

Keywords: Geolocation, Android, Mobile Apps, Maps

Índice

Índice.....	i
Tabla de Figuras	iv
1. Introducción y objetivos.....	1
1.1. Motivación y contexto.....	1
1.2. Objetivo	1
1.3. Organización del documento	1
2. Estado del Arte	3
2.1. Introducción	3
2.2. Aplicaciones basadas en localización	3
2.2.1. Introducción	3
2.2.2. Tecnologías existentes	4
2.2.3. Impacto en la sociedad.....	11
2.2.4. Nuevas tendencias	13
2.2.5. Seguridad y privacidad	14
2.3. Sistemas operativos en dispositivos móviles	17
2.3.1. Introducción	17
2.3.2. Tipos	17
2.3.3. Estudio de mercado, comparativa	25
2.3.4. Android, el sistema operativo dominante	26
3. Diseño de la aplicación.....	28
3.1. Introducción y objetivo del sistema Android	28
3.2. Casos de uso.....	29
3.2.1. Introducción	29
3.2.2. Enviar localización de posición.....	29
3.2.3. Seleccionar compañeros	30
3.2.4. Recibir localizaciones	30
3.3. Privacidad.....	31
4. Implementación	32
4.1. Introducción	32
4.2. Funcionalidad de la aplicación	32
4.2.1. Introducción	32
4.2.2. Diseño visual.....	33
4.2.3. Clases desarrolladas	33
4.2.4. Protocolo con servidor	34

4.3.	Diseño visual.....	36
4.3.1.	Introducción	36
4.3.2.	Layouts	36
4.3.3.	Menú de la actividad principal (MenuMain).....	45
4.4.	Flujo de funcionamiento de las clases	45
4.4.1.	Introducción	45
4.4.2.	Clase MainActivity	47
4.4.3.	Clase Contactos	64
4.4.4.	Clase ActualizarPosiciones	70
4.4.5.	Clase Settings	72
4.4.6.	Clase About	76
4.5.	Manifiesto	76
5.	Manual de usuario	79
5.1.	Introducción	79
5.2.	Manual	79
5.2.1.	Configuración inicial	79
5.2.2.	Ajustar la pantalla a los marcadores	84
5.2.3.	Modificar frecuencias de refresco.....	84
5.2.4.	Modificar número de teléfono.....	87
5.2.5.	Consultar la información	88
5.2.6.	Confirmar la salida.....	88
6.	Pruebas de funcionamiento	89
6.1.	Introducción	89
6.2.	Introducir número de teléfono	89
6.3.	Seleccionar contactos.....	89
6.4.	Añadir un contacto.....	90
6.5.	Cambiar número teléfono	91
6.6.	Refresh y Fit to markers	92
6.7.	Marcadores distinto color por última conexión.....	92
6.7.1.	Introducción	92
6.7.2.	Azul.....	92
6.7.3.	Amarillo	93
6.7.4.	Rojo	93
6.8.	Indicaciones de Google Maps	95
7.	Historia del proyecto y presupuesto	96
7.1.	Introducción	96

7.2.	Planificación	96
7.2.1.	Introducción	96
7.2.2.	Fases	96
7.2.3.	Diagrama de Gantt	97
7.3.	Presupuesto	97
7.3.1.	Introducción	97
7.3.2.	Horas totales	97
7.3.3.	Costes	98
8.	Conclusiones y trabajos futuros	99
8.1.	Introducción	99
8.2.	Privacidad	99
8.3.	Trabajos Futuros.....	100
8.3.1.	Mejoras	100
8.3.2.	Adaptación en dominios concretos.....	101
	Bibliografía	102

Tabla de Figuras

Figura 1: Aplicaciones de la información [2]	3
Figura 2: Escenario general del sistema GPS [6]	5
Figura 3: Funcionamiento de CGI-TA [6]	8
Figura 4: Esquema de triangulación UBA [6].....	10
Figura 5: Pirámide de las necesidades de Maslow [11]	11
Figura 6: Arquitectura de Android [13]	18
Figura 7: Arquitectura de iOS [16].....	21
Figura 8: Arquitectura de Windows Phone [18].....	23
Figura 9: Evolución de la cuota de mercado de los sistemas operativos más significativos [13]26	
Figura 10: Estructura del escenario del proyecto	28
Figura 11: Envío de localización de posición	29
Figura 12: Selección de los compañeros	30
Figura 13: Recepción de localizaciones.....	31
Figura 14: Diagrama de flujo de clases.....	34
Figura 15: Layout asociado a la actividad principal.....	37
Figura 16: Layout asociado a la actividad Contactos	38
Figura 17: Layout asociado a la actividad Settings.....	39
Figura 18: Layout asociado a la actividad About.....	39
Figura 19: Layout asociado al cuadro de diálogo para activar la localización.....	40
Figura 20: Layout asociado al cuadro de diálogo de introducir el número.....	41
Figura 21: Layout asociado al cuadro de diálogo de mostrar el número.....	42
Figura 22: Layout asociado al cuadro de diálogo de editar el número	42
Figura 23: Layout asociado al cuadro de diálogo de mostrar la frecuencia de refresco de la posición	43
Figura 24: Layout asociado al cuadro de diálogo de modificar la frecuencia de refresco de la posición de los contactos	44
Figura 25: Layout asociado al cuadro de diálogo de salir de la aplicación.....	44
Figura 26: Layout asociado al menú principal.....	45
Figura 27: Diagrama de flujo de una actividad Android.....	46
Figura 28: Ciclo de vida del método "onCreate" de la actividad principal.....	49
Figura 29: Ciclo de vida del método "onResume" de la actividad principal	50
Figura 30: Ciclo de vida del método "onStop" de la actividad principal	51
Figura 31: Ciclo de vida del método "onRestart" de la actividad principal.....	52
Figura 32: Ciclo de vida del método "onDestroy" de la actividad principal.....	53
Figura 33: Ciclo de vida del método "onOptionsItemSelected" de la actividad principal	54
Figura 34: Ciclo de vida del método auxiliar "inicializarMapa" de la actividad principal.....	55
Figura 35: Ciclo de vida del método auxiliar "obtenerPosicion" de la actividad principal.....	56
Figura 36: Ciclo de vida del método auxiliar "moverCamara" de la actividad principal	57
Figura 37: Ciclo de vida del método auxiliar "representarColegas" de la actividad principal	58
Figura 38: Ciclo de vida del método auxiliar "addMarcador" de la actividad principal	59
Figura 39: Ciclo de vida del método auxiliar "removerMarcadores" de la actividad principal...	59
Figura 40: Ciclo de vida del método auxiliar "enviarLocalizacion" de la actividad principal	60
Figura 41: Ciclo de vida del método auxiliar "comprobarNumero" de la actividad principal.....	62
Figura 42: Ciclo de vida del método auxiliar "pedirPosiciones" de la actividad principal	63
Figura 43: Ciclo de vida del método auxiliar "refresh" de la actividad principal	64
Figura 44: Ciclo de vida del método auxiliar "ajustarMarcadores" de la actividad principal	65

Figura 45: Ciclo de vida del método auxiliar "activarLocalizacion" de la actividad principal	66
Figura 46: Ciclo de vida del método auxiliar "onKeyDown" de la actividad principal	67
Figura 47: Ciclo de vida del método "onReceive" de la clase ReceptorEventos dentro de la actividad principal	68
Figura 48: Ciclo de vida del método "onCreate" de la clase Contactos	69
Figura 49: Ciclo de vida del método "onListItemClick" de la clase Contactos	69
Figura 50: Ciclo de vida del método "enviarColegas" de la clase Contactos	70
Figura 51: Ciclo de vida del método "onHandleIntent" de la clase ActualizarPosiciones.....	71
Figura 52: Ciclo de vida del método "viewNumber" de la clase Settings	73
Figura 53: Ciclo de vida del método "setNumber" de la clase Settings	73
Figura 54: Ciclo de vida del método "refreshPositionFrequency" de la clase Settings.....	74
Figura 55: Ciclo de vida del método auxiliar "setPositionFrequency" de la clase Settings.....	75
Figura 56: Pantalla de habilitar la localización	80
Figura 57: Pantalla de introducción del número de teléfono por primera vez.....	80
Figura 58: Pantalla de introducir número de teléfono con teclado numérico	81
Figura 59: Pantalla principal de la aplicación	81
Figura 60: Pantalla de despliegue del menú superior	82
Figura 61: Pantalla de ajustes "Settings"	82
Figura 62: Pantalla de selección de contactos	83
Figura 63: Pantalla una vez que se han añadido compañeros	83
Figura 64: Pantalla ajustada al pulsar "Fit to markers"	84
Figura 65: Pantalla de consulta de la frecuencia de refresco de la posición	85
Figura 66: Pantalla para introducir una nueva frecuencia de refresco de la posición.....	85
Figura 67: Pantalla que muestra la frecuencia de refresco de los contactos	86
Figura 68: Pantalla para modificar la frecuencia de refresco de los contactos	86
Figura 69: Pantalla que muestra el número propio	87
Figura 70: Pantalla para modificar el número propio	87
Figura 71: Pantalla de About	88
Figura 72: Pantalla de confirmación de cierre	88
Figura 73: Prueba de funcionamiento de introducir número de teléfono	89
Figura 74: Prueba de funcionamiento de seleccionar contactos	90
Figura 75: Prueba de funcionamiento de añadir un contacto	90
Figura 76: Prueba de funcionamiento de modificar el número de teléfono	91
Figura 77: Prueba de funcionamiento de "refresh" y "fit to markers"	92
Figura 78: Prueba de funcionamiento de última conexión menor de una hora (marcador azul)	93
Figura 79: Prueba de funcionamiento de última conexión mayor de una hora en el mismo día (marcador amarillo)	94
Figura 80: Prueba de funcionamiento de última conexión mayor de un día (marcador rojo) ...	94
Figura 81: Prueba de funcionamiento de las indicaciones de Google Maps	95
Figura 82: Diagrama de Gantt	97
Figura 83: Horas empleadas en el proyecto.....	97
Figura 84: Coste relativo a los participantes	98
Figura 85: Coste relativo a los equipos	98
Figura 86: Coste Total del Proyecto	98

1. Introducción y objetivos

1.1. Motivación y contexto

En los últimos años, el fenómeno de la geolocalización se ha convertido en una realidad, de la que todos somos partícipes. Con el simple gesto de sacar de nuestro bolsillo el teléfono móvil, somos capaces de conocer nuestra posición automáticamente en un mapa, crear una ruta hacia otra, o incluso compartirla para ser visibles a los demás. Mirando hacia atrás en el tiempo, esto no resultaba tan trivial, debido a que se ha ido produciendo una evolución.

La necesidad psicológica de comunicación del hombre con sus congéneres, ha sido palpable en todas las etapas de la evolución humana. En la antigüedad, empezó a ser vital la comunicación a larga distancia y se emprendieron distintas formas de hacerlo [1], principalmente a través de señales de humo, que gozaban de gran precisión para localizar lugares, y los mensajes con pájaros (en mayor parte palomas), que eran entrenados para encontrar su hogar desde largas distancias. También se desarrolló, en mayor medida en la marina, la navegación celestial, midiendo matemáticamente el ángulo del sol o las estrellas para determinar la latitud, que fue mejorado en el siglo XVIII con la medición de la longitud a través de cronómetros que medían el cambio de tiempo entre el punto de partida y la localización actual. Aunque el mayor descubrimiento fue el de la brújula, que dio lugar a una navegación más precisa.

Ya en el siglo XX, la triangulación de la posición a través de la medición de la intensidad de las señales radio, fue un gran avance en el terreno militar, que posteriormente dio el paso al civil. La evolución hacia los satélites GPS y su uso en el terreno civil, los cuales usan la triangulación junto con la medición del tiempo, ha dado lugar a que hoy día podamos disfrutar de estas ventajas en nuestra vida cotidiana.

1.2. Objetivo

El objetivo del proyecto es diseñar, implementar y probar una aplicación para dispositivos móviles, llamada TrackMe, que consiste en geolocalizar a un grupo de personas, pudiéndolas localizar en todo momento mediante un mapa mostrado por pantalla.

Primeramente se analizarán las diferentes tecnologías existentes, su impacto en la sociedad, las tendencias y sus niveles de seguridad, buscando a la par las mejores características de implementación, en función de los diferentes tipos de sistemas operativos que predominan en el mercado, para finalmente llevar a cabo el desarrollo de la aplicación con sus correspondientes pruebas.

1.3. Organización del documento

El presente documento comenzará con la explicación del Estado del Arte (capítulo 2), en la que se explicarán las diferentes aplicaciones basadas en localización junto con los sistemas operativos que se portan en los dispositivos móviles actualmente. Más específicamente, este capítulo se dividirá en dos bloques principalmente, que se explican a continuación.

En el primer bloque se describirán las principales aplicaciones basadas en localización existentes en la actualidad, donde se explicarán los sistemas de geolocalización por satélite,

pseudosatélites, redes celulares y redes wifi, apuntando también otros con menor relevancia. También se explicará el impacto, tanto económico como social, que éstos crean en la sociedad, y se expondrán también las nuevas tendencias de futuro. Además, se analizarán privacidad y seguridad en la información enviada por este tipo de redes y las técnicas que hoy en día, se presumen como las más fiables para consolidarlas. Y por último se particularizarán las técnicas de geolocalización de las que hoy en día disponen los dispositivos móviles.

La segunda parte tendrá como objetivo la descripción de los diferentes sistemas operativos para móviles, junto con una comparativa de la evolución de éstos en el mercado, justificando la elección de Android como sistema operativo predominante.

Una vez desglosado el contexto actual, en el capítulo 3 se procede a explicar el diseño de la aplicación en sí, mediante una introducción del motivo de la aplicación, sus diferentes casos de uso así como una referencia a la privacidad.

A continuación, el capítulo 4 detalla la implementación, entrando más a fondo en temas de programación Android y XML, con una explicación didáctica a través de diagramas de flujo así como con las pantallas diseñadas. También se explicará el manifiesto de la aplicación, que básicamente dará los permisos necesarios para el correcto funcionamiento.

Una vez hecha la presentación de cómo se ha realizado la aplicación, en el capítulo 5 se explica en forma de manual de usuario de una forma didáctica cómo utilizar la aplicación.

Ya en el capítulo 6 se procede a mostrar las diferentes pruebas de funcionamiento consumadas, explicando las distintas funcionalidades implementadas.

El capítulo 7 dará una visión conjunta de la historia del proyecto, mostrando su planificación con el desglose de las horas dedicadas así como los costes totales.

Finalmente se incluyen en el capítulo 8 las conclusiones que ha reportado la realización de este proyecto, en distintos aspectos como las posibles mejoras, trabajos futuros y privacidad.

Cerrando el documento se añade la bibliografía utilizada de consulta.

2. Estado del Arte

2.1. Introducción

En este capítulo se hará una presentación sobre las diferentes tecnologías de geolocalización existentes, en las que se profundizará sobre las características de las técnicas más asentadas, su impacto en la sociedad, las nuevas tendencias emergentes y se explicarán al hilo los asuntos de seguridad y privacidad. Finalmente se explicarán los diferentes sistemas operativos existentes para dispositivos móviles, tanto a nivel técnico como de mercado.

2.2. Aplicaciones basadas en localización

2.2.1. Introducción

La localización se ha convertido en un tema esencial en cuanto al estilo de vida de las personas [2]. El hecho de poder conocer tu posición en cada instante da lugar a un sinnúmero de nuevos desarrollos, no sólo creando rutas hacia un destino sino también en otros campos, como por ejemplo en E-health, micro pagos, publicidad más eficiente o los emergentes “wearables”. Además la implementación y gran crecimiento de estas tecnologías en dispositivos móviles, ha favorecido a su desarrollo e implementación a gran escala. A continuación se muestra un esquema en el que se pueden observar algunas funcionalidades que proporciona la localización.

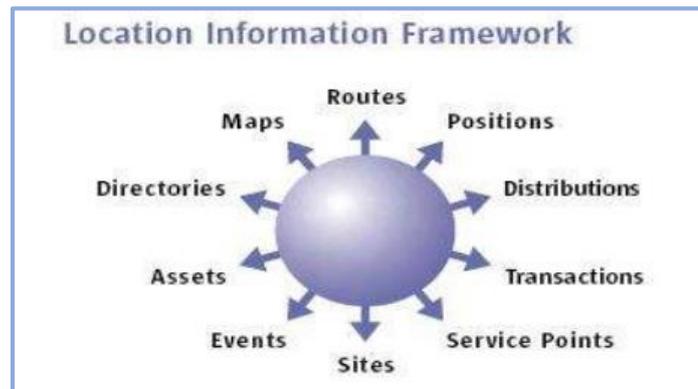


FIGURA 1: APLICACIONES DE LA INFORMACIÓN [2]

Es importante hacer hincapié en la convergencia que se ha producido de diversos campos, como son la geolocalización y las telecomunicaciones para dar lugar a este fenómeno, que ha posibilitado combinar la adquisición, el análisis y la diseminación de la información espacial, para conseguir propósitos específicos. Las tecnologías tradicionales como pueden ser la captación de imágenes por satélite o aeronaves, los satélites GNSS y los sistemas de información geográfica, se están complementando hoy día con nuevos desarrollos que posibiliten nuevas formas de posicionamiento. WIFI (redes de acceso a Internet fijo), LIDAR (posicionamiento Láser) y las redes de sensores son claros ejemplos de estos nuevos sistemas, junto por supuesto las redes de banda ancha móvil, cada vez más sofisticadas.

Haciendo uso de estos desarrollos en el creciente mercado de los dispositivos móviles se ha llegado a lo que hoy día se conoce como LBS (Location-Based Services), que orienta localización y dispositivos a los usos más demandados en la actualidad, como juegos, asistencia en navegación, seguridad personal, respuesta ante emergencias, comercio electrónico, etc.

A continuación se procede a resumir los rasgos principales de las tecnologías de mayor trascendencia en la actualidad.

2.2.2. Tecnologías existentes

2.2.2.1. Introducción

Hoy en día existen multitud de tecnologías de geolocalización. En este apartado se verán las más importantes, explicando sus principales características así como sus variantes más importantes. Por último, se expondrán los rangos principales de tecnologías menos influyentes pero con cierta repercusión.

2.2.2.2. Global Navigation Satellite Systems (GNSS)

GNSS [3] [4] [5] [6] es el término genérico de los sistemas de navegación por satélite que proporcionan posicionamiento tridimensional y temporal (sincronización), en cualquier momento del día y condición meteorológica mediante técnicas de radiocomunicaciones de múltiples satélites. Sólo dos de los tipos de GNSS han conseguido funcionar de forma completa, siendo estos el GPS, de creación militar estadounidense, y el GLONASS, de elaboración rusa. El resto se encuentran aún en desarrollo y fases experimentales.

2.2.2.2.1. Global Positioning System (GPS)

El sistema GPS [7] [6] [8] tiene su origen en el Departamento de Defensa Norteamericano, que en 1973 inició el proyecto de NAVSTAR (NAVigation System Timing And Ranging). A partir de entonces, desde que se lanzó el primer satélite en 1978, se trabajó para conseguir que estuviera totalmente operacional, lo cual se consiguió en 1993.

Principalmente, los objetivos que se buscaban eran:

- Disponibilidad para todo tipo de condiciones (tierra, mar, aire, espacio) y usuarios.
- Posicionamiento y velocidad en tiempo real.
- Restricciones para uso militar de la mejor precisión.
- Uso de redundancia para el caso de que haya problemas en el sistema.
- Uso de unidades de bajo coste y potencia.

La constelación de satélites está compuesta por 24 unidades de los que 3 son redundantes, asegurando así en cada zona de la Tierra una visibilidad de entre cinco y ocho satélites de forma simultánea. Según se ha ido evolucionando el sistema, se han ido añadiendo nuevos satélites con nuevas funcionalidades adecuadas a las nuevas demandas.

Su configuración se puede descomponer en tres segmentos: El del satélite, que es la constelación de satélites en sí misma; el de control, que se encarga del mantenimiento del sistema y funciona conjuntamente con estaciones terrenas; y el de usuario, que son aquellos que disponen de un receptor. A continuación se muestra una representación gráfica del escenario.

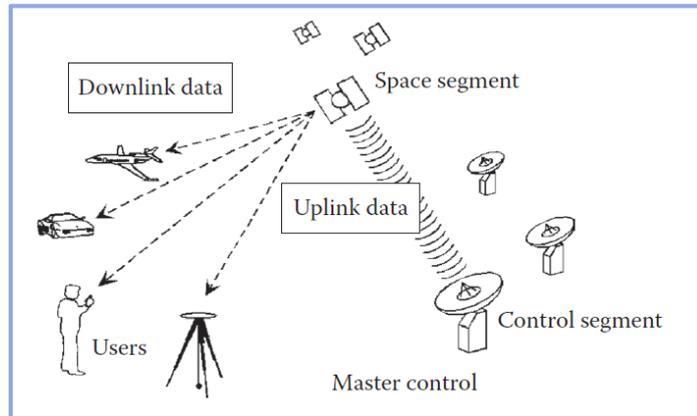


FIGURA 2: ESCENARIO GENERAL DEL SISTEMA GPS [6]

En cuanto a las señales, están formadas por tres tipos de componentes. Éstas son por un lado portadoras, junto al mensaje y un ruido añadido pseudo-aleatorio (PRN). Los códigos PRN a su vez pueden ser de dos tipos: P (preciso), de uso militar cifrado que hace uso de dos portadoras; y C/A (Adquisición gruesa) que tiene connotación civil y sólo utiliza una de ellas. Actualmente se encuentra en desarrollo un nuevo tipo de señalización L2C que podrá dotar de más precisión a los usuarios cotidianos. El sistema también dispone de tratamiento de la señal para tratar los errores sistemáticos y de propagación.

La manera de posicionar al objetivo, el sistema utiliza la trilateración en el espacio, basada en la medida de un rango (pseudo-rango) entre el receptor y los satélites. Este posicionamiento se realiza con tres de los satélites, pero para salvar el problema del retardo en el envío y adquisición de las señales, es preciso el uso de un cuarto satélite que sirva de sincronizador de los relojes y corrector de tiempos. Existen dos tipos de posicionamientos, el absoluto y el relativo. El posicionamiento puntual o absoluto, funciona de manera que el receptor determina su localización mediante la medida del pseudo-rango de varios satélites que determinan sus coordenadas y el término de corrección de reloj. Por su parte, el relativo, emplea al menos dos satélites, junto con una estación base de referencia, para determinar la localización relativa del receptor.

En cuanto a la precisión del sistema, solo notar que depende de distintos factores, como pueden ser el modo de observación, el tipo, el nivel de errores, que variarán el tipo de respuesta.

2.2.2.2.2. *GLONASS*

Se trata del sistema ruso de navegación [6], que usa la mayoría de principios del GPS en cuanto a transmisión de señales y posicionamiento. Se diferencian principalmente en las frecuencias de portadora (GLONASS usa distintas para distinguir cada satélite), en que los satélites de ese sistema usan los mismos códigos (GPS usa diferentes en cada satélite), y en los datos que éstos envían para la localización y mantenimiento.

Hoy en día, este sistema funciona como complemento del GPS para una mejorar las señales en los receptores en aplicaciones de alta precisión.

2.2.2.2.3. *GALILEO*

Es el sistema de navegación europeo [6], coordinado por la Unión Europea y la Agencia Espacial Europea (ESA). Su despliegue total consistirá en 30 satélites, que orbitarán al nivel medio. El origen del proyecto es la creación de un sistema de propósito civil pero aun así será compatible con GPS. Actualmente, este sistema está en fase de pruebas y se están lanzado satélites gradualmente para empezar a prestar servicio en un futuro cercano.

2.2.2.2.4. *COMPASS*

El servicio chino de satélites fue creado en 2003 [6] y con el lanzamiento de éstos junto con el desarrollo del servicio, se declaró su funcionamiento en 2011. Está previsto que para 2020 se despliegue la constelación completa de 35 satélites, que presumiblemente será compatible con receptores GPS, GLONASS y GALILEO. Ofrecerá cuatro servicios que serán:

- Posicionamiento abierto, con un nivel de precisión de 10 metros.
- Servicio autorizado (uso militar)
- Posicionamiento diferencial (o apoyado en una estación terrena) en áreas extensas con precisión de un metro.
- Servicio de SMS de 120 caracteres chinos.

2.2.2.2.5. *Sistemas de augmentación de satélite*

Se han presentado los sistemas de navegación por satélite más importantes, y faltaría por nombrar los que están un paso por detrás que serían [6]:

1. QZSS (Quasi-Zenith Satellite System), o el sistema japonés de navegación. Aún se encuentra en fase de desarrollo y no tiene suficientes satélites en órbita para poder funcionar.
2. GAGAN (GPS Aided Geo Augmented Navigation), que se trata de un proyecto indio de mejora de la señal GNSS para el uso de la aviación civil sobre territorio indio.

2.2.2.3. Pseudosatélites

Un pseudo-satélite o pseudolite (PL) es algo que se asemeja a un mini-satélite [6], que puede proporcionar posicionamiento autónomo en entornos interiores o exteriores. El sistema es capaz de triangular la posición de un objeto midiendo las distancias desde el objeto a un array de pseudolites, cuyas coordenadas son conocidas. Además de para localización, este servicio también se utiliza para sistemas precisos de aterrizaje y otros sistemas de augmentación.

Anteriormente se definía pseudolite como “algo que se asemeja a un satélite”, y esto quiere decir que, tomando el ejemplo de localización outdoor, puede ser una antena posicionada en lo alto de un edificio, que mejore la señal GNSS en situaciones o áreas donde la constelación es insuficiente. Además, los PL pueden ser diseñados tanto para transmitir como para recibir señales, incluso consiguiendo determinar su propia localización. Los dispositivos GPS, también pueden ser usados como rastreadores PL, mediante modificación firmware. También hay que tener en cuenta que estos sistemas no tienen problemas de errores ionosféricos o troposféricos, pero sí que son sensibles a multitrayecto o near-far.

2.2.2.4. Redes celulares

La técnica más usada en este tipo de redes es la localización por triangulación, basada o bien en el ángulo de llegada o en la observación de la distancia, entre el terminal móvil y las estaciones base [6].

A su vez se puede dividir en otros tres tipos:

1. Posicionamiento Centrado en el terminal: está basado en la instalación de un software en el dispositivo móvil, que hace que él mismo sea capaz de determinar su posición. Éste a su vez se puede dividir en otros tres tipos:
 - a) *Método GPS*: Usa señales de varios satélites GPS y proporciona posicionamiento instantáneo, con una resolución de 5 a 50 metros.
 - b) *A-GPS (Assisted)*: Utiliza una red asistente de receptores GPS que proporcionan información, reduciendo el tiempo necesario para posicionarse.
 - c) *E-OTD (Enhanced observed time difference)*: Mide el tiempo de llegada de la señal para múltiples estaciones base. La diferencia de tiempos entre ellas determinan la posición del terminal, siendo conocidas las coordenadas de éstas.
2. Posicionamiento Centrado en la red e híbridos: Se basan, en rasgos generales, en que la estación base determina la posición el usuario. A su vez también se subdivide en seis métodos:
 - a) *CGI-TA (Cell Global Identity with Timing Advance)*: Usa el identificador de celda para localizar al usuario. El Timing Advance es una información suplementaria ofrecida en las tramas intercambiadas para ajustar el tiempo de envío de estas y así obtener una mejor precisión. En la Figura 3, se puede observar la precisión adquirida

mediante la conjunción de identidad y sector de celda, junto con el timing advance.

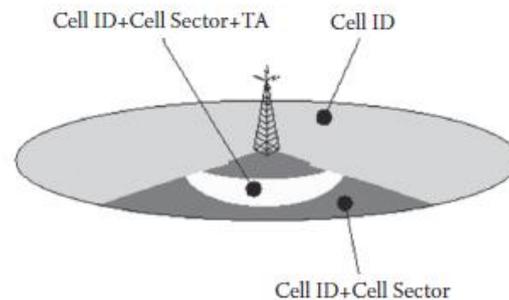


FIGURA 3: FUNCIONAMIENTO DE CGI-TA [6]

- b) *TOA (Time Of Arrival): Funciona comprobando el tiempo de vuelo de la señal, triangulando el objetivo entre al menos tres estaciones base.*
 - c) *TDOA (uplink Time Difference Of Arrival): Tiene un comportamiento similar a E-TOD, salvo que los tiempos dejan de medirse en el dispositivo y pasan a hacerlo en la red.*
 - d) *AOA (Angle Of Arrival): Está basado en la observación del ángulo de llegada, por el menos dos estaciones base. Éstas lo mandan a un equipo AOA que se encarga de calcular la posición.*
 - e) *Búsqueda de patrones de coincidencia de localización (multitrayecto): La estación base dispone de una base de datos que contiene las características de las señales multitrayecto que recibe. Compara la señal recibida y puede establecer su posición en base a sus características.*
 - f) *RSS (Received Signal Strength): El sistema analiza la señal recibida y basándose en los umbrales de intensidad de señal medida por un conjunto de estaciones base, posiciona al usuario.*
3. Precisión GSM y UMTS: El sistema GSM hace uso de técnicas CDMA, mientras que UMTS utiliza W-CDMA (wideband-code division multiplex access). Este método es el mismo que utilizan los satélites en técnicas de navegación, dando lugar a periodos de comunicación vacíos en el enlace descendente, que permite a todos los receptores recibir la señal y al transmisor ver las diferencias de tiempo de llegada. Esto permite crear métodos de triangulación avanzados.

2.2.2.5. Wifi

Las redes locales inalámbricas (WLAN) funcionan a 2,4 GHz [6] y el uso de éstas para el posicionamiento resulta muy favorable debido a su generalización y disponibilidad, ya que la gran mayoría de edificios y dispositivos tienen implementada su funcionalidad. Aquí podemos diferenciar dos métodos, de los que se puede también implementar una combinación de ambos:

2.2.2.5.1. *Trilateración*

Su funcionamiento es similar a GPS, donde se hace uso de como mínimo tres puntos de acceso que se usan en combinación con valores RSS. Hay que reseñar, que las señales Wifi son muy sensibles a la interferencia causada por la variedad de caminos que puede seguir la señal, el entorno,... lo que puede causar un posicionamiento erróneo del usuario.

2.2.2.5.2. *Fingerprinting*

Este método es más eficaz que el de trilateración pero necesita crear bases de datos que contengan la potencia de la señal de varios puntos de acceso (AP), de ubicación conocida. Al comparar las señales recibidas con las almacenadas, se elimina el problema de los desajustes que existen en trilateración.

El sistema operativo Android utiliza este método de localización, accediendo a la base de datos de Google, y comparando así las distintas señales de un conjunto de usuarios.

También se está extendiendo un uso conjunto de esta técnica con el escaneo de señales wifi y el uso de información física (mapas), predominante en lugares con multitud de antenas, que son capaces de posicionar a niveles entre 1 y 3 metros. Así se consigue un gran incremento de la técnica de Fingerprinting.

2.2.2.6. *Otros*

2.2.2.6.1. *Ultra-Banda Ancha*

Este tipo de señales de gran banda ancha y precisión en el sincronismo, tienen grandes mejoras en cuanto a interferencias debidas a multicamino y penetración y también mejora la precisión de rango [6]. A nivel europeo funciona en el rango de (6 - 8,5 GHz) y opera para distancias cortas (menores de 100 metros) haciendo uso de trilateración generalmente.

En la siguiente figura se puede apreciar un ejemplo de triangulación a partir de señales de Ultra-Banda Ancha, donde se muestran las inconveniencias producidas por rebotes y penetraciones que se solucionan con este tipo de señales.

2.2.2.6.2. *RFID (Identificación por radiofrecuencia)*

Este tipo de comunicaciones necesita el despliegue de scanners RFID a lo largo del entorno operacional, que interrogarán a parches desplegados (ya sean activos o pasivos) que estén conectados al objeto sujeto a medición [9] [6]. El hecho de que los parches o etiquetas sean activos o pasivos influye en el rango de medición, ya que en los activos es mayor. Esto es una característica importante a la hora de decidir la técnica de localización a utilizar, aunque principalmente se hace uso de CoO (Cell Of Origin), que está basado en la identificación a partir de una celda delimitada por el máximo

rango de lectura de una etiqueta. Como se puede vislumbrar, la precisión de esta técnica depende en gran parte del número de etiquetas desplegadas, pero al mismo tiempo incrementa el coste en cantidad. Por tanto, finalmente el uso más común es el fundamentado en triangulación mediante la potencia de la señal recibida, pudiéndose mejorar incluso mediante fingerprinting.

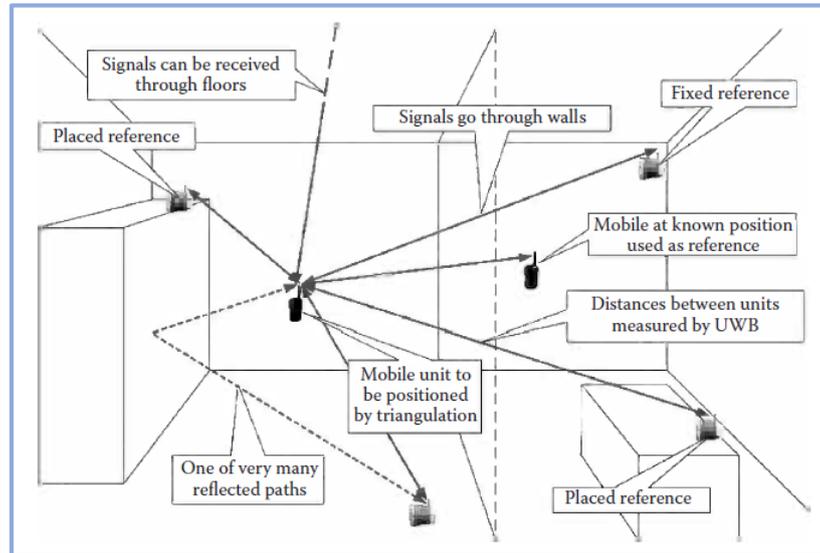


FIGURA 4: ESQUEMA DE TRIANGULACIÓN UBA [6]

2.2.2.6.3. Inerciales, digitales, etc.

Existen otros tipos de localización con mucho menor uso (o muy específico de algún sector) y penetración en el mercado [6].

Entre ellos se encuentran los sistemas inerciales, que se caracterizan por utilizar la información proporcionada por sensores montados en plataformas móviles, lo que lleva a determinar la posición instantánea mediante la medición de las distintas fuerzas, velocidad angular y tiempo.

También tiene su importancia el sistema Digital Compass. Éste está basado en sensores eléctricos y magnéticos con los que, a partir de las mediciones sobre el campo magnético de la Tierra, proporciona una indicación digital de la dirección.

Luego ya por último para posicionamiento en entornos Indoor, hay otros sistemas menos precisos como seguimiento por ultrasonidos, magnético u óptico, que no gozan de las ventajas de los expuestos con anterioridad y por tanto su implantación es menor.

2.2.3. Impacto en la sociedad

2.2.3.1. Introducción

La geolocalización se ha convertido en un aspecto clave, no sólo en temas que llevan ya varios años en auge, como por ejemplo la navegación GPS, sino también en otros que hoy en día tienen mucha repercusión [10]. Por ello, a continuación se estudiará el impacto que ha tenido en la sociedad este tipo de tecnología, tanto a nivel social como económico.

2.2.3.2. Impacto social

La interacción de las personas, sobretodo en redes sociales, ha dado lugar al fenómeno de la compartición de la posición actual, lo cual viene dado por la integración de éstas en nuestras vidas. Los usuarios ven en el hecho de compartir su posición para, de cierto modo sentirse recompensados. Por ejemplo, pueden producir encuentros inesperados, dar lugar a aficiones comunes, descubrir nuevos espacios, recibir promociones o el competir por conocer más lugares.

La geolocalización también ha dado lugar a la satisfacción de necesidades por parte de la sociedad. El ser humano fue descrito por Maslow a mediados del siglo XX a partir de una pirámide de necesidades [10], que podrían encuadrarse también en este contexto. Esta teoría se basaba en que las necesidades del hombre están encuadradas en una pirámide, y para llegar a satisfacer alguna de ellas, primeramente se deben cubrir las de niveles inferiores, como se puede apreciar en la figura.

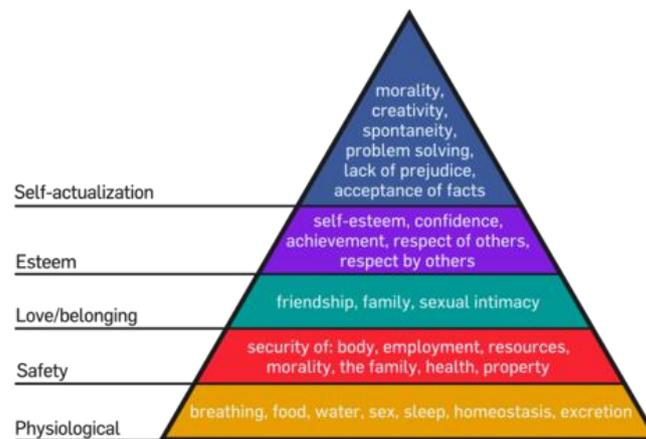


FIGURA 5: PIRÁMIDE DE LAS NECESIDADES DE MASLOW [11]

Para el caso de la geolocalización las necesidades se especificarían de la siguiente forma:

- *Funcionales:* Si nos encontramos en lugar desconocido, las redes de geolocalización nos ayudan a hallar el lugar al que queremos acudir.

- *Seguridad:* Si un lugar tiene referencias buenas o malas de otras personas que han compartido su ubicación, nos dará una cierta seguridad sobre el lugar.
- *Pertenencia:* El hecho de compartir ubicación de lugares o en fotos, hace que grupos de personas que han hecho lo propio sientan un nexo común.
- *Autoestima:* Poder llegar a prácticamente cualquier sitio a través de las tecnologías de geolocalización produce una contribución a mejorar la calidad de vida y autoestima de la sociedad.
- *Autorrealización:* Una vez cubiertas las necesidades previas, es posible contribuir a ayudar a los demás mediante recomendaciones de localizaciones, rutas, o incluso innovando y evolucionando estos sistemas para adaptarlos a las necesidades actuales e incipientes.

También hay que tener en cuenta que no todos son ventajas en cuanto al impacto social de estas nuevas tecnologías. Hay que tener en cuenta que el hecho de hacer uso de dispositivos de geolocalización también conlleva sus riesgos y han de ser tratados, o por lo menos tener conciencia de ellos. Temas como privacidad o seguridad podrán convertirse en problemas, pero estos serán estudiados más adelante.

En resumen, la generalización de los sistemas de geolocalización han supuesto grandes cambios en la sociedad, dando lugar a nuevas formas de desplazarse, comunicarse, compartir, etc. Y, aunque éstas estén en auge y aún quede mucho por desarrollarse, hoy en día se hace un uso masivo de ellas y se hacen imprescindibles para experiencias cotidianas.

2.2.3.3. *Impacto económico*

En el aspecto económico, las empresas han visto en el tema de la geolocalización una oportunidad de negocio, de manera que se puede llegar al cliente de una forma más directa, cercana y avanzada.

Los datos generados a partir de las aplicaciones de geolocalización son utilizados por empresas para mejorar las relaciones con los clientes [12], recogiendo información sobre sus hábitos y comportamientos, pudiendo así ofrecerles productos a su medida y recomendaciones que les satisfaga, dando lugar a mayores beneficios. Servicios como Maps for Business, Foursquare o Yelp están creciendo de forma vertiginosa ofreciendo este tipo de servicios.

La geolocalización ofrece múltiples oportunidades de negocio, incluyendo el marketing directo y la entrega de contenido en relación al contexto. Por ejemplo, las empresas pueden anunciarse mediante notificaciones cuando el usuario ronde las inmediaciones de una de sus tiendas, se pueden hacer visibles en los mapas de localización de los usuarios, desarrollan aplicaciones móviles que recomiendan, etc. También es importante el tema de que el usuario marque como favorito a una empresa al mismo tiempo que estampa su ubicación, ya que el boca a boca funciona bastante bien, y la empresa se difunde al círculo de amigos, creando la conciencia de marca.

Como se cita anteriormente, el principal líder en la difusión de las empresas en términos de geolocalización es actualmente Google Maps for Business [12], con muchas facilidades para que éstas puedan publicitarse, localizarse y referenciarse en la web a Google Maps.

También el sector público ha avanzado con la difusión de las tecnologías de emplazamiento. Hospitales, colegios, oficinas, también quedan reflejadas y se vuelven más accesibles al ciudadano.

2.2.4. Nuevas tendencias

2.2.4.1. *Introducción*

Una vez estudiados los diferentes sistemas de geolocalización predominantes, a continuación se van a exponer las diferentes tendencias de futuro, que ya están dando grandes coletazos en la actualidad. En primer lugar se profundizará sobre las nuevas redes sociales basadas en localización, que tendrán gran importancia ya que conjugará el auge de las redes sociales con la localización en sí. A continuación se explicará en qué consiste el geo-crowdsourcing, basado en la creación de mapas mediante la compartición de la localización. Y finalmente, se hablará sobre las redes de aceras basadas en marcas GPS, cuya base está en la navegación GPS a nivel del peatón.

2.2.4.2. *Redes sociales basadas en localización*

El despegue que están teniendo los smartphones actualmente junto con la capacidad de posicionamiento de los dispositivos, han conseguido incrementar el uso de redes sociales basadas en localización, en gran parte gracias a la inclusión de las redes sociales en sí en nuestras vidas. El uso masivo de éstas ha permitido llevar a cabo un estudio de los datos recogidos, para así poder entender e incluso resolver distintos problemas en terrenos como los negocios, publicidad y sanidad. Se ha estudiado que las conexiones entre los individuos en las redes se puede escindir en dos tipos: las basadas en la similitud con otro individuo/empresa, y las que buscan una oportunidad potencia de crear nuevos nexos. Por tanto, viendo estos comportamientos, es claro que las redes sociales basadas en localización están en proyección y tienen un futuro prometedor.

En este tipo de redes se distinguen principalmente dos tipos de información de localización. La primera de ellas es la fundamentada en el seguimiento de un miembro de la red, al tiempo que se va marcando en un mapa el camino recorrido, que resultan de gran utilidad para monitorización de niños o personas enfermas. La otra está más enfocada a los puntos por los que el dispositivo va efectuando "check in", en los que ha estado presente creando un punto de interés sobre el lugar visitado. Cada uno tiene sus ventajas e inconvenientes pero el uso de uno u otro va a venir definido por el contexto.

Las redes sociales de geolocalización se pueden orientar con fines prácticos muy diversos. Simplemente voy a apuntar como ejemplos los servicios de compartición de geolocalización, las guías de localización y las redes orientadas a los negocios,

a las que también se les puede ver como un juego, gracias a las innovaciones de ofertas de redes como Foursquare, Place Tips o Yelp!.

2.2.4.3. *Geo-Crowdsourcing*

Hasta hace muy poco, los mapas eran creados por agencias especializadas vinculadas a organizaciones militares o gubernamentales, pero hoy en día, la compartición de la posición geográfica ha dado lugar a un nuevo servicio de Geo-Crowdsourcing, el cual está basado en crear mapas online por grupos de personas, como es el caso de OpenStreetMap [6].

La capacidad de colaborar para la creación de datos está asociada al crecimiento de las tecnologías de geolocalización, que se puede considerar también una parte fundamental dentro del conglomerado del Geo-Crowdsourcing.

Dentro de esta tendencia se han impulsado dos tipos de desarrollo web. Uno de ellos son las redes sociales basadas en localización, de las que se ha hablado antes, que estarían más enfocadas a la compartición de la localización y a la socialización que ello implicaría. El otro desarrollo sería el tema de geowikis, que tendrían un carácter más didáctico y de cara al público, compartiendo rutas o localizaciones para el interés ajeno.

Claros ejemplos de esta nueva tendencia es la mencionada anteriormente OpenStreetMaps, que se trata de un mapa interactivo que puede ser modificado por los internautas. También cabe citar Wikimapia, que se trata de una web en la que cualquiera puede describir cualquier lugar del mapa del mundo, o el caso de la española Wikiloc, puntera en compartición de rutas [6].

2.2.4.4. *Redes de aceras por marcas GPS*

Debido al gran éxito acarreado en los últimos años por la navegación GPS, se ha empezado a desarrollar un interés por la navegación a nivel de peatón. Se pasa de analizar y mostrar las rutas de carreteras y calzadas a las rutas de aceras, pasos de peatones, etc. para así trasladar este sistema a nivel de usuario a pie de calle.

Estos sistemas tienen como principales benefactores las personas con limitaciones, como pueden ser los invidentes por ejemplo, a los que se les facilitarían las rutas con instrucciones a partir de la voz, y también los minusválidos, que podrán configurar sus rutas para poder acceder con silla de ruedas a todos los lugares.

2.2.5. Seguridad y privacidad

2.2.5.1. *Introducción*

En la geolocalización no hay que fijarse únicamente en las cuestiones técnicas. La seguridad con la que los datos se envían y reciben, junto con el tratamiento de éstos en temas de compartición y almacenamiento, son aspectos muy a tener en cuenta. Se explicarán estos contenidos así como las principales técnicas para su mejora.

2.2.5.2. *Seguridad en la información*

El compartir la ubicación puede implicar problemas, como podrían ser seguimientos no deseados. Por ello, hay que tener en cuenta que para que todo tipo de información sea segura, tiene que cumplir las cuatro reglas básicas: confidencialidad, integridad, disponibilidad y responsabilidad. Debido a ellos se utilizan las técnicas de autenticación, que se basa en confirmar la identidad para permitir a un sujeto acceder a información dentro de un sistema; control de acceso, para así poder restringir operaciones no autorizadas; y auditoría de sistemas, mediante la recogida de información sobre las actividades de los usuarios que posteriormente podrán ser analizadas para identificar violaciones en el acceso o establecer responsabilidades.

2.2.5.3. *Privacidad en la información*

Este quizás sea el tema más importante, ya que, aunque a rasgos generales parezca que el acceso a los servicios de geolocalización sean gratuitos y de fácil acceso sin compromiso alguno, hay que tener en cuenta que pueden existir brechas.

Debido al incremento de la información almacenada sobre los usuarios y la compartición de ésta, a veces sin su consentimiento, la privacidad se ha convertido en un asunto de importancia tanto a nivel legal como social. Se puede definir como el hecho de que un individuo pueda controlar su información personal y así mantenerse en el anonimato salvo que dé su consentimiento.

Las empresas que proporcionan estos servicios de localización, hacen uso de nuestros datos, normalmente a cambio de ofrecernos publicidad. Además el hecho de que conozcan nuestra posición, también en momentos que no queremos compartirla, crean una incomodidad y posibles problemas legales. Aun así esto también se puede ver como una oportunidad, porque al centrarse las empresas en mejorar la privacidad, conscientes de los problemas que conllevan fallos en este aspecto, lleva a dar la sensación de una mejora, creando el efecto de aumentar estos servicios.

2.2.5.4. *Técnicas para su mejora*

2.2.5.4.1. *Ofuscación de localización*

Su principio de funcionamiento se basa en degradar la calidad de la información de localización sin alterar la identidad del usuario [6]. La ofuscación espacial se puede conseguir mediante tres tipos de incertidumbres: inexactitud, que se refiere a una falta de veracidad en la información; imprecisión, referido a una falta de especificidad en la información; y distracción, como falta de determinación de la información.

El principal método se basa en la ofuscación en forma circular, permitiendo al usuario definir la privacidad por un radio mínimo. Tras ello los operadores pueden alargar o reducir el radio, desplazar el centro o incluso combinar estas técnicas para proteger al usuario con un nivel mayor.

También existen otras formas de ofuscación basadas en algoritmos que dividen las regiones en subregiones, para así proporcionar ofuscación en base a distribuciones estadísticas de usuarios en las distintas localizaciones.

2.2.5.4.2. *Pseudonimización*

La pseudonimicidad proporciona una forma de anonimidad de manera que la asociación usuario-localización sólo puede establecerse bajo ciertas circunstancias [6]. Funciona haciendo uso de una capa intermedia de software entre el sistema de localización y las aplicaciones que sean poco confiables. Así éstas interactuarán con la susodicha capa, que actuará de filtro dependiendo de los permisos que el usuario quiera dar dependiendo de si confía en las aplicaciones o no, o si está en una zona en la que no quiere ser localizado.

2.2.5.4.3. *Anonimización*

Abordar el problema de anonimizar los datos antes de mandarlos se ha convertido en un verdadero problema y se han hecho varios intentos para intentar solucionarlo [6]. Por ejemplo, el principio de k-anonimidad, requiere que cada envío de datos sea indistinguible por, al menos, (k-1) otros envíos, para así protegerlos de ataques por comparación. También existen técnicas de anonimización espacial, donde el usuario se hace invisible en zonas donde el usuario lo requiere.

2.2.5.4.4. *Control de acceso*

En multitud de servicios, como los basados en localización, toma un valor crítico el determinar quién puede tener acceso a la información de localización y en qué condiciones. Los principales métodos, a rasgos generales, son [6]:

- *GEO-RBAC*. Funciona definiendo roles espaciales, en función de fronteras no políticas (accidentes geográficos, carreteras, etc.), que definen un territorio lógico en el que el rol es válido o no, dependiendo de los permisos dados por el usuario.
- *Location and Time-Based RBAC*. Permite crear jerarquías de localizaciones (tanto físicas como lógicas), dotando así a los nodos de permisos, y de poder también permitir a nodos por debajo del árbol que controlen a sus subnodos.

2.2.5.4.5. *Autenticación*

La información sobre la autenticidad de la información proporcionada por los usuarios es un requerimiento básico en las preocupaciones por la privacidad y seguridad. Se pueden producir casos de ataques de suplantación de localización, para así adquirir servicios del proveedor de forma fraudulenta o realizar actividades maliciosas.

Para mejorar la autenticación de los usuarios, se han utilizado diversos métodos como la comprobación geográfica por GPS [6] , el empleo de un certificado de localización mediante el uso de técnicas criptográficas, o utilizando una plataforma de confianza que a través de una clave proporciona la localización a aplicaciones de confianza.

2.2.5.5. *Localización en dispositivos móviles*

Una vez presentadas las distintas tecnologías de geolocalización que gozan de un mayor interés actualmente, es necesario recalcar que los dispositivos móviles que siempre llevamos encima disponen de algunas de ellas, para que así el usuario o los servicios de localización las aprovechen a su gusto y medida.

Como puede intuirse, el principal sistema de localización es el que hace uso de las redes móviles, que a través de las celdas va situando y registrando dónde se encuentra el dispositivo en cada momento.

Otro sistema utilizado es el GPS, que activándolo puede realizar una triangulación de la posición, si se encuentra visible al menos para cuatro satélites de la constelación.

Por último, la triangulación por WIFI es menos común pero también posible, dando lugar a las coordenadas del usuario en función de las señales recibidas en una o varias antenas receptoras.

2.3. Sistemas operativos en dispositivos móviles

2.3.1. Introducción

El otro tema importante a tratar en este documento se centra en la importancia del sistema operativo como lugar de conjunción de todas estas tecnologías y de la definición de su comportamiento. Se procederá a describir los tres tipos de sistemas operativos en dispositivos móviles que predominan en el mercado, tanto sus aspectos más técnicos como su aceptación en el mercado, justificando finalmente el porqué del uso de Android como centro del desarrollo de la aplicación.

2.3.2. Tipos

2.3.2.1. *Android*

2.3.2.1.1. *Orígenes*

Android Inc. fue en sus orígenes una compañía dedicada a producir aplicaciones para terminales [13] [14]. Ésta fue adquirida por Google en 2005, empezando a trabajar en la creación de una máquina virtual de Java optimizada para móviles (Dalvik).

En 2007, Google con la ayuda de otras compañías, entre ellas Intel, Texas, Motorola, etc. crea el consorcio Handset Alliance, para así desarrollar estándares abiertos para dispositivos móviles. Este manto era perfecto para la distribución de la tecnología Android y de código abierto Apache.

A continuación el desarrollo del Android SDK y el lanzamiento de dispositivos con este sistema operativo, ha propiciado un rápido despliegue que le llevó en el año 2011 a colocarse como la plataforma más importante en el mercado. Actualmente la cuota de mercado sigue aumentando y siguen desarrollándose nuevas versiones de sistemas operativos, al mismo tiempo que tanto desarrollo como consumo de aplicaciones, cada vez están al alcance de mayor número de personas.

2.3.2.1.2. Arquitectura

Como se puede ver en la siguiente figura, la arquitectura de Android [13] [15] [14] se podría simplificar a cuatro capas, las cuales están basadas en software libre todas ellas.

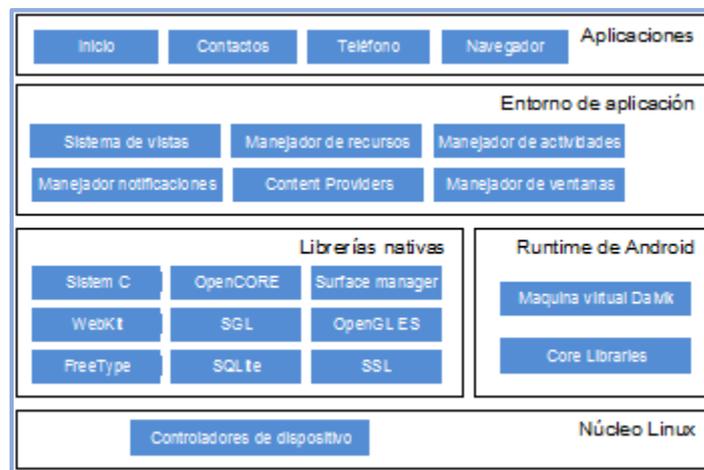


FIGURA 6: ARQUITECTURA DE ANDROID [13]

A nivel más bajo se puede encontrar el núcleo de Linux, que maneja los controles del dispositivo y proporciona los servicios de manejo de memoria, multiproceso, etc. Es una abstracción entre hardware y el software de la pila.

A continuación se encuentran a igual nivel el Runtime de Android y las librerías nativas. El primero está consonancia a lo explicado en los orígenes del sistema operativo. Como Android es un sistema muy potente, y los dispositivos móviles son más limitados en hardware que un ordenador convencional, se optó por sustituir la tradicional máquina virtual de Java por la Dalvik, diseñada específicamente para este propósito. Para optimizar los recursos, su funcionamiento se basa en el uso de archivos .dex y registros, corriendo cada aplicación en su propio proceso Linux, con su correspondiente instancia proporcionada por la máquina virtual Dalvik. También se incluyen las librerías principales de Java en Core Libraries.

En cuanto a las librerías nativas, se tratan de las principales usadas en los componentes de Android. Estas son:

- System C library. Adaptación de la librería estándar de C para dispositivos móviles.
- Media Framework. Aglutina codecs de reproducción y grabación de audio, vídeo e imágenes.
- Surface Manager. Encargada de la representación gráfica en 2D y 3D.
- Webkit. Soporta el navegador web usado en Android.
- SGL. Motor de gráficos 2D.
- Librerías 3D. Usan el acelerador hardware 3D o software 3D (si se encuentran disponibles).
- FreeType. Fuentes en bitmap y uso de vectores.
- SQLite. Motor de base de datos.
- SLL. Proporciona servicios de encriptación.

La siguiente capa sería el Entorno de aplicación, que da al desarrollador una capacidad muy amplia de mejora mediante el uso de componentes y otras aplicaciones. Usando codificación Java se pueden manejar distintos servicios entre los que destacan:

- Views. Parte visual de los componentes.
- Resource Manager. Proveedor de recursos que no son en código.
- Activity Manager. Es el encargado de manejar el flujo de vida de las aplicaciones y permite la navegación entre ellas.
- Notification Manager. Da lugar a que las aplicaciones muestren alertas personalizadas en la barra de estado.
- Content Providers. Mecanismo para el acceso a los datos de otras aplicaciones.

Por último la capa de aplicaciones, aglutina todas las instaladas en el dispositivo, que deberán correr en la máquina Dalvik, para garantizar la seguridad y estabilidad del sistema. Principalmente, las aplicaciones se desarrollan en Java con el Android SDK, pero también existe la opción de programar en C o en C++, mediante el Android NDK [14].

2.3.2.1.3. Versiones

Desde su aparición, Android ha ido desarrollando distintas versiones de sistema operativo [13] [15] [14], y cada nueva plataforma ha sido siempre con las versiones anteriores, añadiendo nuevas funcionalidades o etiquetando como obsoletas las que no se sigan usando. Éstas tienen tres formas alternativas de identificarse: Versión, nivel de API y nombre comercial.

Las principales versiones, en orden cronológico de salida al mercado son:

- 1.1 y 1.2 (API 1 y 2): de escaso desarrollo y comercialización.
- 1.5 (API 3), Cupcake: primera versión con algún usuario. Añade predicción de texto, widgets y soporte para bluetooth estéreo.
- 1.6 (API 4), Donut: aparece la búsqueda avanzada, y se incorporan la síntesis de texto a voz, distintas densidades de pantalla,...

- 2.0 y 2.1 (API 5 y 7) Éclair: prácticamente se pasó a comercializar la versión 2.1. Su mayor incursión fue la clase motionEvent, que soporta pantallas multitáctil.
- 2.2 (API 8) Froyo: Como hecho destacado se introduce un nuevo compilador JIT en la máquina Dalvik, que hace que el sistema operativo mejore sus prestaciones en velocidad.
- 2.3 (API 9) Gingerbread: Es la primera versión pensada para tabletas.
- 3.0 (API 11) Honeycomb: A partir de los cambios de la versión anterior se mejora la interfaz de usuario y aparecen las librerías de compatibilidad, a través de las cuales se puede usar aplicaciones construidas para versiones anteriores.
- 3.1 y 3.2 (API 12 y 13): Se permite la interacción con USB, MTP y RTP.
- 4.0 (API 14) Ice Cream Sandwich: Se unifican las versiones 2 (para teléfonos) y 3 (para tabletas), renovando el interfaz de usuario. Aparece el conocido medidor de consumo de datos.
- 4.0.3 (API 15): Se incluyen aplicaciones de redes sociales, calendario...
- 4.1 (API 16) Jelly Bean: Esta versión busca aumentar la fluidez del interfaz con el usuario, mediante el sincronismo vertical, el triple buffer y aumentando la velocidad del procesador al tocar la pantalla.
- 4.2 (API 17): Aparece el desplazamiento por el teclado Swype y se incorpora en las tabletas la opción de crear varias cuentas por dispositivo.
- 4.3 (API 18): En esta versión se añade por primera vez las restricciones en las apps al mismo tiempo que se mejora la seguridad para la gestión de las claves privadas.
- 4.4 (API 19), KitKat: El nombre de esta versión viene de un acuerdo con Nestlé. Con esta versión se ha buscado ampliar Android a una gama mayor de dispositivos, incluso aquellos con poca capacidad de memoria. Desde el punto de vista del interfaz de usuario, se agrega la capacidad de pantalla completa. Además el fenómeno de la nube empieza a tomar relevancia incorporando al content provider el document provider, que mejora la interacción con estos ficheros. Se introduce además una nueva ART, que mejora las prestaciones de la Dalvik.
- 5.0 (Api 21), Lollipop: Principalmente se añade colorido al diseño, con un realismo mayor en cuanto al tema de interacción con el usuario. Se añade un nuevo control de notificaciones e interrupciones, que da más posibilidades de control y respuesta al usuario, junto con la extensión de Android a nuevas plataformas como Google Wear, Google Tv y Google Card. También el tiempo de batería se mejora en unos 90 minutos como máximo y aparecen otras mejoras menores, en mayor parte de rendimiento.

2.3.2.2. IOS

El sistema operativo IOS [16], es el desarrollo realizado por Apple para dar soporte a sus productos. Inicialmente creado para el iPhone, se ha ido extendido al resto de dispositivos de la marca progresivamente

2.3.2.2.1. Orígenes

En el año 2007, Apple sacó a la luz el sistema operativo iPhone OS, para incorporarlo en el mismo año al terminal iPhone [15]. A partir de aquí el SDK se iría extendiendo desde dispositivos como el iPod al iPad, dando lugar finalmente en 2010 al cambio de nombre de iPhone OS a iOS, para unificar a todos los productos de la marca. Desde entonces se han ido evolucionando y creando nuevas versiones, normalmente coincidiendo con la presentación de productos estrella de la marca.

2.3.2.2.2. Arquitectura

Su interfaz [15] está basada en el hecho de dar instrucciones a través de gestos multitáctiles, lo cual supone un hecho revolucionario en el sector, creando una fluidez entre terminal y usuario. Además, la nueva pantalla principal, llamada también SpringBoard, junto con las novedades introducidas como las aplicaciones, hacen que esta experiencia sea aún más satisfactoria. En Apple también se le ha dado suma importancia a las aplicaciones multitarea y su consumo de batería, por lo que se han ido introduciendo mejoras en el SDK. Los mayores inconvenientes que tiene esta tecnología es el gran número de tecnologías no admitidas, en parte por política corporativa, para así diferenciarse de la competencia, y por otro lado por otras razones como puede ser la seguridad (Flash Player y Java).

En cuanto la arquitectura en sí, la pila de protocolos sería la mostrada en la figura siguiente:

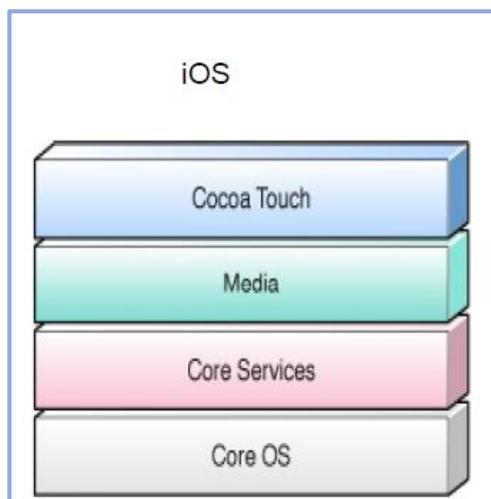


FIGURA 7: ARQUITECTURA DE IOS [16]

Como se puede observar, las capas serían las siguientes:

- Núcleo de OS: Contiene los archivos del sistema y características de bajo nivel (manejo del procesador, memoria, seguridad, hardware,...). Está basada en sistema operativo Unix.
- Núcleo de Servicios: Se encarga de proveer los servicios básicos del sistema operativo para el uso de las aplicaciones.
- Capa de Medios: Es la responsable de los servicios gráficos y multimedia.
- Cocoa Touch: Es la contenedora de todo lo necesario para desarrollar aplicaciones. Se diferencia en dos subcapas:
 - UIKit: con todas las clases útiles para el desarrollo del interfaz del usuario.
 - Foundation Framework: define las clases para manejar objetos del sistema operativo.

2.3.2.2.3. Versiones

Estas son las principales versiones del sistema operativo surgidas a lo largo del tiempo [15] [17] [16]:

- iOS 1.0: Esta versión ya incorporaba las principales aplicaciones de las que disfrutamos hoy en día, como son Google Maps, Safari e iTunes.
- iOS 2.0: El mayor impacto de esta versión fue la introducción del App Store, que hacía mucho más fácil encontrar e instalar aplicaciones. También se introduce el SDK que permite el desarrollo de aplicaciones a través de un kit especial. Además, Google Maps se complementa con Google Street.
- iOS 3: En la versión 3, principalmente hay que recalcar la introducción de las notificaciones push y la capacidad de copiar y pegar.
- iOS 4: Esta versión destaca por realizar grandes lanzamientos. Los principales son las carpetas y multitareas, acompañados también por mejoras para el iPad.
- iOS 5: Aquí aparece el hecho de la conectividad Wireless con el ordenador, el reconocimiento por voz gracias a Siri y el concepto de nube con el iCloud.
- iOS 6: Esta versión es más conocida por sus equivocaciones que por sus mejoras. El nuevo Maps de Apple y el rediseño de la App Store no tuvieron una buena acogida por parte de los usuarios. Aun así el mayor logro fue la introducción de la aplicación de Facebook en iOS.
- iOS 7: Con la versión 7 se vuelve a dar un vuelco en cuanto a la interfaz y a la introducción de nuevas mejoras en cuanto a multitarea y ventanas, y de nuevas características como el iTunes Radio o los filtros de fotos.
- iOS 8: Esta versión llega con la app Health como bandera, que aprovecha el boom de los wearables y la condición física para poder juntarlo todo en el terminal móvil. Pero no se queda aquí la cosa, ya que también se incorporan las notificaciones inteligentes (que se

pueden contestar desde el mismo panel de notificaciones), el teclado Quictype y una remodelación de la aplicación de fotos, que se integra completamente en el iCloud.

- iOS 9: El sistema operativo más reciente incorporado a Apple, tiene muchos cambios agregados. Principalmente se mejora la funcionalidad de Siri, se mejoran las interacciones entre aplicaciones, aparece la multiventana, mejoras en mapas, conexión inalámbrica con el sistema funcional del automóvil y mejoras en autonomía de batería y programación del sistema.

2.3.2.3. Windows Phone

2.3.2.3.1. Orígenes

Windows Phone [15] [18] es el sistema operativo desarrollado para dispositivos móviles por Microsoft, que fue lanzado en el 2010 para suplantar al antiguo sistema operativo Windows Mobile, dado que no había tenido mucho éxito por lo que se decidió empezar a construir otro desde cero.

2.3.2.3.2. Arquitectura

A continuación se muestra un diagrama con la arquitectura [15] [18], organizada por capas y propiedad de cada bloque.

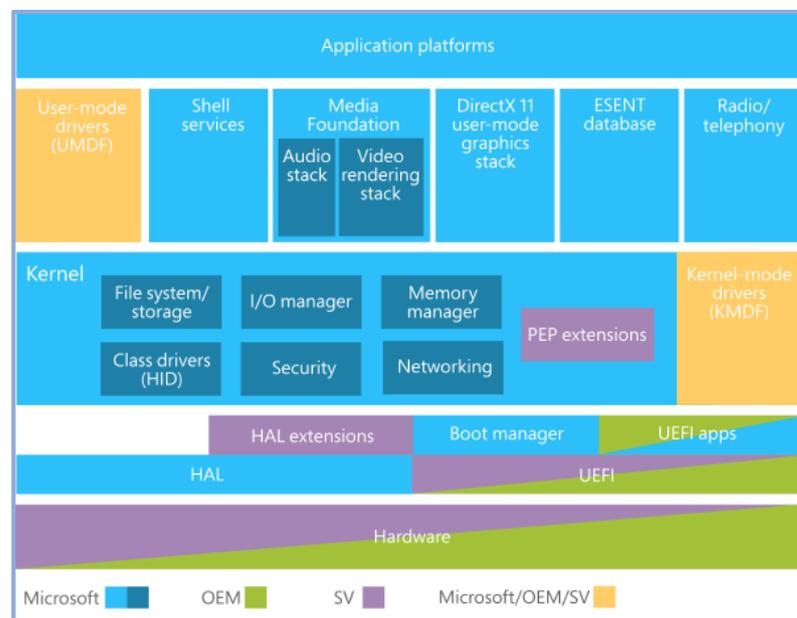


FIGURA 8: ARQUITECTURA DE WINDOWS PHONE [18]

Como se puede observar, en la capa más baja está el hardware y por encima de él los drivers y extensiones para manejarlos. El kernel aglutina esta gran parte de los drivers para que las aplicaciones hagan uso de las características principales del hardware. Un escalón por encima, se situarían las aplicaciones básicas del sistema operativo, como son la radio, los contactos, etc. y en la parte superior se sitúan el resto de aplicaciones.

Es de notar que Microsoft produce la mayoría de bloques de cada capa, aunque en algunas tiene que delegar en el fabricante de equipos o empresas colaboradoras.

2.3.2.3.3. *Versiones*

Tras empezar el sistema operativo de nuevo, Microsoft decidió seguir la numeración dejada atrás por Windows Mobile [15] [18]:

- Windows Phone 7: Versión inicial que se encontraba muy lejos en prestaciones con sus competidores y fue añadiendo funcionalidades en sus distintas mejoras como fueron mejores en rendimiento y sincronización, soporte para copiar y pegar, multitarea, integración con Twitter, LinkedIn, Office, etc. y renovación de la interfaz del usuario.
- Windows Phone 8: Versión presentada en 2012, introduce un cambio de kernel que no es compatible con todos los dispositivos. Con ella se añaden capacidades nuevas como nueva interfaz, medidor de consumo de datos, wallet, Skype y actualizaciones automáticas en el terminal sin conectar al PC. Igualmente se han ido sucediendo actualizaciones de la versión con mejoras diversas.
- Windows Phone 8.1: Desplegada en 2014, es la versión más actualizada hasta que ha entrado en funcionamiento el nuevo Windows 10. Sus principales características son el nuevo centro de notificaciones, Cortana como asistente de voz (en la línea del Siri de Apple) y aplicaciones de salud, viajes y mapas.
- Windows 10 Mobile: Diseñado principalmente para smartphones y tabletas de menos de 8 pulgadas, está todavía en fase de implantación. El primer elemento diferenciador será la Live Tiles, que mostrará información relevante de las aplicaciones en la pantalla de inicio, que será de gran importancia. Junto a ella, mejoras en optimización, estabilidad y seguridad se vuelven claves, siendo ésta última la gran baza de la compañía. Además es importante recalcar que con este sistema operativo se podrán importar aplicaciones de otros y hacerlas funcionar, lo que es una auténtica novedad en el campo. La integración con otros servicios como consolas, programas de Microsoft y la capacidad de conectar el dispositivo con entradas externas también cobran especial relevancia.

2.3.2.4. Otros

Existe una gran variedad de sistemas operativos en el mercado, pero los tres citados anteriormente copan la mayor parte de la cuota. Cabe reseñar los siguientes [19]:

- *Blackberry OS*: Desarrollado por la empresa RIM, tuvo un gran éxito asociado al dominio de Blackberry de hace unos años, siendo utilizado por usuarios profesionales en un principio hasta que se expandió al resto. Actualmente su tendencia en número de usuarios es a la baja.
- *Firefox OS*: Con un lanzamiento reciente al mercado (año 2013), Mozilla ha intentado seguir extendiendo su dominio con este sistema abierto de multiplataforma. Inicialmente, tiene una orientación hacia usuarios de bajos recursos para que facilitarles el acceso a las nuevas tecnologías.
- *Symbian*: Es el más antiguo de los sistemas operativos expuestos, lo que ha dado lugar a características muy buenas en cuanto a rendimiento y seguridad. No se supo adaptar adecuadamente a la aparición del Smartphone por lo que ha quedado en una posición estancada, pero aun así sigue manteniendo un número de usuarios fijos.
- *Ubuntu Phone*: Otro sistema operativo incipiente, también nacido en 2013, tiene una connotación sobre la utilización de código libre y abierto, que sigue a sus predecesores en ordenadores. Con un sector muy potente de seguidores, puede empezar a plantar cara a los grandes sistemas operativos.

2.3.3. Estudio de mercado, comparativa

Una vez presentados los sistemas operativos, es importante ver la tendencia de mercado que han tenido a lo largo de los últimos años [13] [20] [21] [22], que se puede observar en la siguiente gráfica (Figura 9), en base al número de terminales vendidos.

A través del estudio de Gartner Group podemos observar que la plataforma predominante hasta el 2010 (Symbian), empieza a caer en picado coincidiendo con la subida de Android. Aquí tiene mucho que ver la generalización del Smartphone y la capacidad de Google de conseguir distribuir el sistema operativo entre la mayoría de ellos. Al declive de Symbian le acompaña BlackBerry, que también empieza a perder clientes en contra de Android.

Por su parte, IOS mantiene a sus usuarios fieles, con una cuota más o menos continua, lo que es un hecho que ha caracterizado a la marca Apple en los últimos años. Aun así es incapaz de despuntar para darle alcance a Android.

Windows Mobile, viendo que no conseguía adeptos intentó reinventarse pasándose a llamar Windows Phone, pero como se puede observar, los resultados no se acercan si quiera a los de IOS y mucho menos a los de Android.

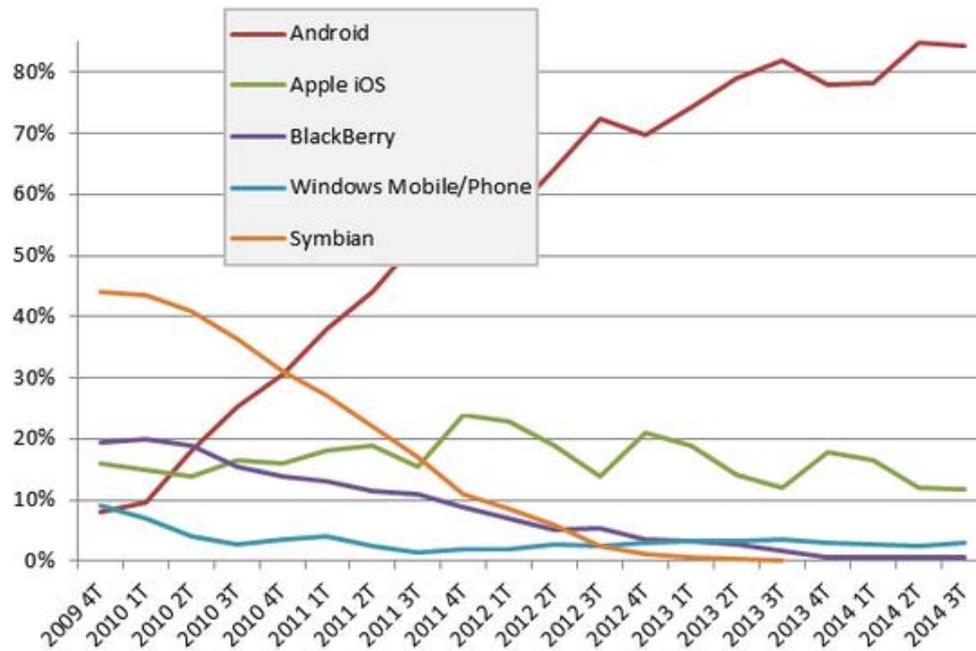


FIGURA 9: EVOLUCIÓN DE LA CUOTA DE MERCADO DE LOS SISTEMAS OPERATIVOS MÁS SIGNIFICATIVOS [13]

2.3.4. Android, el sistema operativo dominante

Tras ver cómo los sistemas operativos han ido evolucionando en el mercado, se puede observar que la tendencia de los últimos años está claramente dominada por Android. En este sentido es la primera opción en la que elegir desarrollar una aplicación que tenga la mayor cabida posible en el mercado. También es cierto que los usuarios de iOS estén más predispuestos a pagar por la descarga y uso de aplicaciones, pero aun así la cuota de usuarios sigue siendo mucho menor, y los reportes que pueda traer el desarrollo Android en temas de publicidad pueden contrarrestar este aspecto.

Además en cuanto a sus características [13] [23] [24] [25] [21], tiene una serie de ventajas sustanciales con respecto al resto de sistemas operativos:

- Se trata de una *plataforma abierta*, basada en Linux (de código abierto), que puede ser usada y mejorada sin necesidad de abonar royalties.
- Se puede *adaptar a cualquier hardware*. La explosión de los wearables ha dejado constancia de este hecho en el que no sólo opera en terminales móviles y tabletas. En iOS por ejemplo, habría que realizar la misma aplicación para iPhone y para iPad.
- El desarrollo en Java, da lugar que las *aplicaciones sean portables* a dispositivos de CPUs futuras, gracias al concepto de máquina virtual.
- La *arquitectura tiene una inspiración en Internet*, lo que permite que las aplicaciones se puedan exportar a distintos tipos de pantalla.
- La filosofía es que el *dispositivo se encuentre permanentemente conectado a Internet*.

- Tiene una *cantidad muy grande de servicios* incorporados, como son el GPS, navegador, bases de datos,...
- El *nivel de seguridad es muy bueno*, debido por un lado al aislamiento de los programas que obtiene del trabajo con Linux, y por otro a los permisos que dispone cada aplicación para limitar el rango de actuación.
- Se busca que el sistema esté *optimizado para baja potencia y memoria*, y de aquí nace la máquina virtual Dalvik, que es similar a la JVM pero optimizada para dispositivos móviles.
- Dispone de una *calidad alta de gráficos y sonidos*, mediante el uso de Flash, códecs, gráficos vectoriales, etc.

En resumen, la combinación de las características de Android, junto con la gran aceptación y evolución de la plataforma entre los usuarios, sumado además a la facilidad de implementación de las aplicaciones, hace de éste el sistema operativo ideal para poder desarrollar cualquier herramienta para una gran variedad de dispositivos.

3. Diseño de la aplicación

3.1. Introducción y objetivo del sistema Android

Una vez hecha una presentación sobre lo diferentes conceptos principales, se va a proceder al desarrollo del proyecto en sí. El principal objetivo es hacer capaz a un usuario, a través de su dispositivo móvil, de conocer la posición de los compañeros seleccionados mediante la lista de contactos del terminal y representarlos en un mapa.

Para ello se propone un escenario en el que participan varios agentes, que interaccionarán entre sí para su correcto funcionamiento. Estos son:

- Cliente: Se trata del propio terminal, que se conectará con el servidor, enviándole periódicamente su posición. Su identificador único será el número de teléfono propio.
- Servidor: Será básicamente una base de datos, que almacenará los contactos agregados (con sus correspondientes localizaciones y última fecha de refresco) junto con la posición propia y última actualización.
- Contactos: mandan periódicamente su posición al servidor y al mismo tiempo actúan como clientes para sus usuarios. Al igual que el cliente, se identificará con el número de teléfono asociado.

El trabajo se centrará en la parte de la aplicación del cliente, que entra en funcionamiento con un servidor de un proyecto previo. El escenario se refleja en la Figura 10, y a partir de éste se llevará a cabo una explicación de los diferentes casos de uso.

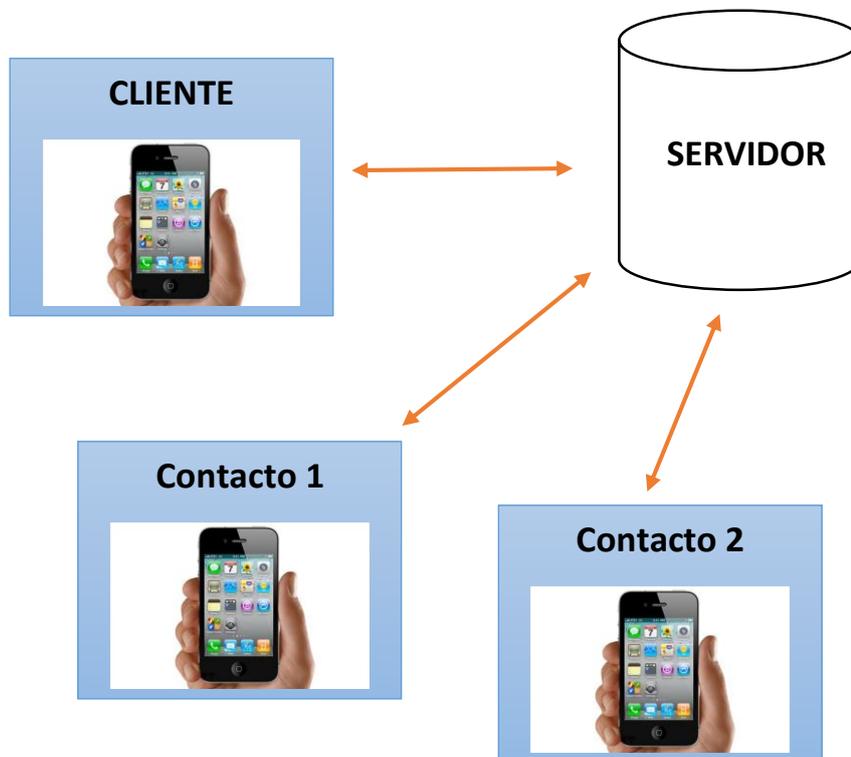


FIGURA 10: ESTRUCTURA DEL ESCENARIO DEL PROYECTO

3.2. Casos de uso

3.2.1. Introducción

Una vez explicados los objetivos del sistema, a continuación en este apartado, se procederá a desarrollar las interacciones que el cliente tendrá con el servidor (casos de uso), para el funcionamiento de la aplicación.

3.2.2. Enviar localización de posición

Primeramente, para un óptimo funcionamiento, el cliente debe enviar su posición de forma periódica al servidor, para que los contactos puedan conocer su localización. Éste puede graduar la frecuencia de emisión de la posición, para así poder alcanzar un compromiso entre la visibilidad propia ante el resto de usuarios y la duración de la batería del propio dispositivo. La Figura 11 hace visible este escenario, en el que se observa que el cliente envía al servidor, a través de una función setLocation, su número y posición. Así el resto de usuarios serán capaces de conocer la situación del cliente actualizada. Éstos al mismo tiempo también envían su posición de forma periódica, ya que funcionan como clientes de cara a sí mismos y como compañeros desde el punto de vista actual.

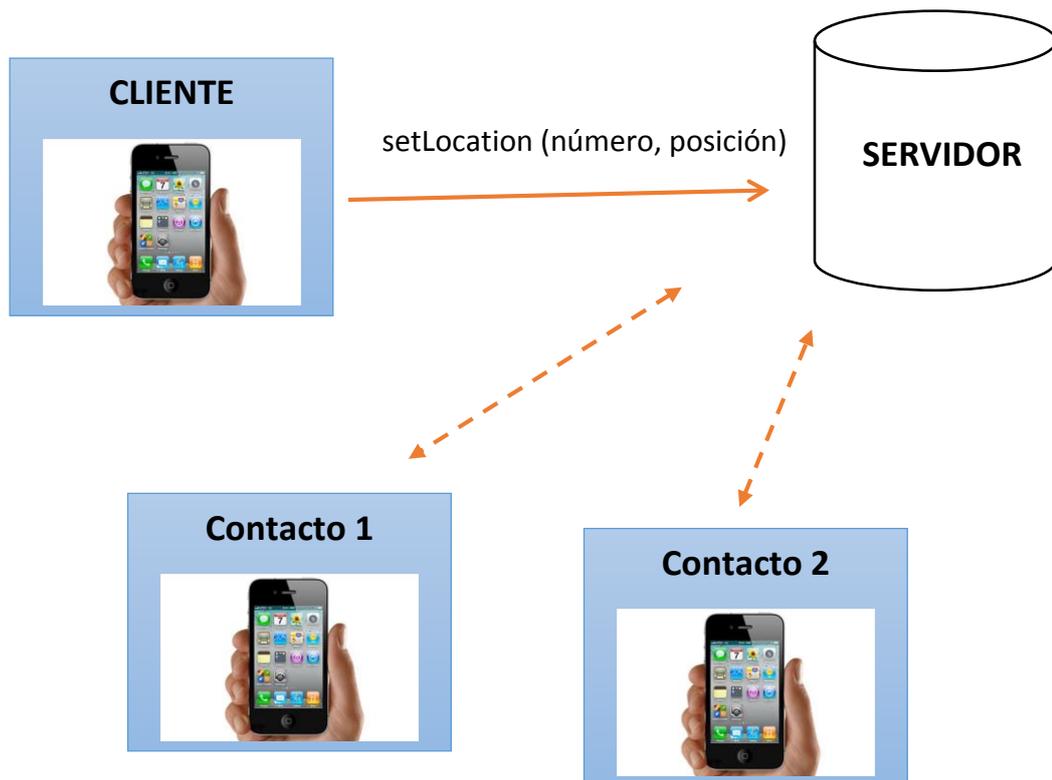


FIGURA 11: ENVÍO DE LOCALIZACIÓN DE POSICIÓN

3.2.3. Seleccionar compañeros

El segundo escenario que se presenta en los casos de uso, se basa en la selección de los compañeros que queremos añadir a nuestra aplicación y por tanto, visualizar en nuestro mapa. Esto se realizará mandando al servidor un comando de `setColleagues`, que incluirá el número propio y una lista de los números de los compañeros seleccionados, como se ilustra en la Figura 12 a continuación.

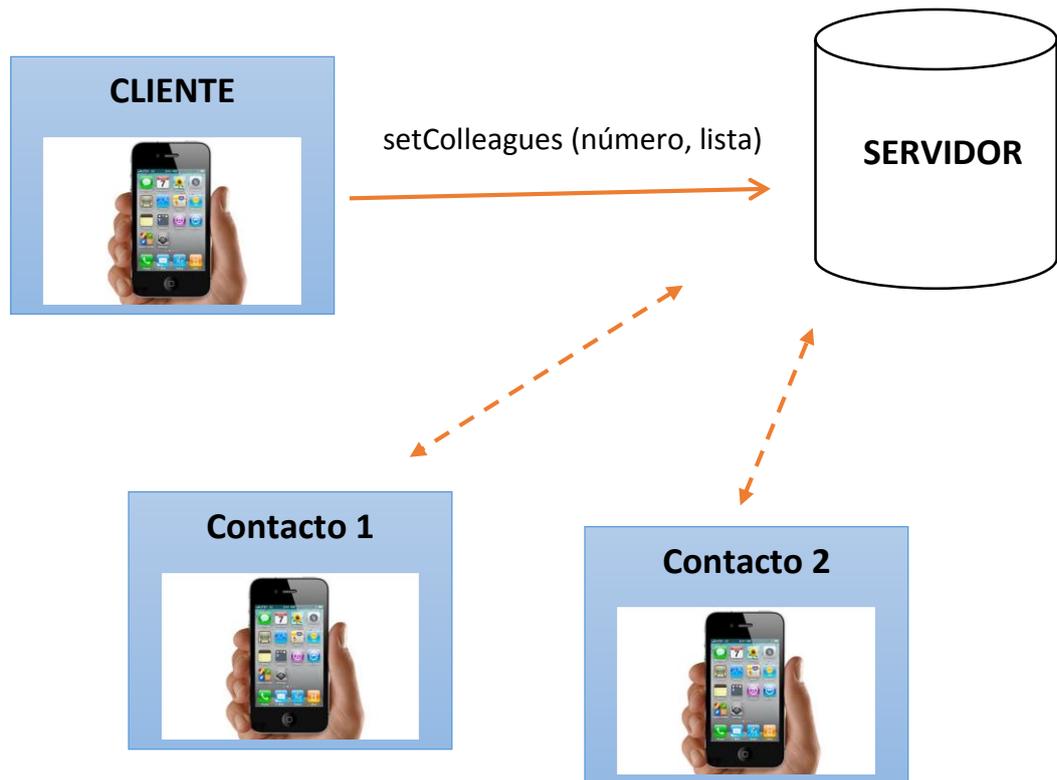


FIGURA 12: SELECCIÓN DE LOS COMPAÑEROS

3.2.4. Recibir localizaciones

Por último, ya sólo quedaría el recoger las posiciones de los compañeros seleccionados, para así representarlos en el mapa y conocer su situación actual. Esto se consigue a partir del envío al servidor del comando `getLocations`, que a través del número propio, devuelve las posiciones de los contactos seleccionados. Este caso de uso se expone a continuación en la Figura 13.

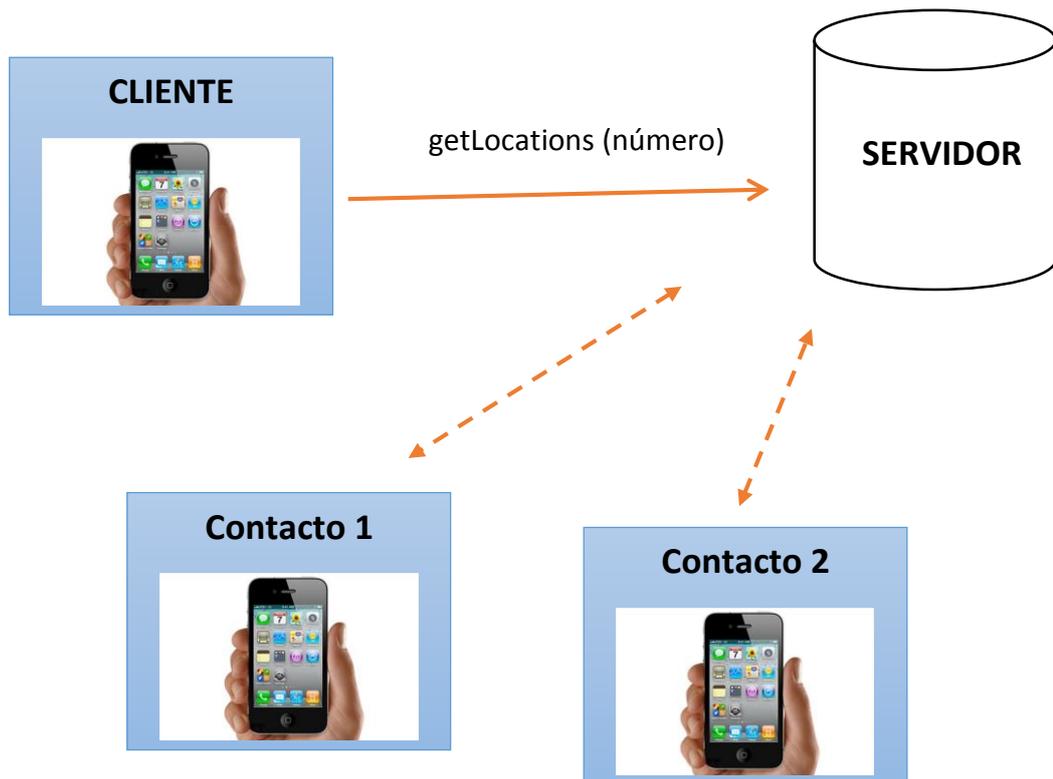


FIGURA 13: RECEPCIÓN DE LOCALIZACIONES

Al igual que en el primer caso, existe una frecuencia de actualización, que permite cada cierto tiempo pedir la posición de los compañeros. Ésta podrá ser configurada aumentando la frecuencia de petición, conllevando un uso superior de batería pero mejor precisión de visibilidad de resto de usuarios, si estos están en movimiento; o disminuyéndola, lo que traería consigo un mayor ahorro energético, pero menos exactitud en los cambios de posición.

3.3. Privacidad

El mayor inconveniente que se puede plantear, una vez que se han desarrollado los diferentes casos de uso, puede ser la gestión de la privacidad. Ante la duda de si un usuario puede conocer la posición de otro sin el consentimiento de éste, la respuesta es que no. Para poder visualizar la posición de un usuario, éste ha tenido que añadir al primero previamente en la lista de contactos. En caso contrario, no estará autorizando a que pueda conocer su posición.

4. Implementación

4.1. Introducción

Ahora nos centraremos en la definición de la aplicación a nivel más técnico, profundizando más en la implementación en sí y en su funcionamiento interno.

Para que la aplicación cumpla los requisitos de diseño, tendrá que tener unas características que hagan posible su funcionamiento. Por ello se han definido una serie de clases y pantallas, que interactuarán entre ellas. Primeramente se explicará el funcionamiento en sí de la aplicación junto con una pequeña introducción a las distintas pantallas y clases. Posteriormente se explicarán en detalle las pantallas que las componen y la programación de las clases implementada para cumplir la funcionalidad descrita.

4.2. Funcionalidad de la aplicación

4.2.1. Introducción

La aplicación funcionará de manera que al iniciarla se mostrará la pantalla principal de ésta, donde aparecerá el mapa con la posición propia centrada, y con un enfoque a nivel de barrio, para así situar mejor al usuario. En el caso de que sea la primera vez que se inicia ésta, habrá que introducir previamente por pantalla el número de teléfono propio, como identificador único de usuario. Ésta será la principal interfaz con el usuario a través de la cual el cliente puede interactuar con las distintas funcionalidades.

Una vez aquí el usuario deberá introducir, mediante el menú, los contactos que quiere añadir para poder visualizar su posición, que se corresponde al comando explicado anteriormente “setColleagues”. A partir de aquí, cada cierto tiempo el servidor enviará al usuario las posiciones de éstos para notificar posibles movimientos, correspondiéndose al caso de uso de recibir localizaciones, desarrollado previamente con el envío del “getLocations”. A la vez, el dispositivo enviará su posición periódicamente al servidor, correspondiéndose a la función “setLocation” de los casos de uso.

Al haber seleccionado los contactos a visualizar, éstos aparecerán en el mapa de la pantalla principal. Desde ella se podrá acceder al menú de Settings (para los distintos ajustes), al About (que muestra la información de copyright), y se podrán pulsar distintos botones cuya funcionalidad se explicará más adelante.

Cabe recalcar que en los ajustes (Settings) se podrá configurar tanto el tiempo de refresco de la posición de los contactos, así como el envío de la posición propia, para poder controlar mejor el compromiso entre la duración de la batería y el propio refresco, que varíe la funcionalidad ligeramente. Además también se incluye en este menú la posible modificación del número de teléfono y el añadir contactos, para poderlo hacer por primera vez, como se mencionó antes, o para cambiar éstos.

Además es necesario reseñar que, el usuario podrá ver la última hora de conexión de un contacto al pulsar sobre su marcador, y al mismo tiempo cambiará su color dependiendo de ella.

Para ampliar los aspectos de la aplicación, a continuación se introducirán las distintas pantallas de las que constará, seguidamente se procederá a introducir también las clases desarrolladas para que se cumpla el funcionamiento y finalmente se explicará el protocolo de funcionamiento con el servidor.

4.2.2. Diseño visual

En la funcionalidad se ha explicado a grandes rasgos cómo se define la aplicación. Ahora se introducirán a grandes rasgos las pantallas de las que constará la aplicación, para así tener una visión general de cara a la explicación detallada en apartados posteriores:

- Principal: Constará del mapa que representa la posición propia junto con los marcadores de los contactos que además da acceso al menú de ajustes Settings y la pantalla de About. También incluye dos botones permanentes para refrescar la posición de los contactos y ajustar posiciones propia y de los compañeros al mapa, y otros dos que aparecen al pulsar sobre el marcador, para visualizarlo con la aplicación de Google Maps y crear una ruta.
- Settings: Dará acceso al menú de la aplicación. Aquí existen las opciones de modificar el número de teléfono, las frecuencias de refresco de envío de posición y de recepción de la de los contactos, y añadir o modificar los contactos.
- About: Se mostrarán los detalles de versión, motivo de la realización, autor de la aplicación y tutor.
- Contactos: Se mostrará la lista de los contactos del teléfono, para así poder seleccionarlos y añadirlos como contactos a visualizar.
- Cuadros de diálogo: Sus motivos principales son mostrar al usuario avisos o modificar características de la aplicación.

4.2.3. Clases desarrolladas

Como toda aplicación Android, el bloque principal de ésta es una actividad principal de la que van derivando otras para complementar su labor. Tanto la MainActivity como el resto se explicarán más adelante, pero en rasgos generales, las diferentes clases tendrán unas funciones una vez iniciadas sus actividades correspondientes:

- MainActivity: Es el esqueleto principal de la aplicación. Se encargará de mostrar el mapa y de dar acceso a las funciones básicas. Permite desplegar un menú en la barra superior para entrar en las opciones, que serán básicamente los ajustes Settings y la información sobre la aplicación About.
- Settings: Reúne los ajustes de la aplicación. Aquí se puede ajustar el número de teléfono, cambiar las frecuencias de refresco de la posición de los contactos y del envío de la posición propia, y al mismo tiempo acceder a la actividad Contactos.
- About: Se trata de una actividad que muestra la información principal sobre la aplicación.

- Contatos: Con esta actividad se accede a la lista de contactos del dispositivo, para así poder seleccionar aquellos que queremos visualizar en nuestro mapa.
- ActualizarPosiciones: Se trata de una actividad en segundo plano (intentService), que una vez iniciada, se mantiene activa durante todo el ciclo de vida de la aplicación, y que está destinada a recoger la posición de los compañeros cada cierto tiempo.

A continuación se detalla en un diagrama de flujo, cómo interactúan estas clases entre sí, en la Figura 14.

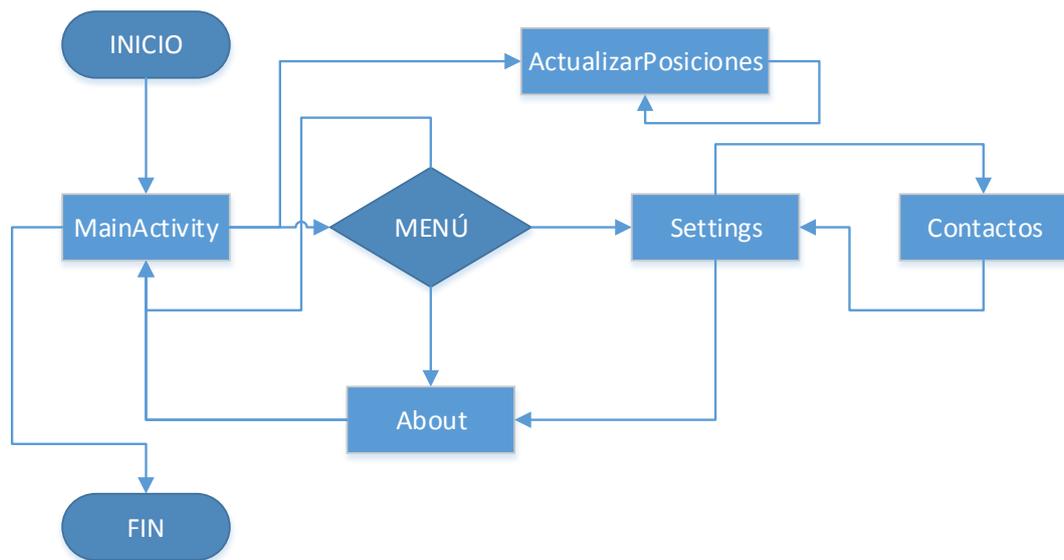


FIGURA 14: DIAGRAMA DE FLUJO DE CLASES

4.2.4. Protocolo con servidor

4.2.4.1. Introducción

Ahora se van a explicar en primer lugar los comandos de interacción con el cliente de forma detallada, y posteriormente el protocolo utilizado para encapsular los datos de comunicación entre cliente y servidor.

4.2.4.2. Interacción con cliente

Como se ha definido en los casos de uso, esta interacción se produciría a partir de tres situaciones en las que el cliente accedería al servidor mediante tres puntos de acceso, descritos a continuación. Los tres, están colgando de la dirección web <http://plato.it.uc3m.es/~jvillena/locationserver/>, y para la comunicación se utiliza el protocolo JSON, y se muestra cómo se construye el encapsulamiento para el último caso, ya que para los dos primeros el servidor sólo encapsula un mensaje de contestación.

4.2.4.2.1. *Recibir posición del cliente*

El cliente mandará el comando “setLocation”, junto con el número de teléfono y su posición, mediante una petición HTTP al servidor.

Un posible ejemplo de envío de mensaje del cliente sería:
<http://plato.it.uc3m.es/~jvillena/locationserver/setlocation?number=628056657&location=40.35820462,-3.69091522>

Como se puede ver, tras el “setLocation”, estaría el comando “¿number=” al que seguiría el número de identificación del propio del cliente. A continuación, seguiría el “&location=” con las coordenadas físicas en la que se encuentra.

El servidor al recibirlo, almacenará estos datos en el caso de que se haya construido bien la petición, respondiendo con un “status=OK”, o en caso contrario mandará un mensaje de “status=ERROR”.

4.2.4.2.2. *Seleccionar compañeros*

Cada vez que el cliente quiera añadir o modificar la lista de compañeros, mandará al servidor un comando de “setColleagues”, en el que se especificará tanto número propio, como lista de contactos a añadir, mediante una petición HTTP como la que se muestra a continuación.

<http://plato.it.uc3m.es/~jvillena/locationserver/setcolleagues?number=660635038&colleagues=628056657|649387276>

En esta petición se puede ver que la url está compuesta de la dirección del servidor seguida de “setcolleagues?number=”, al que se añadirá el número del cliente que hace la petición, y posteriormente “&colleagues=” que a continuación enviará los números de los contactos añadidos separados por “|”.

El servidor almacenará estos datos respondiendo con un “status=OK” o “status=ERROR”, al igual que en el caso anterior, dependiendo de si está bien construida la url.

4.2.4.2.3. *Enviar localizaciones de los compañeros*

Por último, el cliente puede enviar al servidor, mediante otra petición HTTP, el comando “getLocations”, en el que especificará el número propio, con el propósito de recoger todas las posiciones de los contactos añadidos. Un ejemplo podría ser este:

<http://plato.it.uc3m.es/~jvillena/locationserver/getlocations?number=628056657>

La construcción es tan sencilla como marcar la url del servidor, junto con el comando “getlocations” acompañado de “?number=” y el número del cliente.

El servidor contestará con un mensaje JSON, como el que se muestra a continuación:

```
JSON
├── status : "OK"
├── colleagues
│   ├── 609827348
│   │   ├── location : "40.28302166666666,-3.7996483333333337"
│   │   └── date : "2015-05-17 00:11:33"
│   ├── 649387276
│   │   ├── location : "40.3579468,-3.6909139"
│   │   └── date : "2015-05-20 16:30:45"
│   ├── 660635038
│   │   ├── location : "39.6517494,-4.7851914"
│   │   └── date : "2015-08-27 12:13:11"
│   ├── 660971281
│   │   ├── location : "40.3247016,-3.773646"
│   │   └── date : "2015-07-16 00:35:07"
│   └── location : "40.35820462,-3.69091522"
│       └── date : "2015-09-17 19:42:12"
```

Se puede observar que se envía un "OK", mostrando que se ha realizado bien la petición, junto con los números de los contactos, en los que se muestra su posición y la última vez que se conectaron. También se incluye la posición propia, junto con la última conexión.

También puede enviar un mensaje de error si ha habido algún problema en la petición.

4.3. Diseño visual

4.3.1. Introducción

Una parte importante, en cuanto a implementación se refiere, reside en la representación visual en el dispositivo de la aplicación, que a fin de cuentas es el modelo visual que el usuario tendrá y con el que interactuará de una forma u otra.

A continuación, se explicarán las diferentes vistas de las que dispone la aplicación.

4.3.2. Layouts

4.3.2.1. Actividad principal (activity_main)

En la Figura 15, se aprecia el diseño de layout de la pantalla de la MainActivity, que será la que el usuario verá la mayor parte del tiempo que haga uso de la aplicación.

La mayor parte estará ocupada por un fragment que contendrá el mapa interactivo sobre el que se podrán visualizar la posición propia junto con las distintas posiciones de los compañeros.

Sobre este fragment, se han introducido tres botones, para dar al usuario mayor facilidad en la interacción con la aplicación. El primero de ellos será el botón de “Refresh”, que estará asociado al método homónimo definido en la implementación de código, y cuya función será la de actualizar inmediatamente los marcadores de los contactos en el mapa. El segundo será el “Fit to Markers”, que se encargará de ampliar o reducir el mapa, buscando que éste incluya al conjunto de todos los marcadores y posición propia, para así poder contemplar el escenario completo. Estos dos estarían situados en la parte inferior izquierda y central de la pantalla, pero aún faltaría uno más, que sería el que está en la parte superior derecha con forma de brújula, configurado conjuntamente con el mapa, para así poder dar al usuario la opción de centrarse en su posición.

También es importante reseñar que a la derecha de los botones inferiores se ha dejado un hueco de forma consciente, ya que, como se puede ver en la pantalla, al pulsar sobre cualquiera de los marcadores de alguno de los contactos, aparecerán en el susodicho dos seleccionables aplicados a la posibilidad de establecer una ruta hacia él o visualizarlo en la aplicación de Google Maps.



FIGURA 15: LAYOUT ASOCIADO A LA ACTIVIDAD PRINCIPAL

4.3.2.2. *Contactos*

La actividad Contactos será la encargada de mostrar en una lista los usuarios con número de teléfono asociado que estén en la agenda del dispositivo.

La pantalla asociada (Figura 16) estará subdividida en un primer layout lineal que mostrará en la pantalla que se pueden seleccionar los contactos, junto con un

botón para confirmar una vez que se hayan seleccionado todos los compañeros que queremos visualizar.

El resto del layout consistirá en una lista visible, en la que se expondrán los contactos de la agenda acompañados de una casilla en la que se podrá seleccionar o deseleccionar para tener constancia de que se ha añadido.

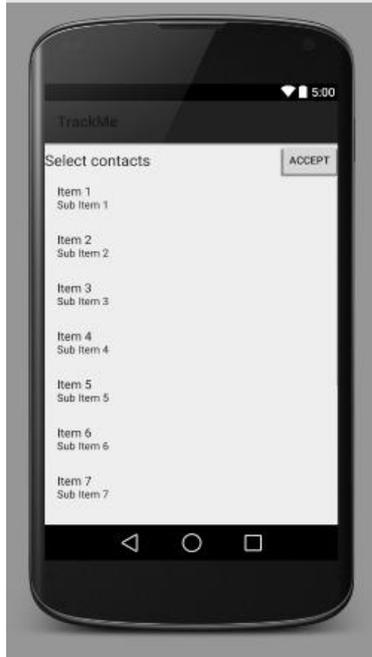


FIGURA 16: LAYOUT ASOCIADO A LA ACTIVIDAD CONTACTOS

4.3.2.3. *Settings*

Al lanzar la actividad de Settings, se dará acceso al menú para modificar así las características más relevantes de la aplicación. Aquí se mostrarán cuatro botones que corresponderán a distintas funciones, como se puede observar en la Figura 17:

- *Telephone Number.* Aquí se podrá tanto consultar como modificar el número propio. En la implementación de código recaería en `viewNumber` y `setNumber`.
- *Refresh Position Frequency* y *Refresh Contacts Frequency.* Ambos permiten tanto visualizar como modificar la frecuencia de envío de la posición propia al servidor y la de petición de los contactos. En el código corresponderían a las funciones homónimas.
- *Add Contacts.* Este botón permitirá iniciar la actividad Contactos, desplegando la pantalla explicada anteriormente.



FIGURA 17: LAYOUT ASOCIADO A LA ACTIVIDAD SETTINGS

4.3.2.4. *About*

La pantalla de About mostrará la información esencial de la aplicación, junto con la versión actual de ésta como se visualiza en la Figura 18.



FIGURA 18: LAYOUT ASOCIADO A LA ACTIVIDAD ABOUT

4.3.2.5. Cuadros de diálogo

4.3.2.5.1. Introducción

Los siguientes apartados se dedicarán a exponer los diferentes cuadros de diálogos creados para que el usuario pueda tanto modificar aspectos del funcionamiento de la aplicación, como ser advertido de las acciones que está tomando en su interacción.

4.3.2.5.2. Activar localización

El siguiente cuadro, mostrado en la Figura 19, emerge sobre la aplicación, si al iniciar ésta no se han activado los servicios de localización. Se le informa que debe de activar al menos los servicios de ubicación por redes móviles/wifi para así poder ser visible.

El usuario una vez visualizado este cuadro, podrá tomar dos opciones en cuanto a botones le aparecerán al final de éste. Primeramente, si decide pulsar Confirmar, será redirigido a los servicios de ubicación del dispositivo para así activarlos. Si no es así y pulsa Cancelar, la aplicación será cerrada.

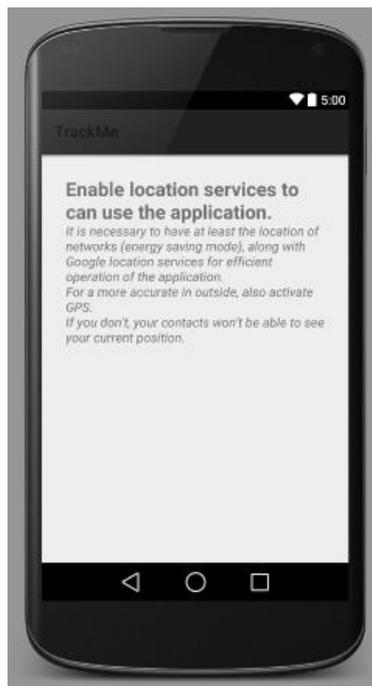


FIGURA 19: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO PARA ACTIVAR LA LOCALIZACIÓN

4.3.2.5.3. *Introducir número*

Este cuadro, expuesto en la Figura 20, está diseñado para que el usuario introduzca su número la primera vez que se inicia la aplicación.

Es importante recalcar que cuando se pulse sobre el espacio destinado a la introducción de éste, se desplegará el teclado numérico destinado a la introducción de números de teléfono, lo que facilitará el trabajo al usuario.

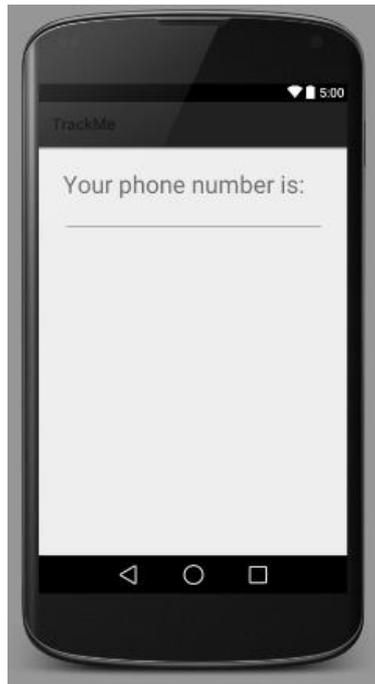


FIGURA 20: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO DE INTRODUCIR EL NÚMERO

4.3.2.5.4. *Mostrar número*

Este cuadro de diálogo representado en la Figura 21 se abre para consultar el número de teléfono propio, estando asociado al botón anteriormente explicado de la pantalla de Settings, Telephone Number.

A partir de aquí se podrá acceder a modificarlo a través del botón correspondiente, llegando a la pantalla de editar número explicada a continuación, o bien cancelar y dejarlo tal como está.

4.3.2.5.5. *Editar número*

Aquí se introduce el nuevo número, sobre el cuadro que se muestra en la Figura 22, con la misma característica explicada en el apartado 4.3.2.5.3, basada en la aparición del teclado numérico en forma telefónica para facilitar la introducción de los dígitos.



FIGURA 21: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO DE MOSTRAR EL NÚMERO

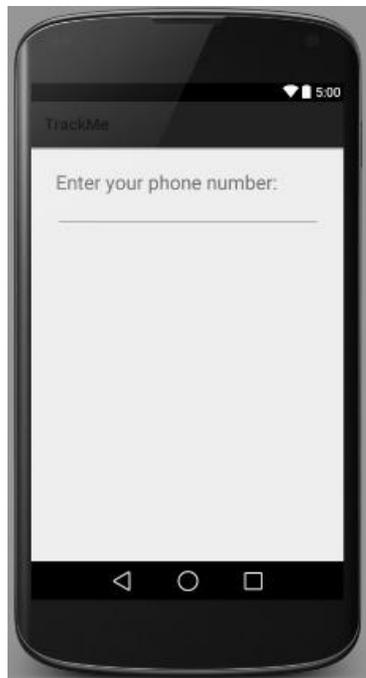


FIGURA 22: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO DE EDITAR EL NÚMERO

4.3.2.5.6. *Mostrar frecuencias*

Tanto en este apartado como en el siguiente se han aglutinado las frecuencias de refresco, tanto de la posición propia, como la de la petición de las de los contactos, ya que son análogas entre ellas.

En este apartado del cuadro de diálogo se muestra en la Figura 23 la frecuencia de refresco de la posición en segundos, a la que se podrá acceder a ella pulsado en modificar, que llevaría al cuadro de diálogo descrito en el siguiente apartado.

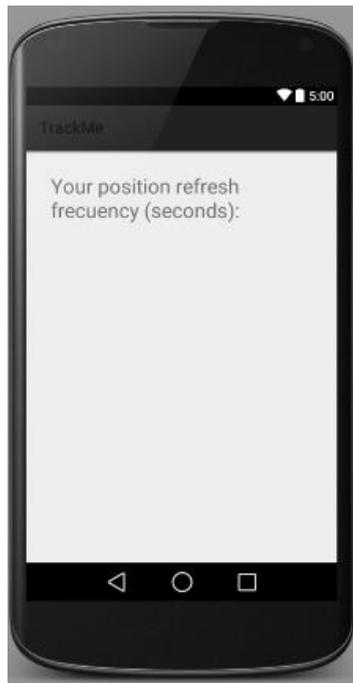


FIGURA 23: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO DE MOSTRAR LA FRECUENCIA DE REFRESCO DE LA POSICIÓN

4.3.2.5.7. *Editar frecuencias*

Para la modificación de las frecuencias, aparece el cuadro de diálogo de la Figura 24. También se favorece ésta mostrando al usuario que debe introducirla en segundos y apareciendo también un teclado numérico. Una vez realizada se podrá confirmar o cancelar en caso de que se quiera retroceder.

4.3.2.5.8. *Salir*

Este cuadro de diálogo representado en la Figura 25, está diseñado para que el usuario sea consciente de que si pulsa el botón de salir, cerrará la aplicación y con ello dejará de funcionar. Por ello, se podrá confirmar o cancelar una vez visualizado este cuadro.

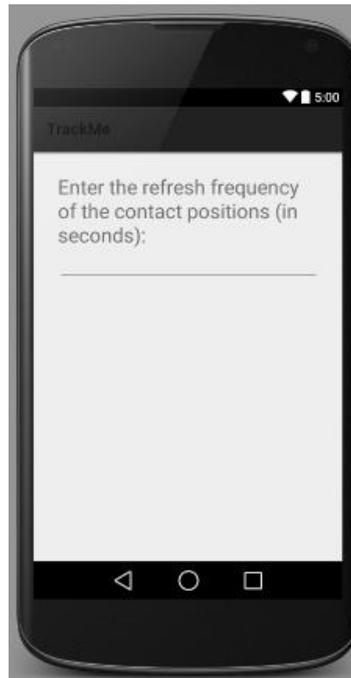


FIGURA 24: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO DE MODIFICAR LA FRECUENCIA DE REFRESCO DE LA POSICIÓN DE LOS CONTACTOS

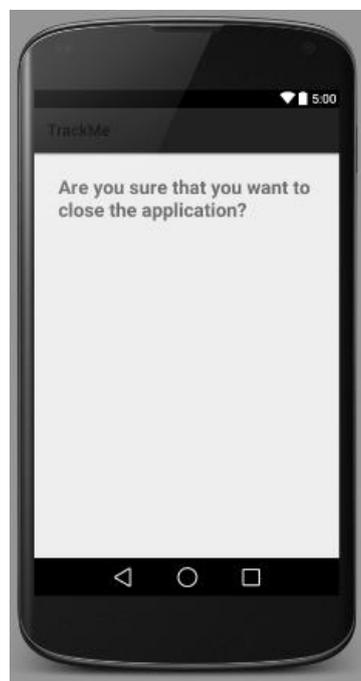


FIGURA 25: LAYOUT ASOCIADO AL CUADRO DE DIÁLOGO DE SALIR DE LA APLICACIÓN

4.3.3. Menú de la actividad principal (MenuMain)

En cuanto al menú desplegable de la aplicación, su diseño quedaría de la siguiente forma, dando acceso a la pantalla de ajustes Settings así como a la de información sobre la aplicación About.

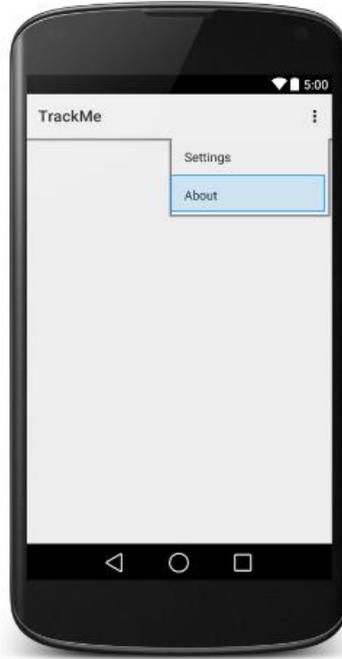


FIGURA 26: LAYOUT ASOCIADO AL MENÚ PRINCIPAL

4.4. Flujo de funcionamiento de las clases

4.4.1. Introducción

Antes de entrar en detalle del ciclo de vida de la aplicación en sí, es importante poner en contexto cómo funciona una aplicación Android en sí, para así partir de este básico en la aplicación desarrollada.

Toda actividad que se encuentra funcionando en un dispositivo Android, tiene que pasar por todas las fases del ciclo de vida. Como se puede ver en la Figura 27, los métodos que se suceden son los siguientes:

- onCreate(). Se realizan todas las inicializaciones, como la interfaz gráfica del usuario o estructuras de datos.
- onStart(). Indica que la actividad está a punto de pasar a primer plano.
- onResume(). Es el paso en el que el usuario puede interactuar con la aplicación.
- onPause(). Se le llama cuando la actividad va a pasar a segundo plano, en la mayoría de los casos porque otra actividad pasa a primer plano.
- onStop(). La actividad pasa a segundo plano y deja de ser visible al usuario. A partir de aquí el sistema operativo podría destruirla en el caso de que necesite memoria.

- onDestroy(). Es el método preámbulo de la destrucción de la actividad. Se le llama cuando hay una referencia "finish()" o si se pulsa la tecla de volver.
- onRestart(). Indica la vuelta de la actividad desde onStop().

También es posible acceder al menú superior (si así se ha configurado), como se muestra en la parte derecha del diagrama de flujo de la Figura 27, en el que entra en juego la función "onOptionsItemSelected()" definida para contemplar las diferentes opciones que se quieren configurar.

No es necesario sobrescribir todos los métodos, sino sólo aquellos cuyo comportamiento queramos modificar, por lo que en los siguientes apartados sólo se mencionarán aquellos que dispongan de cambios en ellos.

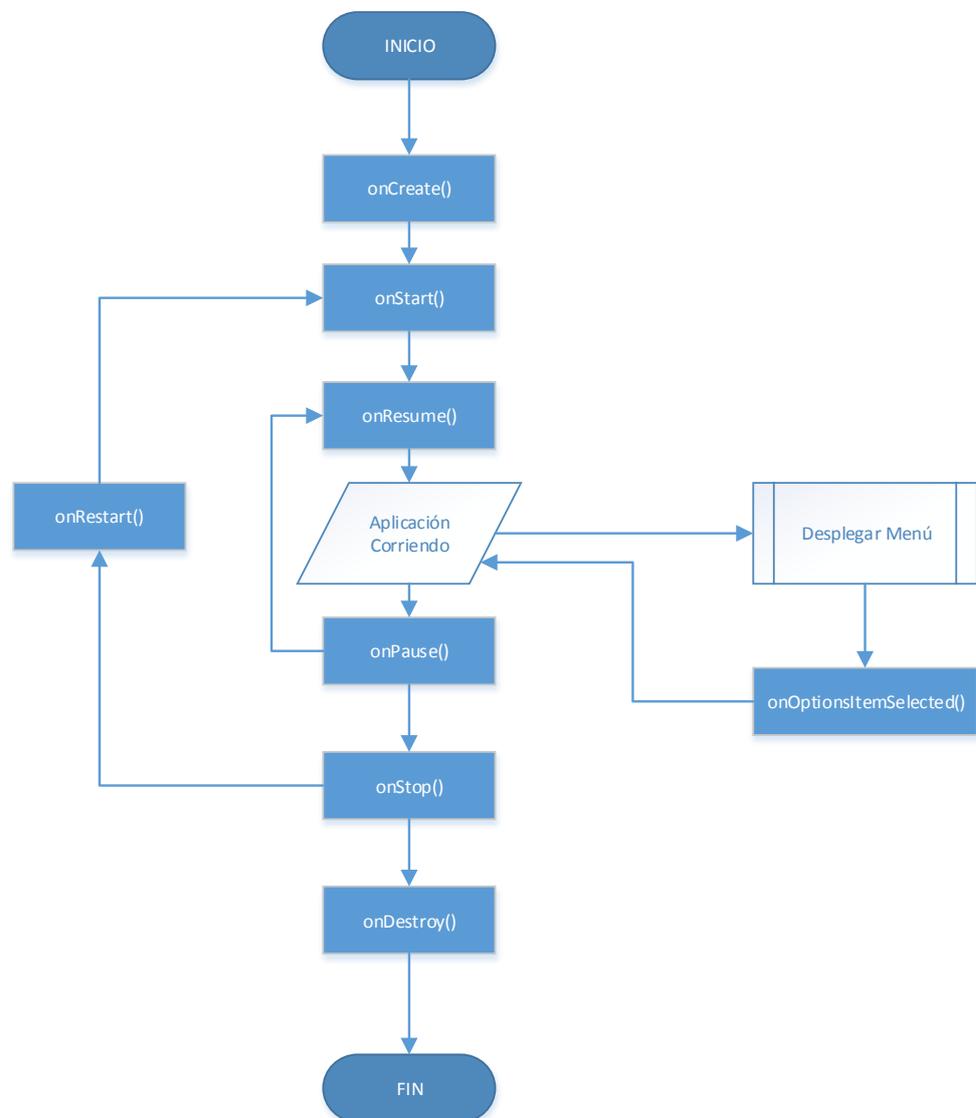


FIGURA 27: DIAGRAMA DE FLUJO DE UNA ACTIVIDAD ANDROID

4.4.2. Clase MainActivity

4.4.2.1. Actividad

La clase MainActivity define el flujo de la actividad principal, que aparecerá al iniciar la aplicación y dará lugar a la posibilidad de acceder a las distintas opciones del menú e iniciar nuevas actividades.

Primeramente se encargará de todas las inicializaciones necesarias, comprobar que la localización esté activa, así como activar la escucha del cambio de posición y representar el mapa en la pantalla.

Una vez que los servicios de localización han encontrado la posición, se representa en el mapa, focalizando a un nivel de barrio, que dará al usuario un contexto de escenario fácil de interpretar. Al mismo tiempo se activa la actividad ActualizarPosiciones, que estará corriendo en segundo plano mientras la aplicación esté activa, y que se encargará de pedir cada cierto tiempo las posiciones de los compañeros añadidos. Ésta se estudiará más adelante en profundidad en su apartado correspondiente.

A continuación, se exponen los principales métodos que se utilizan en la MainActivity.

4.4.2.2. Métodos

4.4.2.2.1. onCreate

Como se ha explicado anteriormente, este método será el encargado de las inicializaciones de las variables y estructuras de datos que se van a utilizar, así como de la interfaz gráfica. Esto es así en rasgos generales, pero específicamente, el diseño ha dado lugar a distintas métodos y órdenes adicionales, los cuales se muestran en la Figura 28¹.

Tras la inicialización de las principales variables, constantes y estructuras de datos a utilizar, se procede a la comprobación del número de teléfono, inicialización del mapa, y la activación de la localización, las cuales se profundizarán más adelante.

Posteriormente, se busca el mejor proveedor en base a unos criterios, entre los que se ha especificado que no tenga en cuenta coste ni altitud, ya que se busca tener un rango grande de definición de la localización. Tras seleccionarlo, se accede a la última posición almacenada por éste, utilizándola como primera posición propia de localización.

Finalmente, se activa la escucha del manejador de localización, la cual será útil para conocer y reaccionar ante los distintos cambios de posición propia, y habilitación, inhabilitación o cambio de los distintos sistemas de localización disponibles en el dispositivo. Se le definen los parámetros de frecuencia de comprobación (por defecto 30 segundos, pero configurable

¹ De aquí en adelante se muestran como subprocesos en los diagramas de flujo (contenedores con fondo blanco) las funciones auxiliares que se explicarán en el apartado posterior correspondiente.

posteriormente) y la distancia de actualización (50 metros), que activarán al manejador de localización cuando se cumpla una de las dos medidas.

4.4.2.2.2. *onResume*

En este método el usuario ya podrá entrar en contacto con la aplicación en sí, pudiendo visualizar tanto su posición propia, como la de los contactos que se hayan añadido. El flujo que seguirá el proceso se representa en la Figura 29.

Primeramente, se comprueba que el proveedor de localización haya encontrado la posición propia.

En caso afirmativo, se llamarán a las subrutinas *obtenerPosicion* y *moverCamara*, que a rasgos generales, obtienen la latitud y longitud propias para a continuación centrar el mapa sobre ésta.

Si no, significará que el manejador de localización no ha sido capaz de recoger la última posición conocida, por lo que se mostrará un mensaje en pantalla de que se encuentra en ello. Esto es importante, ya que el método de *onResume* será llamado cada vez que la aplicación vuelva a primer plano, por lo que es posible que tarde en encontrar la posición, pero una vez que lo consiga, cada vez que se reactive la app se focalizará sobre la posición propia para poner al usuario en situación.

La siguiente decisión viene dada por un problema de concurrencia en el código, ya que puede que se lleguen a pedir las posiciones de los contactos a visualizar al servidor sin disponer aún la aplicación del número propio del terminal, lo que hará que se produzca un error. De ahí que se compruebe antes de llamar al método *pedirPosiciones*, al que se accederá sólo en caso afirmativo, mientras que en caso negativo se mostrará un mensaje en pantalla acorde.

4.4.2.2.3. *onStop*

Se llega a esta rutina, una vez que la aplicación ha pasado a segundo plano, y por tanto se buscará dejar de realizar las funcionalidades que se realizaban de cara al usuario, mostrándose en la Figura 30 su ciclo de vida.

En ella se puede ver cómo primeramente se heredará el *onStop* de la clase padre, y a continuación se llama a *requestLocationUpdates*, para así seguir recibiendo las actualizaciones de los cambios de posición propia.

A continuación se comprobará si la alarma está activa, para desactivarla en caso de que así sea. Ésta se encarga de lanzar cíclicamente el *intentService ActualizarPosiciones*, que recoge las localizaciones de los contactos cada cierto tiempo.

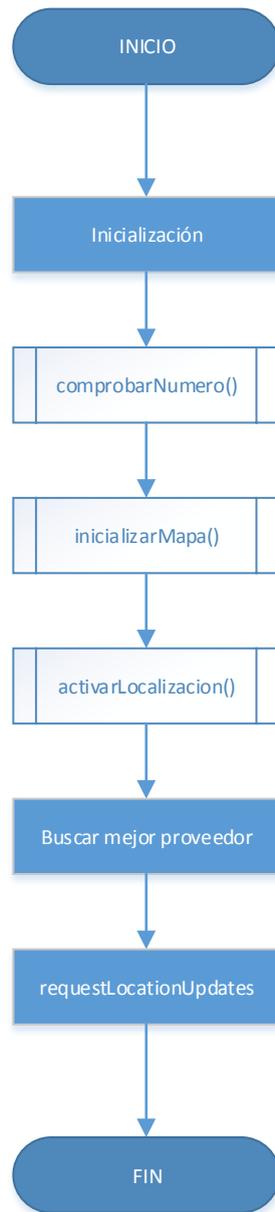


FIGURA 28: CICLO DE VIDA DEL MÉTODO "ONCREATE" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.4. *onRestart*

La rutina actual será un puente de nexos entre `onStop` y `onStart`, o lo que es lo mismo, entre que la aplicación está en segundo plano, a pasar al primero. Como se muestra en la Figura 31, el funcionamiento únicamente incluye la petición de actualizaciones de localización, debido a que ésta puede haber variado su frecuencia de refresco mientras la actividad principal estaba en segundo plano, por modificación directa del usuario.

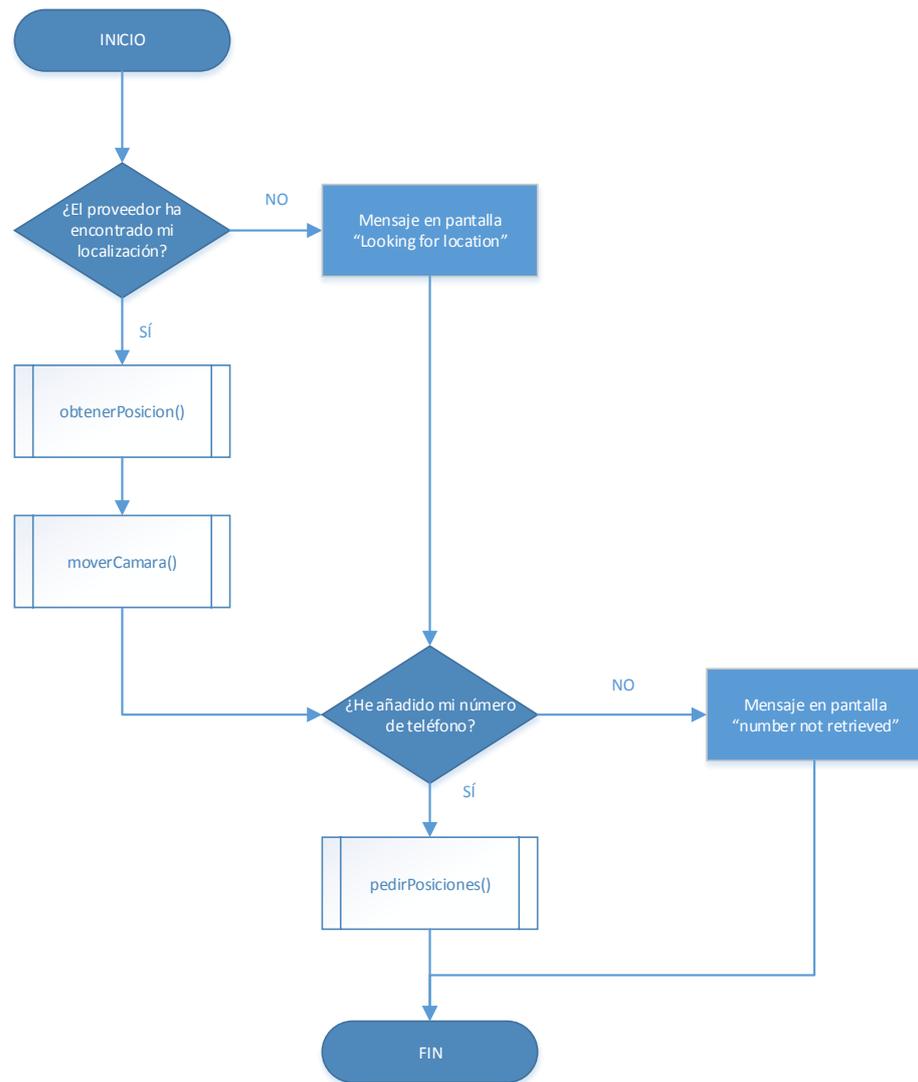


FIGURA 29: CICLO DE VIDA DEL MÉTODO "ONRESUME" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.5. *onDestroy*

El último método del ciclo de vida es el encargado de terminar la aplicación. A continuación en la Figura 32 se pueden ver los diferentes pasos que se siguen para ello.

Al igual que en el resto de métodos del ciclo de vida, se comienza heredando de la clase padre las características de la rutina, desconectando el receptor de eventos (encargado de mandar mensajes desde otras actividades a la principal), y el manejador de localización, para así dejar de recibir actualizaciones de los eventos de geolocalización propios. Una vez concluido se confirma por pantalla el cierre.

4.4.2.2.6. *Menú superior de controles*

4.4.2.2.6.1. *onOptionsItemSelected*

A este método se accederá al entrar en el menú superior de la barra de acción, y en él se seleccionará o entrar en los ajustes (Settings), o en el About. A continuación se puede ver el ciclo de vida del método, en la Figura 33.

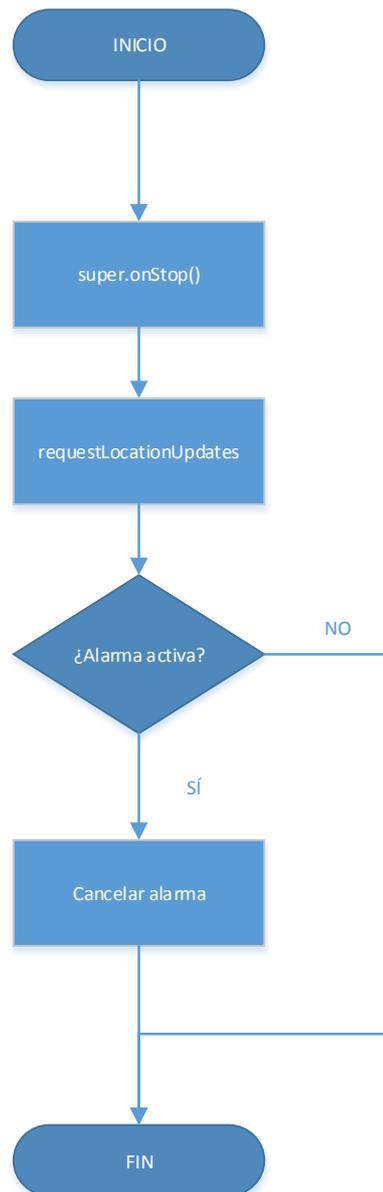


FIGURA 30: CICLO DE VIDA DEL MÉTODO "ONSTOP" DE LA ACTIVIDAD PRINCIPAL

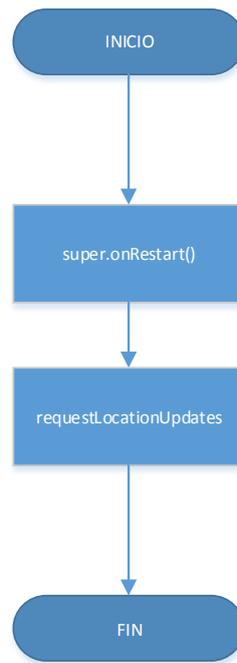


FIGURA 31: CICLO DE VIDA DEL MÉTODO "ONRESTART" DE LA ACTIVIDAD PRINCIPAL



FIGURA 32: CICLO DE VIDA DEL MÉTODO "ONDESTROY" DE LA ACTIVIDAD PRINCIPAL

Para arrancar la actividad de Settings, se le añadirá a la intención que la lanza el número propio, la dirección web del servidor, y las frecuencias de refresco de la posición y de los contactos. Estos datos son necesarios pasarlos, ya que pueden ser configurados en la actividad principal o en otra, pero siempre deben de estar en la primera para que se puedan acceder a ellos desde todas.

4.4.2.2.7. Auxiliares

4.4.2.2.7.1. Introducción

En este apartado se explicarán las funciones especificadas anteriormente como subrutinas, que cumplen una función específica dentro del código.

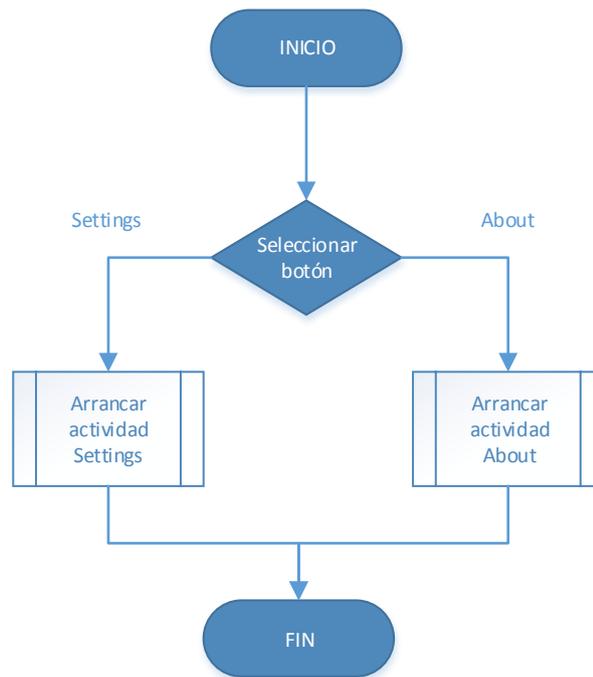


FIGURA 33: CICLO DE VIDA DEL MÉTODO "ONOPTIONSITEMSELECTED" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.7.2. *InicializarMapa*

Esta función está encargada de crear las condiciones para poder mostrar el mapa en la pantalla.

Para ello, primeramente se habilita la localización propia, como se muestra en el ciclo de vida de la Figura 34. Esto permite mostrar en el mapa un punto azul con nuestra posición junto con un botón en la esquina superior que permite centrar el mapa en ésta en cualquier momento. Se trata de una característica relevante ya que así el usuario no perderá el norte en ningún momento.

A continuación se selecciona el tipo de mapa a mostrar. En el diseño se ha elegido el tipo híbrido porque así muestra la mayor información posible, tanto nombre de las calles como aspecto del lugar.

Finalmente se crea un manejador de localización, para que a partir de él se puedan recibir actualizaciones de la localización y así vaya actualizándose en el mapa.



FIGURA 34: CICLO DE VIDA DEL MÉTODO AUXILIAR "INICIALIZARMAPA" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.7.3. *obtenerPosicion*

La función actual se encargará de obtener latitud y longitud propia, para así conseguir la posición propia y hacer uso de ella cuando sea necesario. El flujo se puede observar a continuación en la Figura 35.

4.4.2.2.7.4. *moverCamara*

El presente método moverá la cámara a la posición indicada, haciendo una animación hacia ella hasta el zoom indicado, como se puede ver en el diagrama de flujo de la Figura 36.

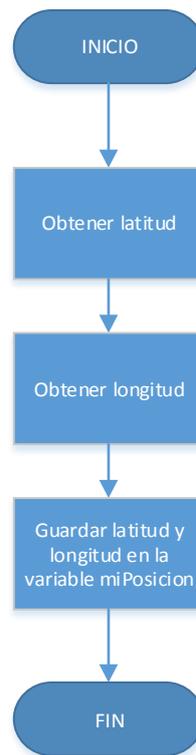


FIGURA 35: CICLO DE VIDA DEL MÉTODO AUXILIAR "OBTENERPOSICION" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.7.5. *representarColegas*

El método actual representa las posiciones de los amigos, mediante una lista de marcadores. Para ello hace uso de una variable global llamada listaContactos que almacenará los valores en latitud y longitud de los usuarios asociados al número de teléfono. El marcador discernirá si el usuario se ha conectado dentro de la última hora, en el mismo día o más allá de este, con colores azul, amarillo y rojo respectivamente.

En la Figura 37, se expone el diagrama de flujo que seguirá el método.

4.4.2.2.7.6. *addMarcador*

Este subproceso es llamado desde la anterior rutina, para representar en el mapa la posición, con los datos correspondientes al marcador. Éstos incluirán el nombre del contacto y su número de teléfono asociado, la última conexión y el color asociado a ésta para una visualización más rápida. La Figura 38 muestra el flujo de funcionamiento.



FIGURA 36: CICLO DE VIDA DEL MÉTODO AUXILIAR "MOVERCAMARA" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.7.7. *removeMarcadores*

Se llama a este método para poder refrescar los marcadores al actualizar la lista de contactos. El funcionamiento es tan simple como remover los marcadores uno a uno (Figura 39).

4.4.2.2.7.8. *enviarLocalizacion*

El presente método se encarga de enviar la localización propia al servidor, asimilando el papel del setLocation, representado en los casos de uso, el cual hace una petición HTTP al servidor con el número propio y la posición, para que éste lo almacene en caso de que no se produzca ningún error en la petición. Será llamado cuando el manejador de localización detecte un cambio en la posición o una activación de los servicios de geolocalización. Su flujo de funcionamiento se representa en la figura 40.

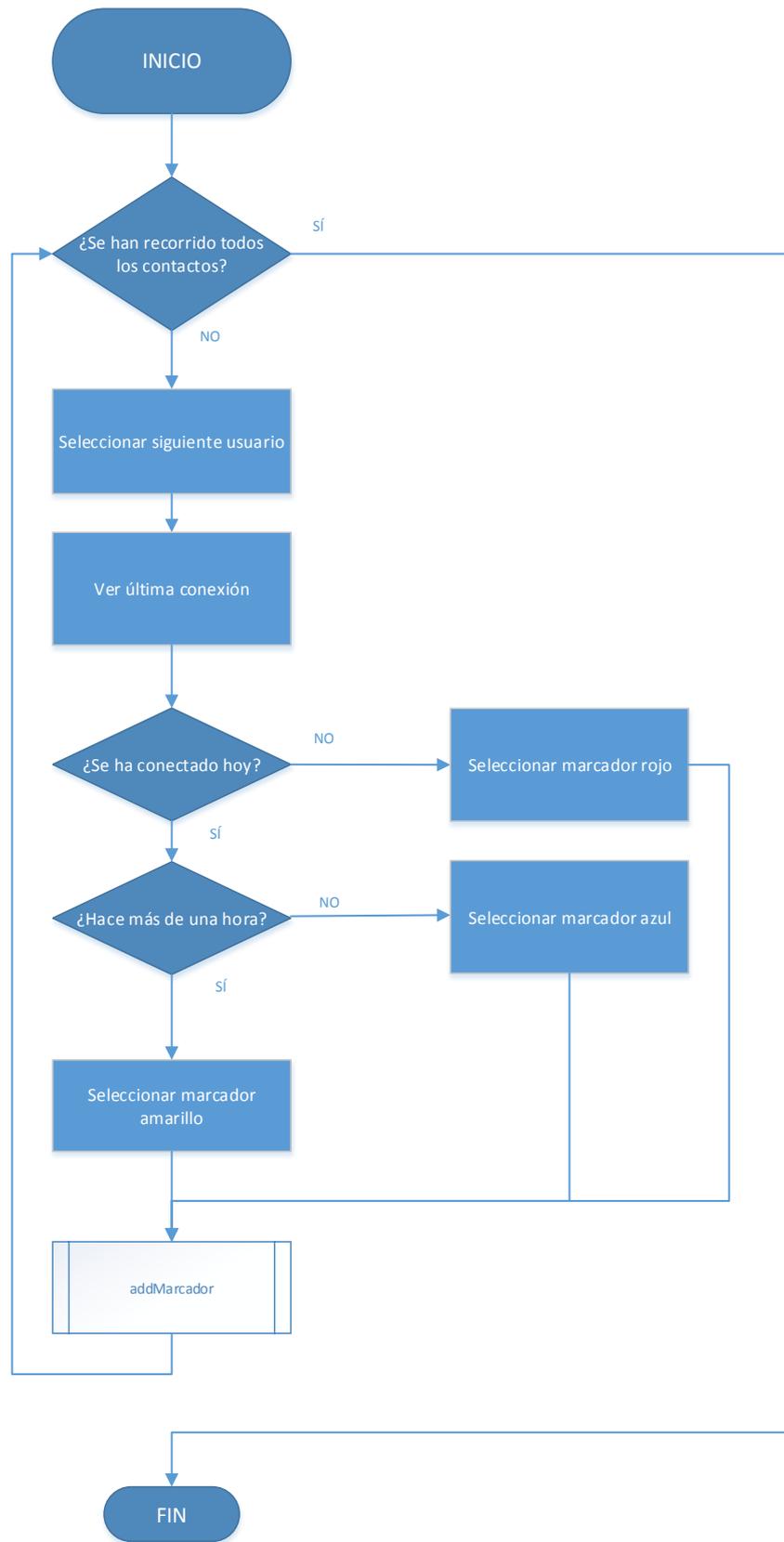


FIGURA 37: CICLO DE VIDA DEL MÉTODO AUXILIAR "REPRESENTARCOLEGAS" DE LA ACTIVIDAD PRINCIPAL



FIGURA 38: CICLO DE VIDA DEL MÉTODO AUXILIAR "ADDMARCADOR" DE LA ACTIVIDAD PRINCIPAL

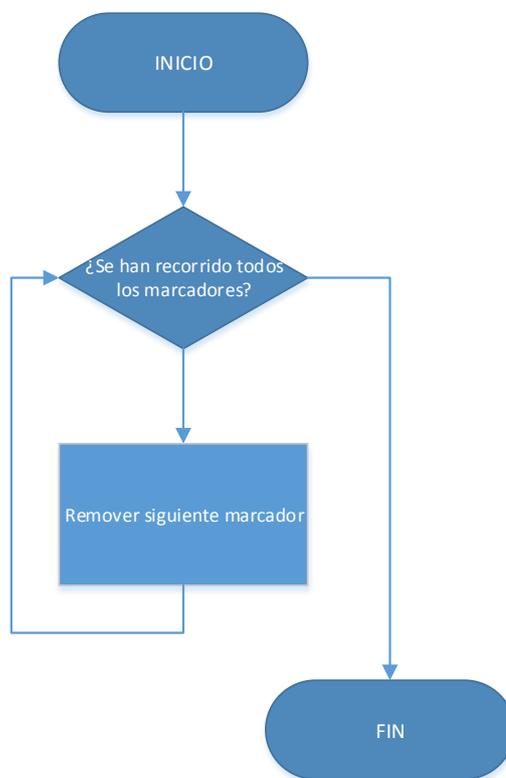


FIGURA 39: CICLO DE VIDA DEL MÉTODO AUXILIAR "REMOVERMARCADORES" DE LA ACTIVIDAD PRINCIPAL



FIGURA 40: CICLO DE VIDA DEL MÉTODO AUXILIAR "ENVIARLOCALIZACION" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.7.9. *comprobarNumero*

En la creación de la actividad, es necesario conocer y almacenar el número del usuario para poder identificarlo de cara al servidor. Por ello este método será llamado cada vez que se inicie la aplicación, para recuperar el número de teléfono como identificador propio. Si se trata de la primera vez que se inicia, aparecerá un cuadro de diálogo para introducirlo y almacenarlo en memoria. Si no, se recuperará el que ya esté guardado previamente. Estas situaciones se pueden observar en el ciclo de vida mostrado a continuación (Figura 41).

4.4.2.2.7.10. *pedirPosiciones*

Al llamar a esta rutina auxiliar, se activará el hilo ActualizarPosiciones, que en segundo plano y cada cierto tiempo de refresco especificado, solicitará las posiciones al servidor de los contactos añadidos. El flujo de funcionamiento es el de la Figura 42.

4.4.2.2.7.11. *refresh*

Éste método estará asociado al botón de la pantalla llamado "Refresh", que al pulsarlo, actualizará las posiciones de los compañeros sobre el mapa.

Funcionará de manera que cancelará la alarma anterior, para no poner en cola la intención que se lanzará a continuación, para hacer una llamada inmediata a la petición de posiciones de los contactos al servidor y así representarlos, como se puede seguir en el flujo mostrado en la Figura 43.

4.4.2.2.7.12. *ajustarMarcadores*

El presente método se encarga de recoger las posiciones de los compañeros añadidos y representarlas en el mapa junto con la posición propia, ajustando el mapa al conjunto de todos los marcadores y la localización del usuario. Así se consigue dar una visión de conjunto de todos los usuarios al mismo tiempo. El funcionamiento está representado a continuación en la Figura 44.

4.4.2.2.7.13. *activarLocalizacion*

A este método se accede cuando se inicia la aplicación y se comprueba si están activados los servicios de ubicación, al menos en localización por redes y wifi. Si no es así, se encarga de mostrar al usuario un cuadro de diálogo por pantalla indicándole que debe activarlos para que el programa pueda funcionar correctamente, llevándole a los ajustes del dispositivo. La subrutina se desarrolla como se muestra en la Figura 45.

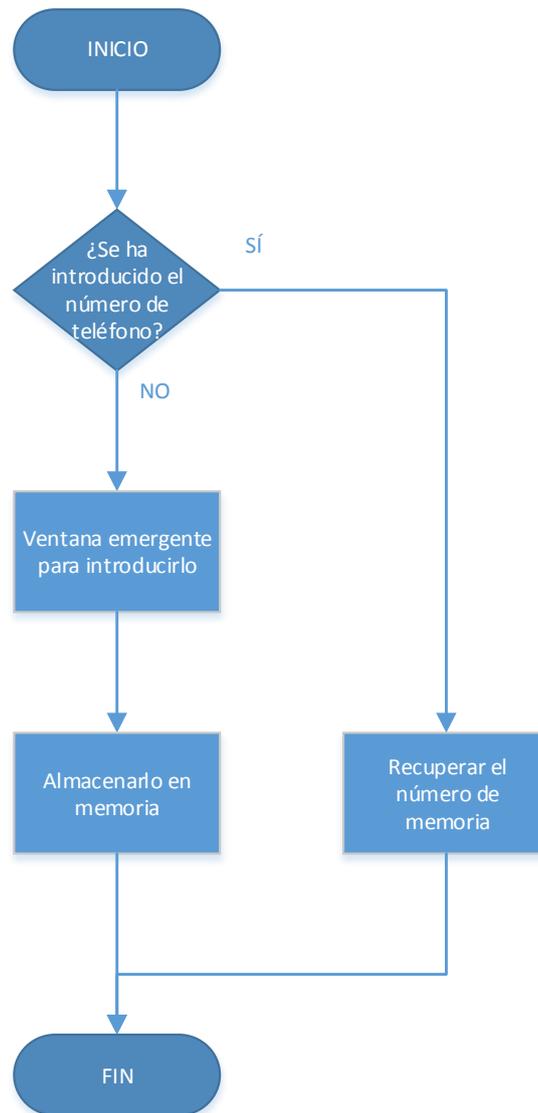


FIGURA 41: CICLO DE VIDA DEL MÉTODO AUXILIAR "COMPROBARNUMERO" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.8. *onKeyDown*

Este método se sobreescribe con el fin de crear un cuadro de diálogo para confirmar que se quiere cerrar la aplicación y dar así conciencia al usuario de que cesará el funcionamiento de la aplicación y evitarle un cierre accidental innecesario. Se puede observar el diagrama de flujo en la Figura 46.

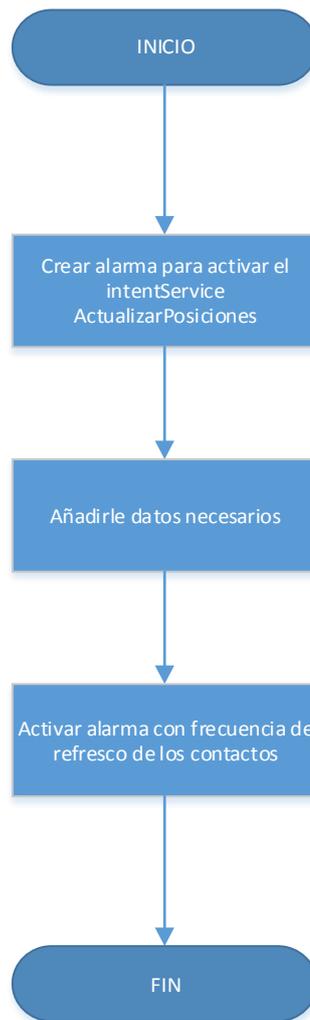


FIGURA 42: CICLO DE VIDA DEL MÉTODO AUXILIAR "PEDIRPOSICIONES" DE LA ACTIVIDAD PRINCIPAL

4.4.2.2.9. Clase ReceptorEventos

Dentro de la actividad principal se ha definido esta clase heredera de BroadcastReceiver, cuya función principal es recoger los mensajes mandados desde otras actividades, para que pueda tratarlos la MainActivity y así establecer un canal de comunicación con el resto de clases.

Sólo contendrá el método onReceive, que primeramente discernirá entre el tipo de mensaje recibido y a continuación actuará en consecuencia. Su diagrama de flujo se muestra en la Figura 47.



FIGURA 43: CICLO DE VIDA DEL MÉTODO AUXILIAR "REFRESH" DE LA ACTIVIDAD PRINCIPAL

4.4.3. Clase Contactos

4.4.3.1. Actividad

La clase Contactos está diseñada para crear una lista con todos los contactos del teléfono para poder seleccionarlos y añadirlos como compañeros.

A continuación se explicarán los métodos utilizados en esta actividad.

4.4.3.2. Métodos

4.4.3.2.1. onCreate

Este método es el que se encarga de crear el entorno necesario para que la actividad se desarrolle en condiciones, básicamente las inicializaciones.

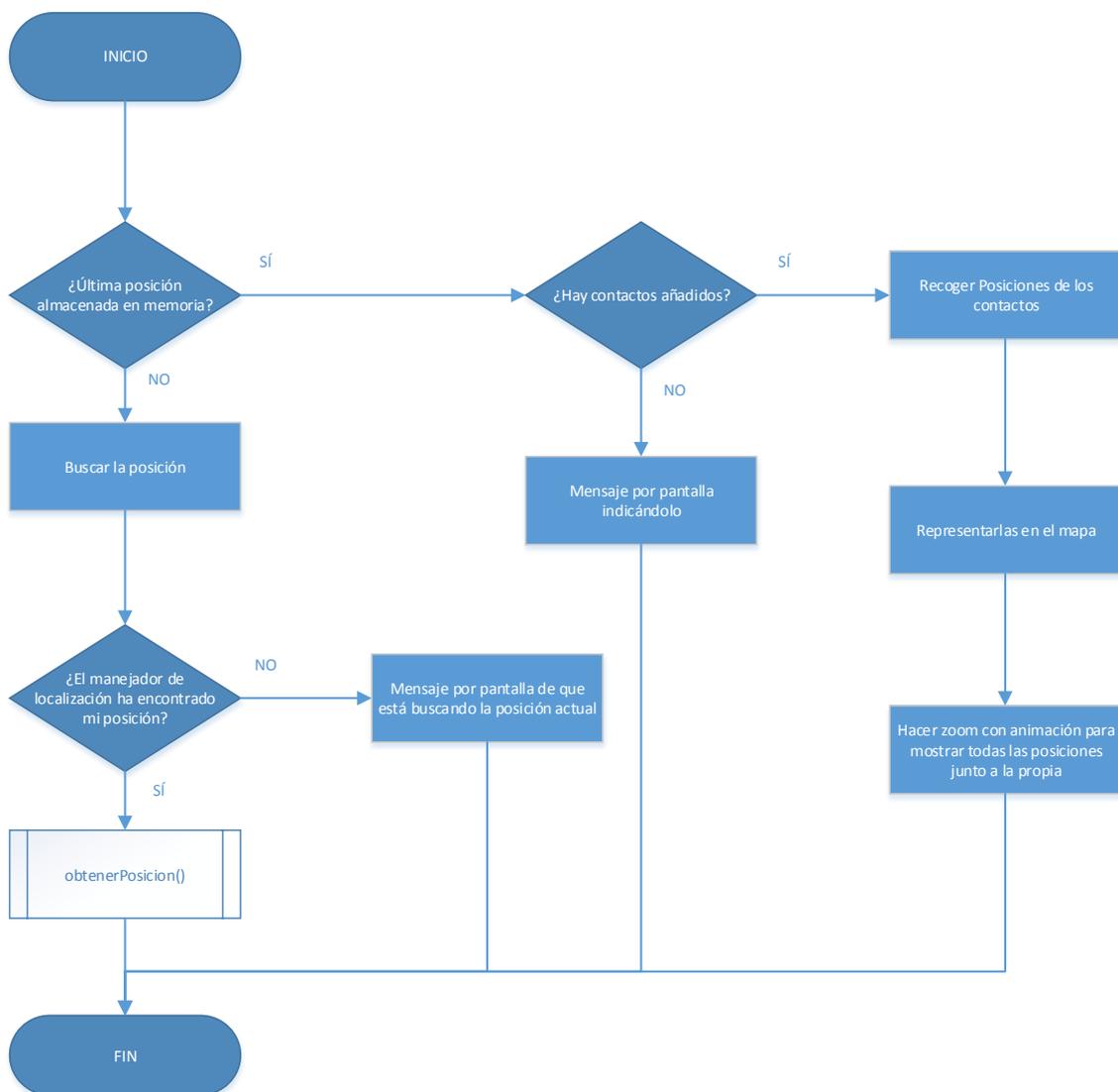


FIGURA 44: CICLO DE VIDA DEL MÉTODO AUXILIAR "AJUSTAR MARCADORES" DE LA ACTIVIDAD PRINCIPAL

En primer lugar recoge el número del terminal y la url del servidor, enviados al iniciar la actividad, y a continuación crea la lista de los contactos, la cual será de selección múltiple, mediante un cursor y adaptador. Ésta se filtrará de manera que se escojan aquellos contactos que dispongan de número de teléfono asociado.

Una vez seleccionados los contactos, el usuario deberá pulsar el botón de confirmar, con lo que se enviará al servidor la lista de los compañeros añadidos.

En la Figura 48 se muestra el ciclo de vida del método.

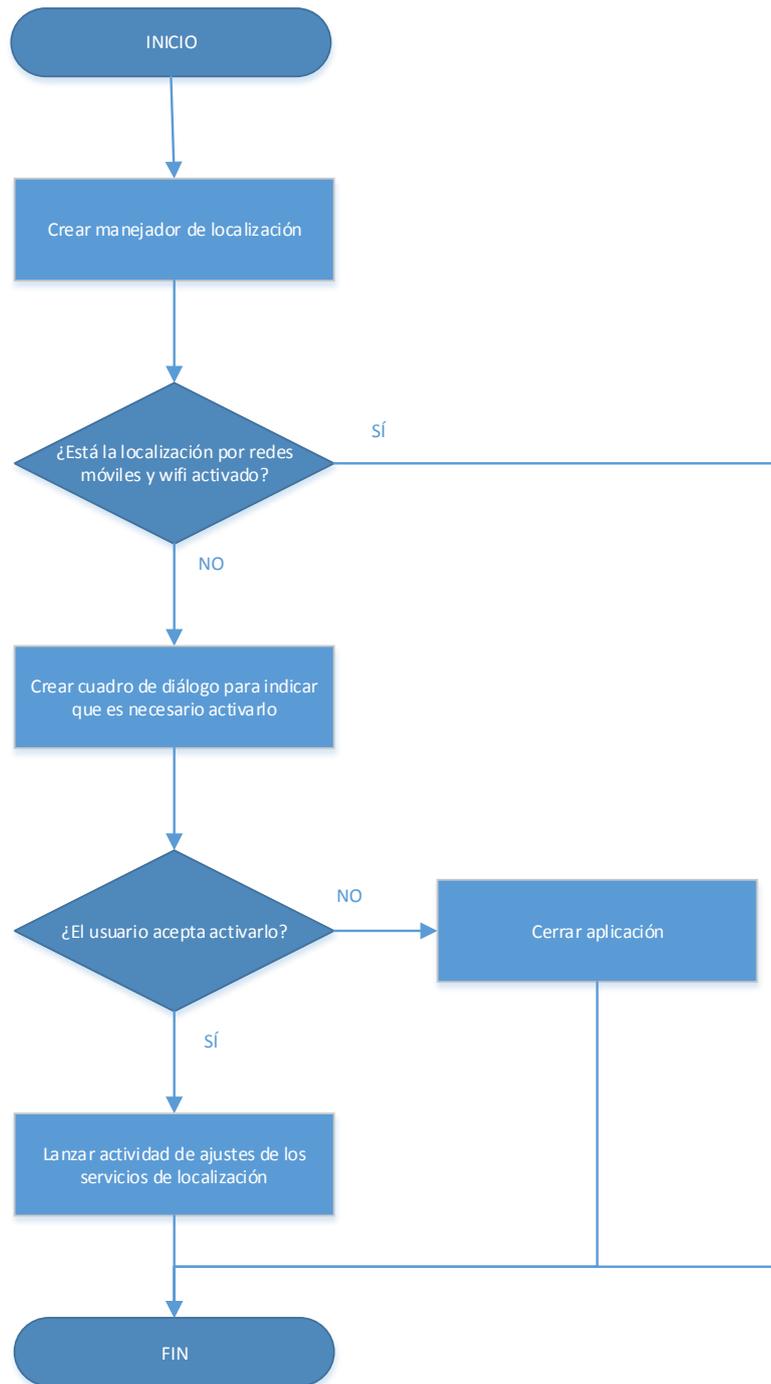


FIGURA 45: CICLO DE VIDA DEL MÉTODO AUXILIAR "ACTIVARLOCALIZACION" DE LA ACTIVIDAD PRINCIPAL



FIGURA 46: CICLO DE VIDA DEL MÉTODO AUXILIAR "ONKEYDOWN" DE LA ACTIVIDAD PRINCIPAL

4.4.3.2.2. *onDestroy*

En este caso, el método para cerrar la aplicación únicamente se ocupará de cancelar el cursor dedicado a mostrar la lista de contactos.

4.4.3.2.3. *onListItemClick*

Esta rutina está asociada a la actuación de la actividad cuando se pulse un botón de la lista. En este caso se añadirá a la lista de contactos, que luego se tomará para enviarla al servidor. También tendrá en cuenta que al deseleccionar el contacto se elimine el número de la lista. La secuencia del método se presenta en la Figura 49.

4.4.3.2.4. *enviarColegas*

La subrutina está diseñada para enviar al servidor la lista de contactos añadidos nada más pulsar la tecla de confirmar. Se recorrerá la lista de contactos, para ir concatenándolos en la petición HTTP y a continuación se construirá la url completa. A partir de aquí se realizará la petición en sí, mostrando por pantalla el resultado de ésta. Se puede ver el funcionamiento en la Figura 50.

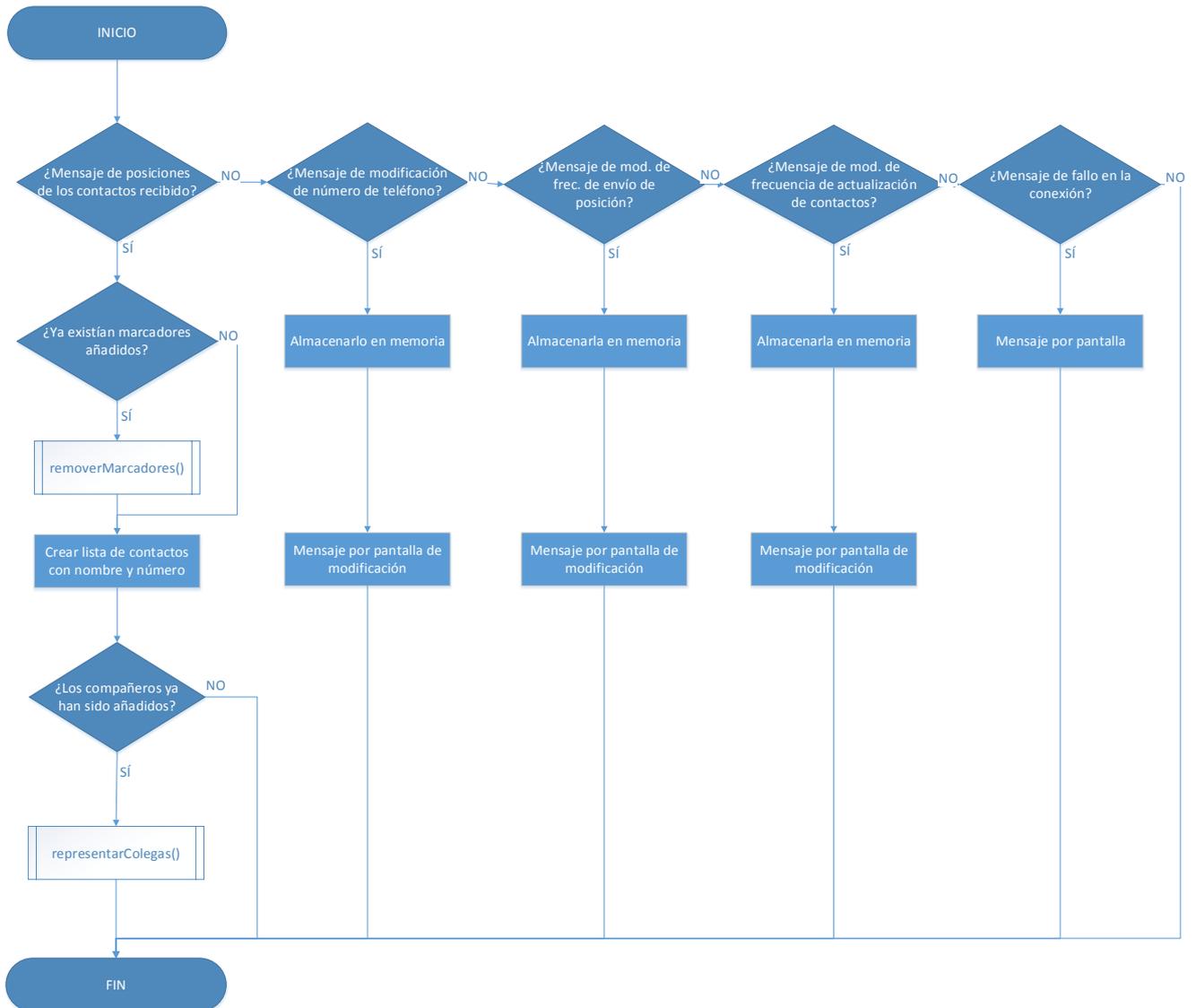


FIGURA 47: CICLO DE VIDA DEL MÉTODO "ONRECEIVE" DE LA CLASE RECEPTOREVENTOS DENTRO DE LA ACTIVIDAD PRINCIPAL

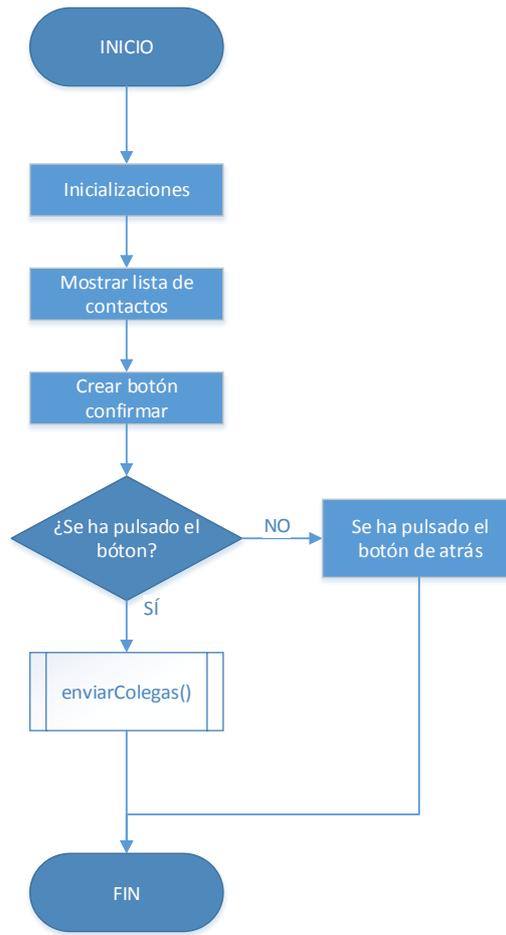


FIGURA 48: CICLO DE VIDA DEL MÉTODO "ONCREATE" DE LA CLASE CONTACTOS

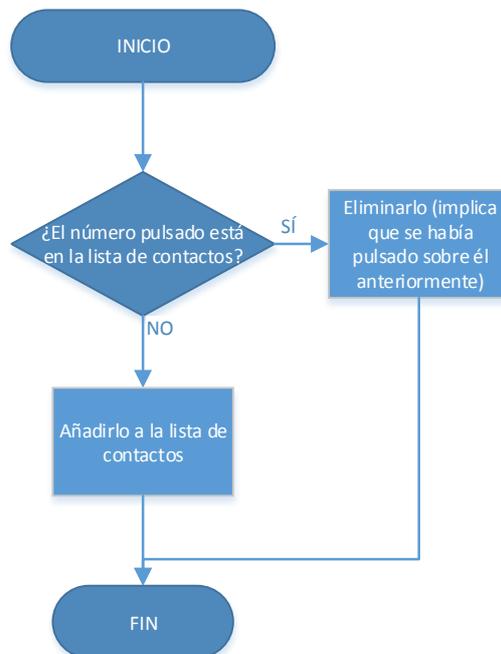


FIGURA 49: CICLO DE VIDA DEL MÉTODO "ONLISTITEMCLICK" DE LA CLASE CONTACTOS

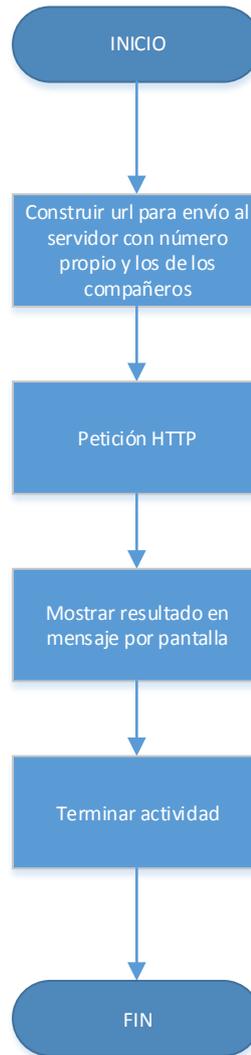


FIGURA 50: CICLO DE VIDA DEL MÉTODO "ENVIARCOLEGAS" DE LA CLASE CONTACTOS

4.4.4. Clase ActualizarPosiciones

4.4.4.1. Actividad

La clase ActualizarPosiciones será un hilo que, una vez activado, actuará en segundo plano con una cierta frecuencia de actualización, para así pedirle al servidor las posiciones de los contactos para así refrescar el mapa consecuentemente. Se correspondería a la función getLocation, presentada en los casos de uso.

Al tratarse de un IntentService, únicamente dispondrá de constructor y del método onHandleIntent, siendo este último el que aglutina las operaciones del hilo. En el siguiente apartado se explican más detalladamente.

4.4.4.2. Métodos

4.4.4.2.1. Constructor

Aquí hay que recalcar que al crear un `IntentService`, la declaración del constructor cambiará ligeramente, entrecomillando el nombre en la declaración "super", pero sin haber más cambios.

4.4.4.2.2. `onHandleIntent`

Aquí primeramente se pedirán los amigos al servidor, enviando el número de teléfono propio. Al recogerlos se descomprimirá el objeto JSON, para almacenar en una lista, por un lado, el número del compañero junto con su posición, y por otro éste junto a la última conexión. A continuación se mandará un mensaje a la actividad principal para que pueda hacer uso de estos datos y representarlos en el mapa. También se hará esto en caso de que se produzca un error, ya que están contempladas las excepciones en el código.

El diagrama de la Figura 51 muestra la sucesión de órdenes que se ejecutan.

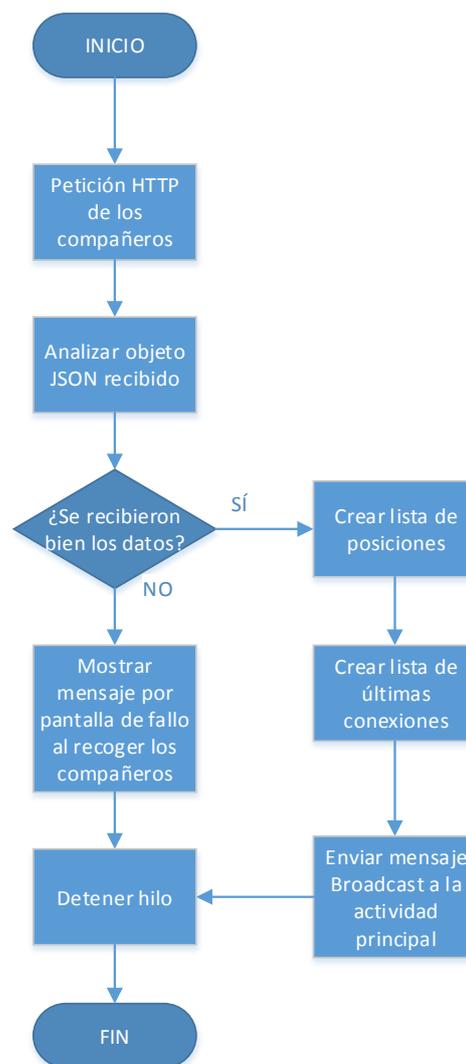


FIGURA 51: CICLO DE VIDA DEL MÉTODO "ONHANDLEINTENT" DE LA CLASE ACTUALIZARPOSICIONES

4.4.5. Clase Settings

4.4.5.1. Actividad

La actividad Settings es la encargada de desplegar la pantalla para poder modificar las diferentes opciones de la aplicación. Principalmente desde aquí se podrá modificar el número de identificación propio, las frecuencias de refresco propia y de los contactos y lanzar la actividad de añadir compañeros. El número del cliente estará continuamente almacenado en la memoria del teléfono de cara a futuros usos, mientras que los valores de las frecuencias se resetearán a su valor por defecto al reiniciar la aplicación.

A continuación se explican los métodos desarrollados en esta clase.

4.4.5.2. Métodos

4.4.5.2.1. onCreate

Encargado simplemente de inicializar las variables globales de la clase recogiendo los valores que se han transferido en la llamada desde la actividad principal.

4.4.5.2.2. viewNumber

Se trata del método llamado desde el botón "Telephone Number", para consultar el número actual y/o cambiarlo, mediante un cuadro de diálogo. El flujo queda reflejado en la Figura 52.

4.4.5.2.3. setNumber

Se trata de un método auxiliar para introducir el número de teléfono en caso de que queramos modificarlo. El funcionamiento es tan simple como se muestra en la Figura 53.

4.4.5.2.4. refreshPositionFrequency

El siguiente método está diseñado para ajustar la frecuencia de actualización de la posición propia, de manera que se podrá calibrar el envío de la localización al servidor manualmente, siendo éste por defecto de 30 segundos. Se realizará mediante un cuadro de diálogo de la manera que se muestra en la Figura 54, con la 55 mostrando el funcionamiento de la subrutina "setPositionFrequency".

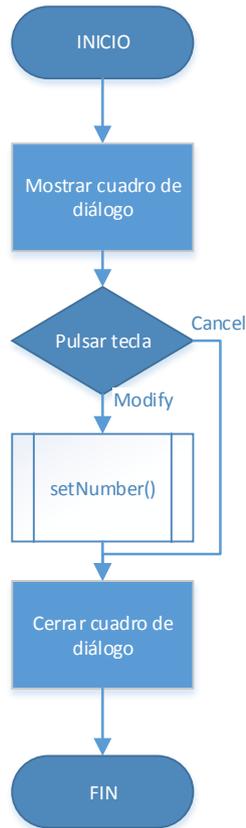


FIGURA 52: CICLO DE VIDA DEL MÉTODO "VIEWNUMBER" DE LA CLASE SETTINGS



FIGURA 53: CICLO DE VIDA DEL MÉTODO "SETNUMBER" DE LA CLASE SETTINGS

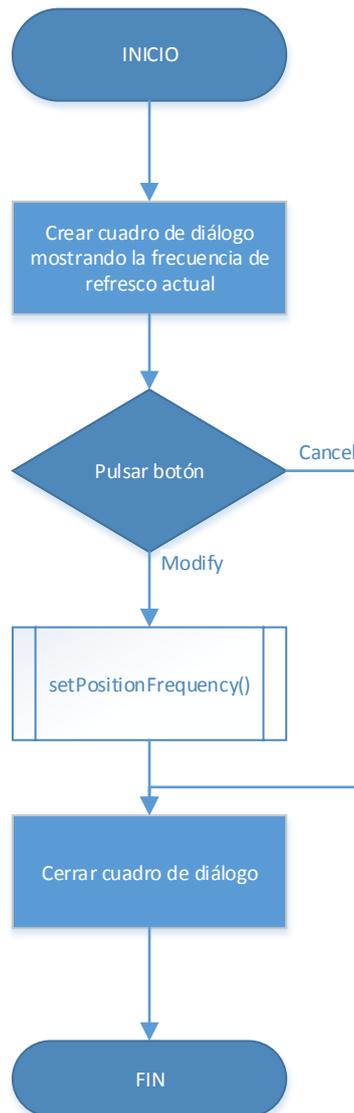


FIGURA 54: CICLO DE VIDA DEL MÉTODO "REFRESHPOSITIONFREQUENCY" DE LA CLASE SETTINGS

4.4.5.2.5. *refreshContactsFrequency*

Este método es análogo al anterior, pero su utilidad es para la frecuencia de petición al servidor de las posiciones de los compañeros, también siendo por defecto de 30 segundos. También dispone del método equivalente al auxiliar, que en este caso es llamado "setContactsFrequency". Tanto el principal como la subrutina siguen similares ciclos de vida que los representados en las Figura 54 y 55 respectivamente.

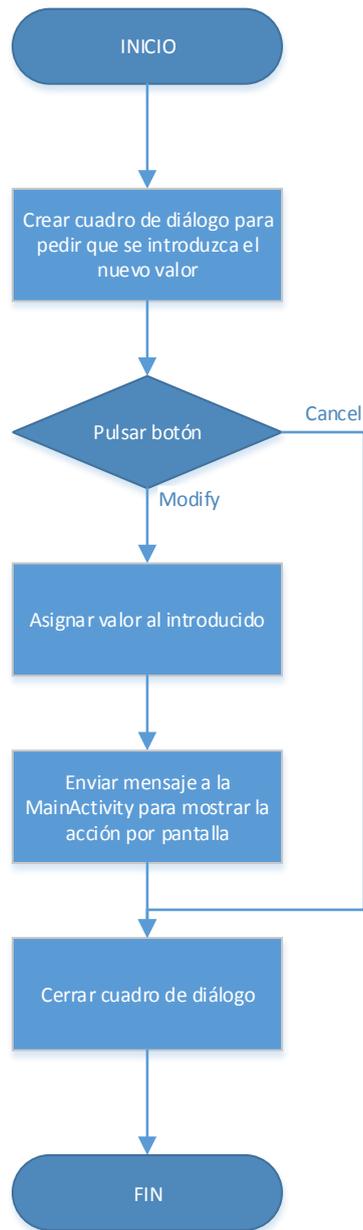


FIGURA 55: CICLO DE VIDA DEL MÉTODO AUXILIAR "SETPOSITIONFREQUENCY" DE LA CLASE SETTINGS

4.4.5.2.6. *addContacts*

Este método completaría el cuarto botón definido en la pantalla de la actividad Settings. Aquí simplemente se iniciaría la actividad Contactos para así poder añadir aquellos que queramos ver representados en el mapa.

4.4.6. Clase About

4.4.6.1. Actividad

Esta clase únicamente se encargará de representar la pantalla con las características de la aplicación y dispondrá únicamente del método onCreate, que lanzará a primer plano el layout propio.

4.5. Manifiesto

El manifiesto de una aplicación es el archivo en donde se definen las características principales de ésta, como son la versión, permisos, etc. A continuación se desganan en distintos puntos:

- Paquete y versión:

```
package="javiergarcia_pfc_uc3m.trackme"  
android:versionName="1.0">
```

Como se puede observar, se define el nombre que caracterizará al paquete de toda la aplicación, junto con el número de versión, que será modificable a medida que se vayan modificando o añadiendo distintas características.

- Permisos:

```
<!-- Permisos para:  
- acceder a recibir mapas de google (a través de una firma)  
- que el usuario pueda:  
  + acceder a internet  
  + acceder al estado de la red  
  + escribir en almacenamiento externo (a partir del API 19)  
  + leer los servicios de google y usar los mapas  
  + acceder a localización gruesa y precisa  
  + leer los contactos del teléfono  
  + leer el estado del teléfono para saber en número propio  
-->  
<permission  
  android:name="javiergarcia_pfc_uc3m.trackme.permission.MAPS_RECEIVE"  
  android:protectionLevel="signature"/>  
<uses-permission  
  android:name="android.permission.INTERNET"/>  
<uses-permission  
  android:name="android.permission.ACCESS_NETWORK_STATE"/>  
<uses-permission  
  android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-permission  
  android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>  
<uses-permission  
  android:name="javiergarcia_pfc_uc3m.trackme.permission.MAPS_RECEIVE"/>  
<uses-permission  
  android:name="android.permission.ACCESS_COARSE_LOCATION"/>  
<uses-permission  
  android:name="android.permission.ACCESS_FINE_LOCATION"/>  
<uses-permission  
  android:name="android.permission.READ_CONTACTS" />  
<uses-permission  
  android:name="android.permission.READ_PHONE_STATE" />
```

Aquí se definen todos los permisos definidos en la aplicación. A través de ellos se habilita que el usuario tenga la capacidad de recibir mapas, acceder a internet y al estado de la red, escribir en almacenamiento externo, leer servicios de Google y usar sus mapas, acceder a los servicios de localización (tanto GPS como redes) leer los contactos del teléfono así como el estado.

- Características adicionales de los mapas:

```
<!--Se indica que Google Maps requiere OpenGL
(estándar para representar gráficos en 2D y 3D).
Uso la 2.0 porque abarca a la mayor parte de los dispositivos (65%)
y el resto soportan la 3, que es compatible hacia atrás-->
<uses-feature
  android:glEsVersion="0x00020000"
  android:required="true"/>
```

Como bien se define en el comentario, es necesario especificar el requerimiento del OpenGL para poder representar los gráficos de Google Maps, usando la versión 2.0 que aglutina el mayor número de terminales.

- Definición de la aplicación:

Se divide en distintas partes que se explican a continuación.

- *Fundamentos*, donde se define que se pueda realizar almacenamiento de resguardo de la aplicación, se define el icono con el que se mostrará el acceso a la aplicación, el nombre que tendrá y el estilo general de ésta.

```
<!--definición de la aplicación-->
<application
  android:allowBackup="true"
  android:icon="@mipmap/ic_launcher"
  android:label="TrackMe"
  android:theme="@style/AppTheme" >
```

- *Versión Google Play y clave de Google Maps*. Primeramente hay que especificar la versión para que la aplicación pueda hacer uso de sus servicios y a continuación se introduce la clave obtenida para desarrollar aplicaciones que hagan uso de los mapas de Google.

```
<!--se le proporciona a la aplicación la versión de Google Play
y la clave para utilizar Google Maps-->
<meta-data
  android:name="com.google.android.gms.version"
  android:value="6587000" />
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="AIzaSyA76CUfVZO9-CL1EfcY3vv85F5CkNwF1b0"/>
<!--clave debug:AIzaSyA76CUfVZO9-CL1EfcY3vv85F5CkNwF1b0-->
```

○ *Descripción de las actividades*

- *MainActivity*, especificando en este caso que se trata de la actividad que se iniciará al lanzar la aplicación, mientras que el resto estarán definidas como "DEFAULT" en vez de "LAUNCHER".

```
<!-- descripción de las actividades -->
<activity
  android:name=".MainActivity"
  android:label="TrackMe" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

- *Contacts*

```
<activity
  android:name=".Contactos"
  android:label="contacts" >
  <intent-filter>
    <action android:name="javiergarcia_pfc_uc3m.trackme.contactos" />

    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

- *Settings*

```
<activity
  android:name=".Settings"
  android:label="Settings" >
  <intent-filter>
    <action android:name="javiergarcia_pfc_uc3m.trackme.settings" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

- *About*

```
<activity
  android:name=".About"
  android:label="About" >
  <intent-filter>
    <action android:name="javiergarcia_pfc_uc3m.trackme.about" />
    <category android:name="android.intent.category.DEFAULT" />
  </intent-filter>
</activity>
```

- *ActualizarPosiciones*, que en este caso se especifica como servicio y no como actividad, dado su carácter de funcionar en segundo plano como intentService.

```
<service
  android:name=".ActualizarPosiciones">
</service>
```

5. Manual de usuario

5.1. Introducción

Se explicará de forma didáctica a través de capturas de pantalla, cómo funciona la aplicación, para aprender a usarla con facilidad.

5.2. Manual

5.2.1. Configuración inicial

Al iniciar la aplicación lo primero que aparecerá será la pantalla de la Figura 56, que será la indicadora de que es necesario habilitar los servicios de geolocalización para que pueda funcionar correctamente. En caso de confirmar, se accederá a los servicios de ubicación del dispositivo para activarlos, y en caso contrario, saldrá de la aplicación.

Al volver a primer plano, aparecerá una ventana emergente para introducir el número de teléfono por primera vez como se puede ver en la Figura 57, y que tendrá la facilidad del uso de un teclado numérico mostrado en la Figura 58.

Una vez realizados los primeros ajustes, la pantalla aparecerá como en la Figura 59, en la que se puede ver el mapa centrado en la posición propia.

A continuación se deben añadir los contactos de los compañeros para poder visualizarlos en el mapa de la pantalla principal. Esto se consigue accediendo al menú superior como se muestra en la Figura 60 y pulsando la tecla Settings, para así acceder al menú que se visualiza en la Figura 61. Una vez aquí aparecerá una pantalla como la de la Figura 62, en la que se desplegará la lista con los contactos del teléfono, con un cuadrado para marcar todos los que se quieran seleccionar. Una vez hecho esto, dando a confirmar se vuelve a la pantalla principal, en la que ya se podrá visualizar la posición de los compañeros añadidos, mostrado en la Figura 63.

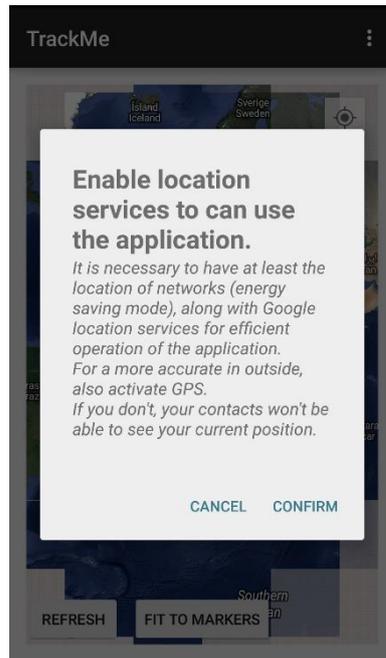


FIGURA 56: PANTALLA DE HABILITAR LA LOCALIZACIÓN

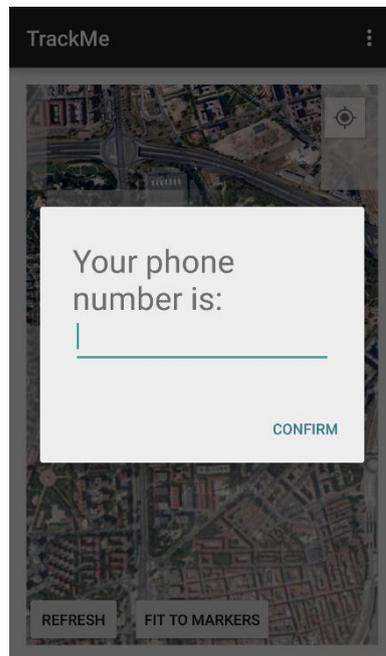


FIGURA 57: PANTALLA DE INTRODUCCIÓN DEL NÚMERO DE TELÉFONO POR PRIMERA VEZ

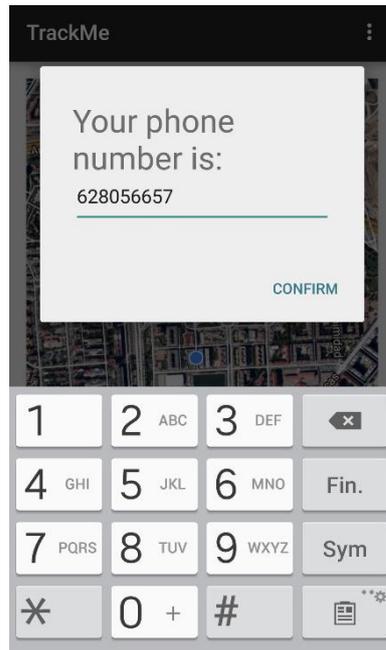


FIGURA 58: PANTALLA DE INTRODUCIR NÚMERO DE TELÉFONO CON TECLADO NUMÉRICO



FIGURA 59: PANTALLA PRINCIPAL DE LA APLICACIÓN



FIGURA 60: PANTALLA DE DESPLIEGUE DEL MENÚ SUPERIOR

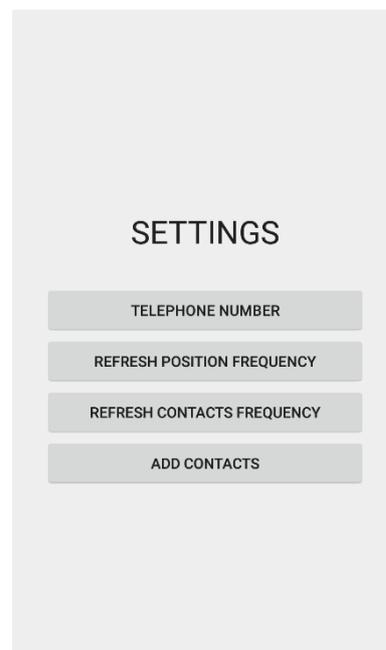


FIGURA 61: PANTALLA DE AJUSTES "SETTINGS"

5.2.2. Ajustar la pantalla a los marcadores

Una vez configurados los aspectos básicos de la aplicación, lo siguiente será ajustar el mapa a todos los contactos. Esto se podrá hacer pulsando el botón “Fit to markers”, visualizándose como en la Figura 64.

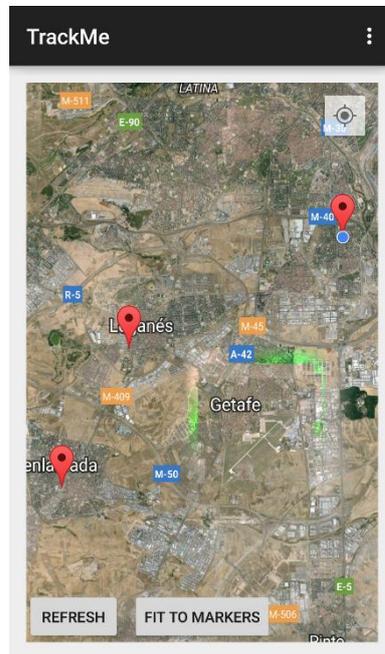


FIGURA 64: PANTALLA AJUSTADA AL PULSAR "FIT TO MARKERS"

5.2.3. Modificar frecuencias de refresco

También es posible modificar las frecuencias de envío de la posición propia al servidor y de recepción de las de los compañeros. Esto se hace accediendo desde la pantalla de “Settings” de la Figura 61 y pulsando bien sobre “Refresh Position Frequency” o “Refresh Contacts Frequency”. El primero de estos botones dará lugar a la pantalla representada en la Figura 65, con un valor por defecto de 30 segundos. Al pulsar el modificar, se accederá a la pantalla de la Figura 66 en el que se podrá introducir el nuevo dato.

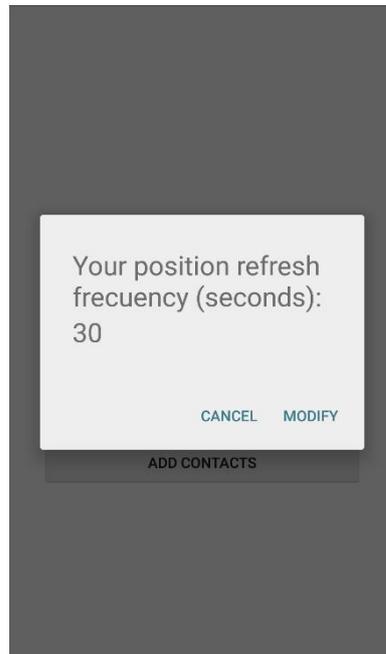


FIGURA 65: PANTALLA DE CONSULTA DE LA FRECUENCIA DE REFRESCO DE LA POSICIÓN

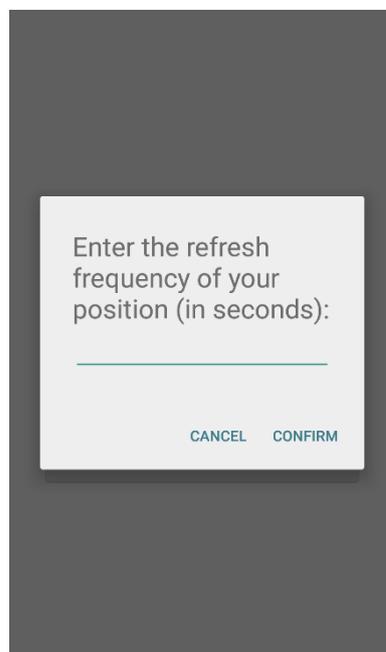


FIGURA 66: PANTALLA PARA INTRODUCIR UNA NUEVA FRECUENCIA DE REFRESCO DE LA POSICIÓN

Idéntico procedimiento se seguirá para hacer lo propio con la frecuencia de refresco de los contactos, pulsando sobre la segunda tecla y dando lugar a las Figuras 67 y 68.

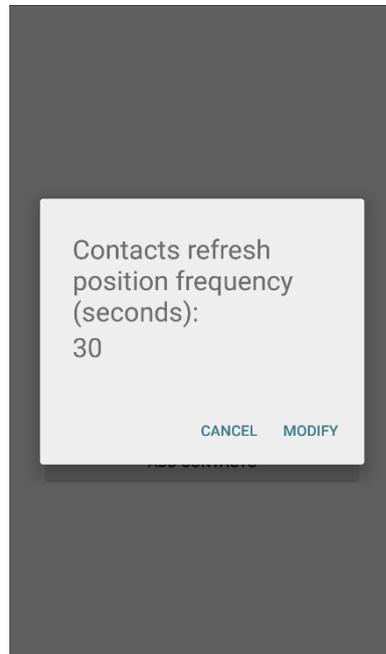


FIGURA 67: PANTALLA QUE MUESTRA LA FRECUENCIA DE REFRESCO DE LOS CONTACTOS

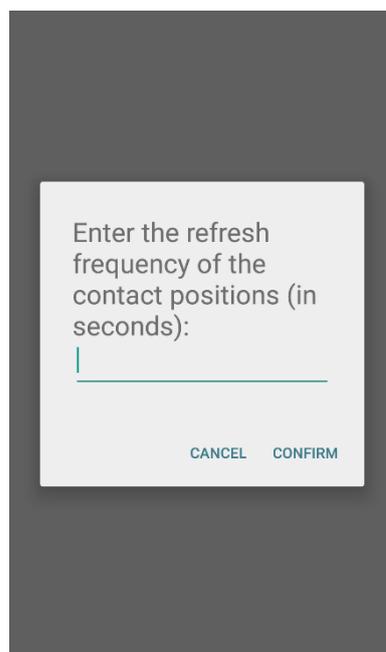


FIGURA 68: PANTALLA PARA MODIFICAR LA FRECUENCIA DE REFRESCO DE LOS CONTACTOS

5.2.4. Modificar número de teléfono

También se podrá acceder a cambiar el identificador del usuario, pulsando sobre la tecla "Telephone Number" de "Settings", accediendo a la pantalla de la Figura 69, en la que se mostrará el número actual, y pudiendo modificarlo, con lo que se accedería a la pantalla de la Figura 70.

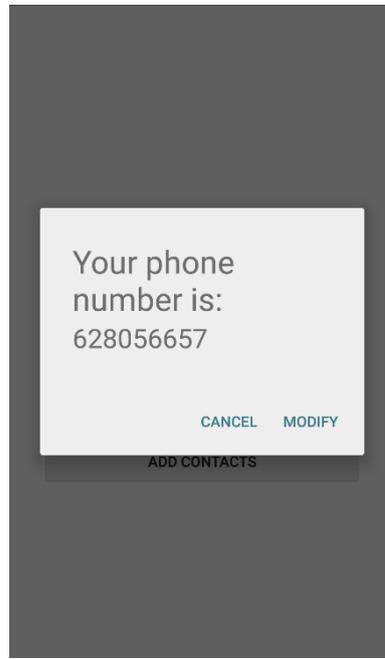


FIGURA 69: PANTALLA QUE MUESTRA EL NÚMERO PROPIO

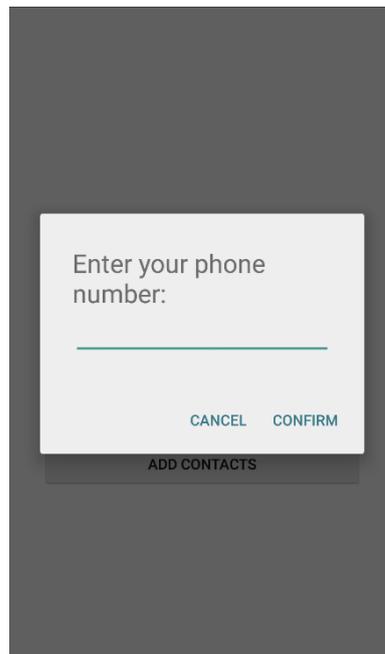


FIGURA 70: PANTALLA PARA MODIFICAR EL NÚMERO PROPIO

5.2.5. Consultar la información

También es posible consultar los datos relevantes de la aplicación, pulsando en la tecla de menú sobre el “About” (Figura 60), y desplegándose la Figura 71.

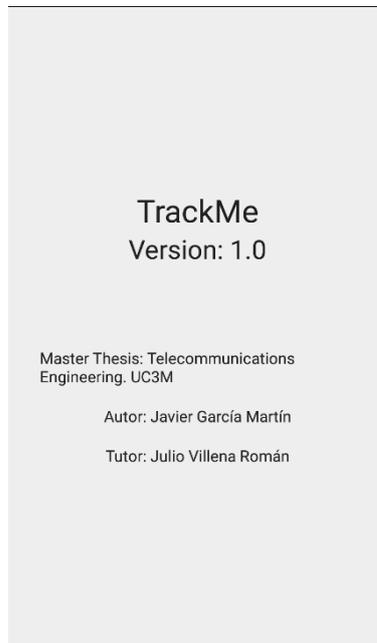


FIGURA 71: PANTALLA DE ABOUT

5.2.6. Confirmar la salida

Si se pulsa la tecla de retroceso, la aplicación emergerá una ventana para confirmar la salida, tal como se muestra en la figura 72, y así prevenir un cierre accidental de ésta.

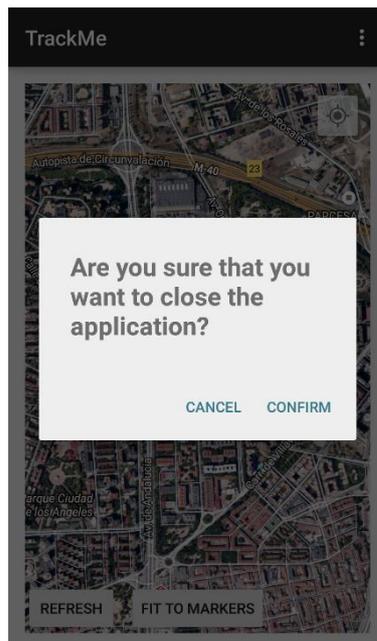


FIGURA 72: PANTALLA DE CONFIRMACIÓN DE CIERRE

6. Pruebas de funcionamiento

6.1. Introducción

Se han realizado distintas pruebas sobre las funcionalidades principales de la aplicación, que se explicarán en los siguientes apartados mediante capturas de pantallas.

6.2. Introducir número de teléfono

Al iniciar por primera vez la aplicación aparecerá en la pantalla el cuadro de diálogo de introducir el número de teléfono. Una vez realizado el procedimiento, el dato quedará almacenado en la memoria del dispositivo, quedando disponible para los usos futuros de la aplicación. Se puede ver en la Figura 73, cómo una vez introducido, al consultarlo se muestra correctamente.

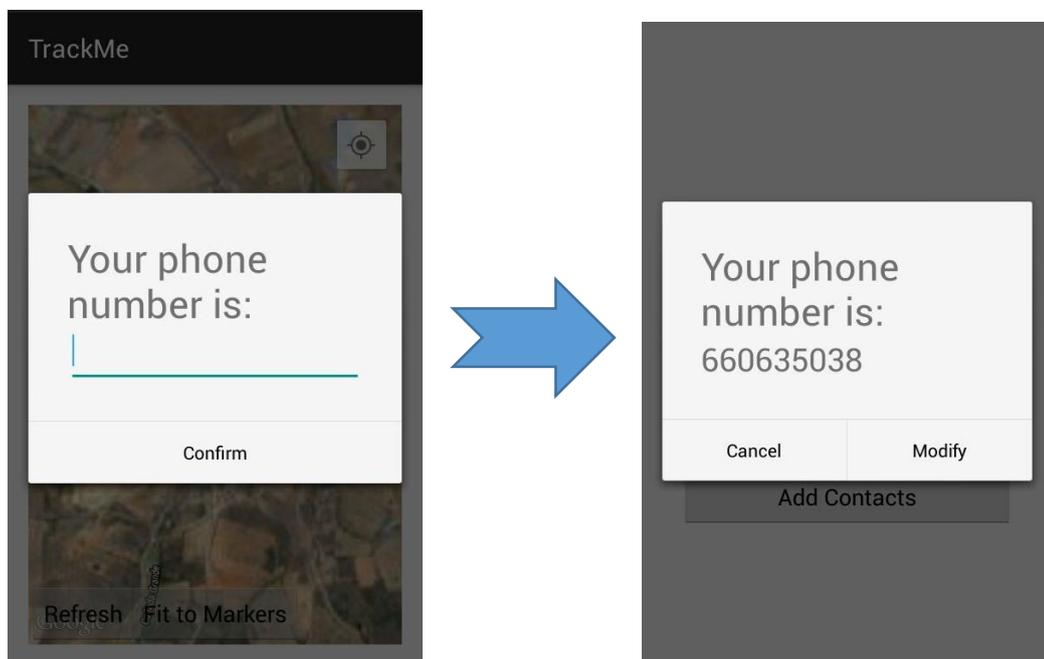


FIGURA 73: PRUEBA DE FUNCIONAMIENTO DE INTRODUCIR NÚMERO DE TELÉFONO

6.3. Seleccionar contactos

La siguiente prueba ha sido seleccionar los contactos de la lista y así ver cómo, cada vez que se añade un contacto nuevo, se marca con un tic, y al mismo tiempo se muestra con un mensaje de pantalla el número de éste, como se puede apreciar en la Figura 74.

Select contacts	Accept
FÉLIX CASA	<input type="checkbox"/>
GABRIEL	<input type="checkbox"/>
ISABEL MORENO	<input type="checkbox"/>
Isa Grego	<input type="checkbox"/>
JAVIER	<input checked="" type="checkbox"/>
JOSÉ MENCIA	<input type="checkbox"/>
JULITA LEONARD	<input type="checkbox"/>
Javi Luci	<input type="checkbox"/>
628056657	
Javi Primo	<input type="checkbox"/>

FIGURA 74: PRUEBA DE FUNCIONAMIENTO DE SELECCIONAR CONTACTOS

6.4. Añadir un contacto

A continuación, se ilustra en la Figura 75 la prueba de funcionamiento de cómo al añadir un nuevo contacto pasa a aparecer en el mapa.

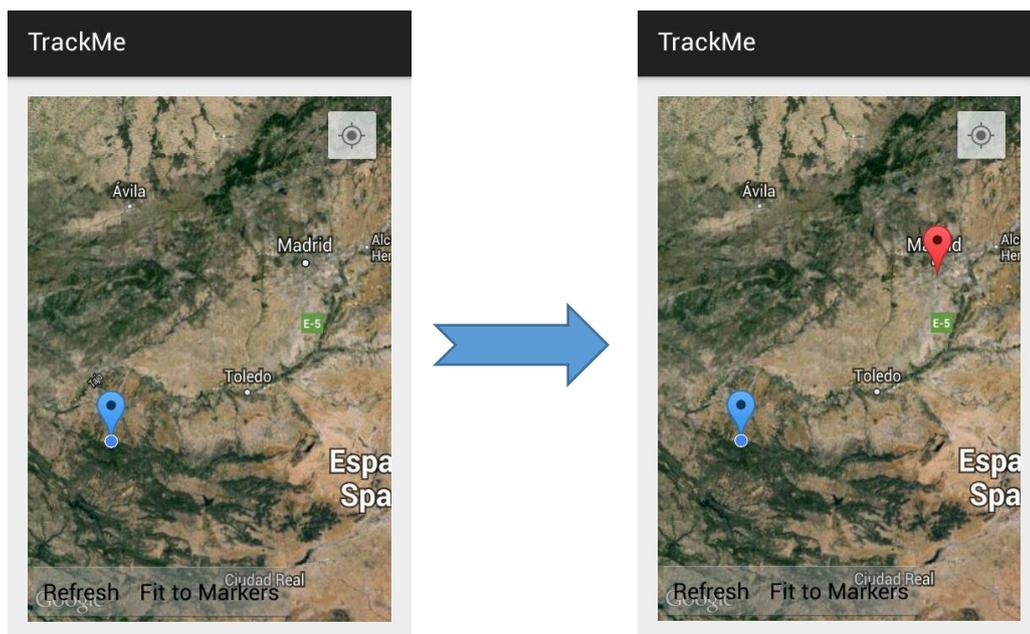


FIGURA 75: PRUEBA DE FUNCIONAMIENTO DE AÑADIR UN CONTACTO

6.5. Cambiar número teléfono

La siguiente prueba consiste en modificar el número de teléfono, que será útil en caso por ejemplo de cambiar la tarjeta SIM del dispositivo o quizás por una introducción errónea de éste al inicio de la aplicación. En la Figura 76 se puede observar cómo primeramente se consulta el número actual, a continuación se modifica éste introduciendo el nuevo, y una vez confirmado aparece una mensaje por pantalla confirmando que lo incluye. Por último al consultar el número aparece ya el modificado.



FIGURA 76: PRUEBA DE FUNCIONAMIENTO DE MODIFICAR EL NÚMERO DE TELÉFONO

6.6. Refresh y Fit to markers

Aquí se ha probado el efecto que tienen los botones añadidos en la parte inferior de la pantalla.

El primero de ellos, el “Refresh”, actualiza los marcadores inmediatamente. En la Figura 77 se puede observar cómo al pulsarlo, uno de los marcadores cambia de color, mostrando que se conectó hace más de una hora, como se explicará más adelante.

El botón de “Fit to Markers” por su parte, ajusta el mapa a los marcadores y la posición propia, por lo que se puede visualizar un aumento del zoom de la pantalla que los encaja de una manera más eficiente.

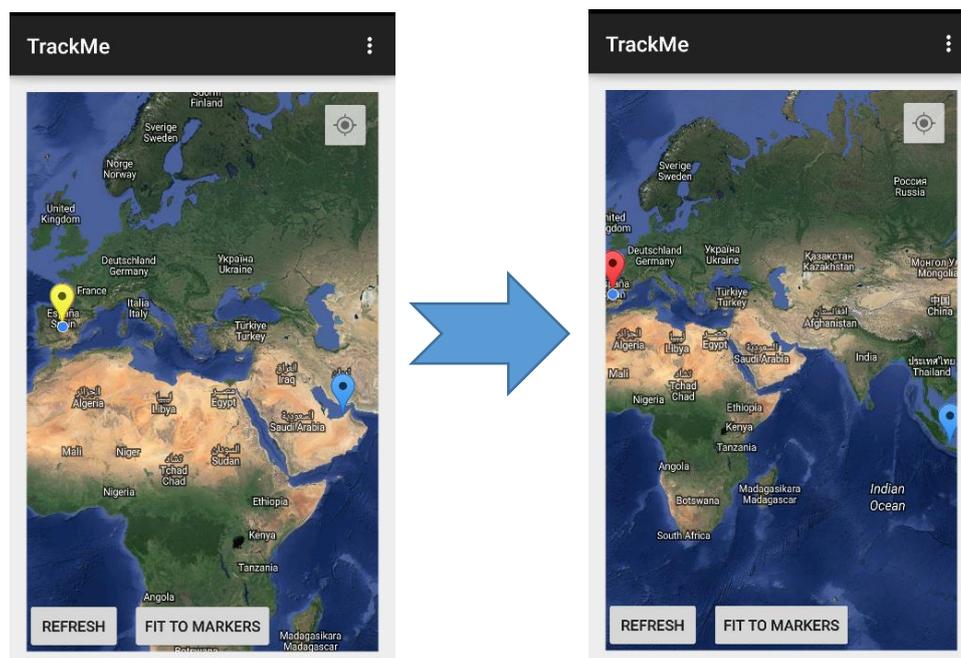


FIGURA 77: PRUEBA DE FUNCIONAMIENTO DE “REFRESH” Y “FIT TO MARKERS”

6.7. Marcadores distinto color por última conexión

6.7.1. Introducción

Una funcionalidad importante de esta aplicación es que se puede distinguir visualmente la actividad de los contactos. El color del marcador podrá diferenciar si el contacto se recibió información de la posición en la última hora (azul), el último día (amarillo), o hace más de un día (rojo).

6.7.2. Azul

En la Figura 78 se observa cómo al pulsar sobre el marcador se puede ver la información del compañero, mostrando el nombre, número de teléfono y última actualización.

En este caso, el contacto se conectó dos minutos antes de la hora actual y por ello el marcador tiene el color azul.

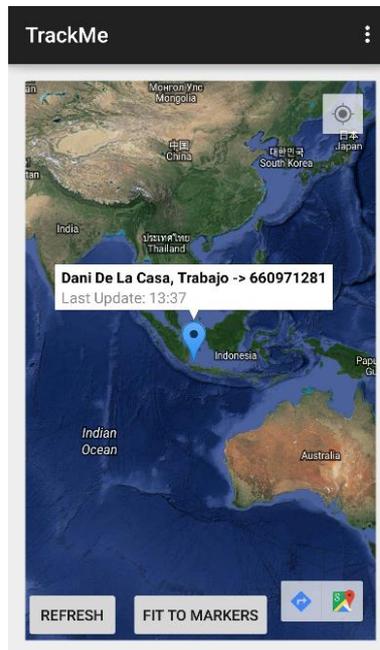


FIGURA 78: PRUEBA DE FUNCIONAMIENTO DE ÚLTIMA CONEXIÓN MENOR DE UNA HORA (MARCADOR AZUL)

6.7.3. Amarillo

En el caso de que el contacto se haya conectado el día actual y hace más de una hora, el marcador se mostrará en amarillo. En la captura de la Figura 79, se ve cómo el contacto se conectó dentro de este rango y por tanto el marcador se muestra consecuentemente.

6.7.4. Rojo

Aquí se puede ver cómo al pulsar sobre un contacto en la Figura 80, se muestra que se conectó hace más de un día con la fecha y la hora de esta última conexión.

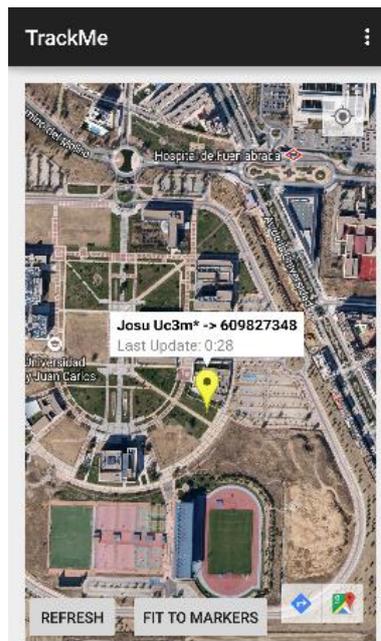


FIGURA 79: PRUEBA DE FUNCIONAMIENTO DE ÚLTIMA CONEXIÓN MAYOR DE UNA HORA EN EL MISMO DÍA (MARCADOR AMARILLO)



FIGURA 80: PRUEBA DE FUNCIONAMIENTO DE ÚLTIMA CONEXIÓN MAYOR DE UN DÍA (MARCADOR ROJO)

6.8. Indicaciones de Google Maps

También se ha incorporado, como se ha mencionado en apartados anteriores, la posibilidad de acceder a una ruta al compañero, calculada a través de Google Maps.

Como se puede observar en la Figura 81, al pulsar sobre un marcador, aparecen dos botones en la parte inferior derecha de la pantalla. El de más a la derecha de los dos (que contiene el símbolo de Google Maps), llevará al usuario a la aplicación de los mapas, mostrándole la posición y pudiendo interactuar con ella a partir de aquí. El restante caracterizado por una flecha sobre fondo azul, creará una ruta directa desde la posición del usuario a la del marcador, a través de la aplicación de Google Maps, llevando a la pantalla de la derecha en la Figura 81.

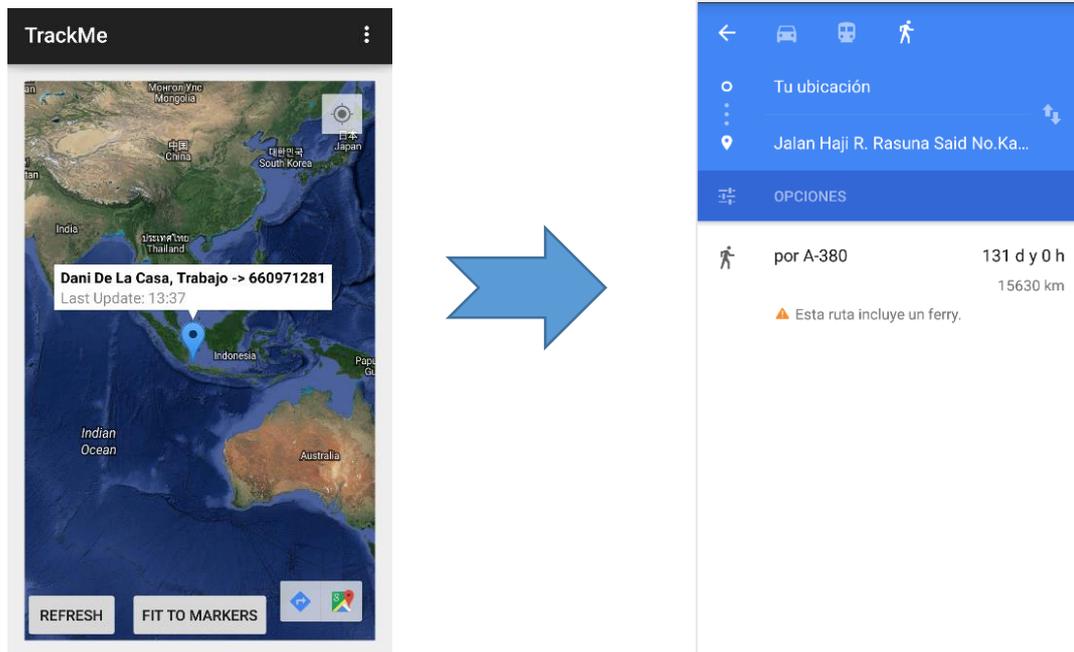


FIGURA 81: PRUEBA DE FUNCIONAMIENTO DE LAS INDICACIONES DE GOOGLE MAPS

7. Historia del proyecto y presupuesto

7.1. Introducción

En este apartado, se explicará el desarrollo del proyecto a lo largo del tiempo y el coste que éste ha tenido en recursos. Para la realización de éste se han tenido en cuenta los perfiles de:

- Ingeniero Senior, con el rol de tutor. Experto en el ámbito de las telecomunicaciones, responsable de supervisar el proyecto así como de la documentación asociada.
- Ingeniero Junior, con el rol de alumno. Persona con perfil técnico con conocimientos de Android y Java, que se encarga del desarrollo e implementación del proyecto, una vez realizado el estado del arte y estudio previo. También se encargará de realizar una batería de pruebas que demuestren el funcionamiento y proponer una serie de implementaciones futuras a partir del proceso planteado.

7.2. Planificación

7.2.1. Introducción

El proyecto está dividido en distintas fases que se explicarán en el siguiente apartado, y serán divididas en tareas en el Diagrama de Gantt.

7.2.2. Fases

Las fases de las que consta el proyecto serán las siguientes:

- 1) Adquisición de conocimiento. Se basó en la recopilación de información sobre el contexto del proyecto a desarrollar, para su posterior estudio.
- 2) Análisis. Aquí se procedió a examinar el material recopilado, para sintetizarlo y plasmar las mejores alternativas, tanto físicas como lógicas de implementación de la aplicación.
- 3) Implementación, codificación y pruebas. Esta fase contiene la implementación de la aplicación Android, en base al análisis previo. Al mismo tiempo, se van realizando pruebas de funcionamiento, para ir viendo tanto que se cumplen los requisitos, como para ir perfeccionando el funcionamiento de ésta y así solventar posibles errores.
- 4) Documentación. Una vez comprobados el funcionamiento, se procede a plasmar el conjunto del proyecto documentadamente y a su defensa.

7.2.3. Diagrama de Gantt

El siguiente diagrama de Gantt, representado en la Figura 82, muestra la planificación que se ha seguido en la realización del proyecto, desglosando el tiempo que ha ocupado la realización de cada fase con sus correspondientes tareas asociadas.

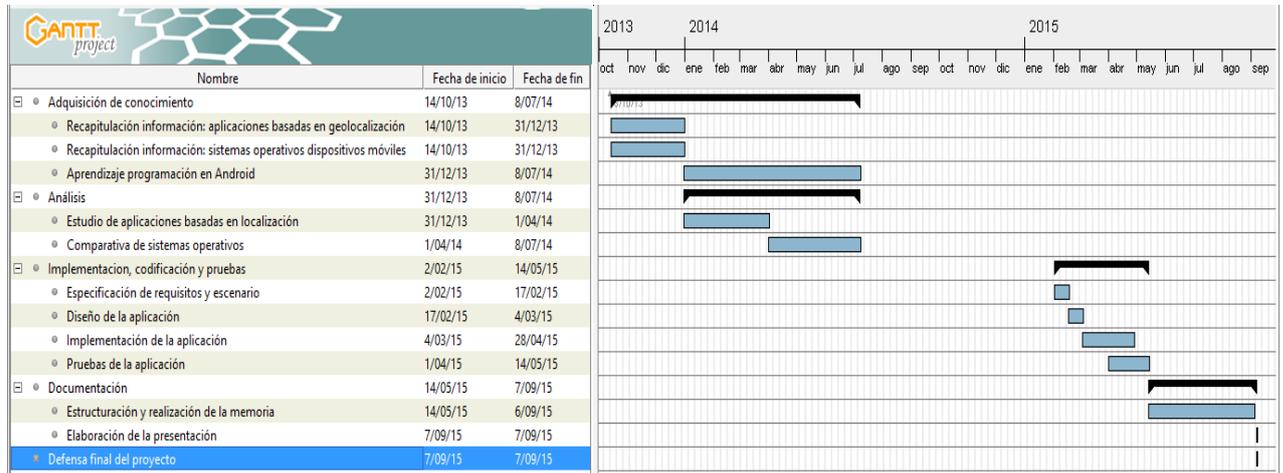


FIGURA 82: DIAGRAMA DE GANTT

7.3. Presupuesto

7.3.1. Introducción

Finalmente, se hará un resumen de las horas empleadas en el proyecto y del presupuesto aproximado que éste ha supuesto.

7.3.2. Horas totales

En la Figura 83 se desglosan las horas dedicadas a cada fase del proyecto y al final el número total de horas dedicadas.

Fase	Horas dedicadas
Adquisición de conocimiento	540
Análisis	360
Implementación, codificación y pruebas	210
Documentación	240
Total de horas	1350

FIGURA 83: HORAS EMPLEADAS EN EL PROYECTO

7.3.3. Costes

A continuación se explican los costes que se han producido al realizar el proyecto.

El coste del personal implicado en el desarrollo se refleja a continuación en la Figura 84, seguido de la de los equipos en la 85 y finalmente el total en la 86, en el que se añaden otros costes indirectos como pueden ser desplazamientos, material de oficina, etc.

Trabajador	Perfil	Horas	Coste/hora	Coste total
García Martín, Javier	Ingeniero Jr.	1350	22.5 €	30375 €
Villena Román, Julio	Ingeniero Sr.	20	45 €	900 €
			Suma Total	31275 €

FIGURA 84: COSTE RELATIVO A LOS PARTICIPANTES

Equipos	Coste	% dedicación	Total
PC HP Envy	1000 €	80	800 €
2 dispositivos móviles	900 €	50	450 €
		Suma Total	1250 €

FIGURA 85: COSTE RELATIVO A LOS EQUIPOS

Coste	Total
Participantes	31275 €
Equipos	1250 €
Indirectos	200 €
Suma Total	32752 €

FIGURA 86: COSTE TOTAL DEL PROYECTO

8. Conclusiones y trabajos futuros

8.1. Introducción

Tras presentar y describir las diferentes técnicas basadas en localización, se ha llegado a la conclusión de que su impacto tanto social, como económico son de gran calibre. El hecho de la necesidad continua del ser humano de comunicarse y expresar sus sentimientos, junto con el de compartir dónde está físicamente en cualquier momento (ubicación), han dado lugar a un gran auge y generalización de las redes sociales basadas en geolocalización y a la aparición de redes Geo-crowdsourcing y redes de aceras GPS. De esta manera, se ha realizado un trabajo paralelo por parte de las empresas para recoger estos datos dejados por los usuarios. Así, han visto la oportunidad de mejorar su modelo de negocio, recomendando de forma personalizada a éstos según sus gustos.

No sólo son ventajas en estas tecnologías; también existen vulnerabilidades que hay que solventar, como son problemas de seguridad y privacidad. La cantidad de datos almacenados sobre cada usuario es cada vez mayor, y el acceso a ellos de forma no autorizada y su uso no consentido son aspectos a tener en cuenta. Por ello técnicas de ofuscación de la información, seudonimización, anonimación, control de acceso y autenticación, toman una gran notabilidad.

La principal baza de aplicar estas técnicas a nuestra vida cotidiana son los terminales móviles. Éstos reúnen principalmente tres tecnologías para la geolocalización: redes celulares, GPS y Wifi. Mediante su uso independiente o la conjunción entre varias de ellas, se puede establecer una posición bastante precisa del usuario.

En cuanto a los sistemas operativos, Android toma una clara ventaja sobre el resto [26]. Su clara tendencia creciente en el mercado está acompañada de otras características fundamentales, como son el hecho de ser una plataforma abierta sobre la que desarrollar nuevas aplicaciones, el ser adaptable a casi cualquier dispositivo y tener una buena seguridad. Por tanto, Android es la mejor opción, si se desea penetrar en la mayor cuota de usuarios posibles.

Se ha implementado el desarrollo de la aplicación en Android, cumpliendo los objetivos que se esperaban de ella. Es capaz de localizar en todo momento a un grupo de usuarios seleccionados dentro de la lista de contactos y mostrarlos en un mapa, siendo la posición propia al mismo tiempo visible para ellos. También han surgido asuntos de implementación relativos a la privacidad y otros aspectos que serán tratados después, al igual que posibles adaptaciones a situaciones actuales.

8.2. Privacidad

Durante el desarrollo de la memoria se han mencionado temas que pueden preocupar a la privacidad del usuario, por lo que aquí se explicarán con un poco más de detalle.

Primeramente es importante tratar el tema de suplantación de identidad. Aquí simplemente se pide el número propio al iniciar la aplicación y se le permite al usuario modificarlo posteriormente, debido a la posibilidad de que éste se haya introducido incorrectamente o que el cliente haya cambiado de número. Al tratarse de una primera versión no se profundiza mucho en este aspecto, ya que se confía en que el primer usuario

utilizará la aplicación para gozar de sus características y no para crear problemas de confidencialidad. Si se decide crear una versión futura más segura, será necesario crear un servicio de verificación del número por SMS, ya que en multitud de dispositivos no es posible acceder a éste internamente, y además mediante esta confirmación se conseguirá asegurar que el número se corresponde al proporcionado por el usuario.

Un aspecto relacionado es el envío de la información de la web en claro. Al no ser una https, alguien puede recoger la información por la red de la web e intentar suplantar la identidad, por lo que se podría crear un problema. Por tanto habría que trabajar junto con la parte del servidor, para crear un sistema más seguro de petición y respuesta, que podría consistir en enviar el número de teléfono del cliente, junto con una clave cifrada que hagan prácticamente imposible una réplica por parte de un tercero.

Otro tema concerniente a este campo es el de cómo saber si hay usuarios que pueden conocer la posición propia sin darles consentimiento. En realidad el funcionamiento consiste en que, al elegir una lista de compañeros que se quieren visualizar, al mismo tiempo se está diciendo que se le da permiso a éstos para que puedan conocer mi posición, por lo que al recibir las localizaciones, sólo se hará de las que se tenga permiso.

8.3. Trabajos Futuros

8.3.1. Mejoras

Una vez desarrollada la aplicación, mediante la realización de las pruebas de funcionamiento, se han detectado algunos detalles de la aplicación, que son susceptibles de ser perfeccionados en versiones futuras. Primeramente, la aplicación hace saber al usuario al principio de ésta que es necesario tener los servicios de ubicación activados para que ésta funcione correctamente, pero en realidad no se controla que sean desconectados durante su uso, por lo que se presupone que el usuario será responsable de que se mantengan activados. A pesar de todo, la desconexión de éstos no supondrá un problema grave, ya que la aplicación seguirá funcionando, con la salvedad de que el cliente dejará de enviar su posición periódicamente.

Durante el uso de ésta también se ha revelado que es necesaria una conexión a Internet estable por parte del dispositivo. En caso contrario, al estar en constante contacto con Google Maps y con el servidor, se puede bloquear realizando peticiones, pero eso sí, sin llegar a detenerse.

También cabe reseñar que desde la aplicación del cliente, no se tiene control sobre posibles pérdidas de conexión por parte del servidor, por lo que el sistema podría dejar de funcionar en estos casos. Esto es un error que habría que tratarlo desde el punto de vista de este último para garantizar la conexión continua, mediante algún tipo de respaldo.

Otro detalle de implementación que se ha descubierto es la posibilidad de que pueda existir multiplicidad del mismo contacto en el dispositivo, asociando el mismo número a distintas cuentas del mismo usuario. Por ejemplo el tener almacenado en el teléfono un contacto con un número, el mismo contacto pero en su versión de cuenta de correo electrónico con el mismo número, al mismo tiempo mismo contacto con mismo número pero asociado a la cuenta de alguna

red social... dando todo esto lugar a que pueda aparecer en la lista de contactos de la aplicación el contacto duplicado. Esto no supone ningún problema de funcionamiento porque al seleccionar cualquiera de ellos, sería añadido el número de teléfono asociado. En una versión posterior de la app, se podría tratar este detalle para mejorarla visualmente.

8.3.2. Adaptación en dominios concretos

La aplicación desarrollada es una buena base para aplicarla en distintos trabajos futuros sobre los que ampliarla y mejorarla dependiendo de las distintas funcionalidades requeridas. Seguidamente se explican algunos ejemplos posibles, que no excluyen cualquier tipo de implementación que se ajuste a las características de la app:

- Centro de discapacitados o residencia de ancianos. Mediante un dispositivo asociado a cada una de las personas con problemas y necesidad de control de que no salgan de una determinada distancia, se podría crear una alarma de proximidad personalizada que notificara estos eventos.
- Control de presos. En la misma línea que el caso anterior.
- Excursiones de niños. Sería de gran utilidad para los responsables de un grupo de menores de edad saber en todo momento dónde se encuentran éstos y así eliminar las posibilidades de que se pierda alguno. Se podría además añadir alarmas de proximidad.
- Órdenes de alejamiento. En sintonía con lo que actualmente está en funcionamiento, se podrían delimitar zonas de exclusión e inclusión para personas que deban permanecer fuera o dentro de un perímetro por orden judicial.
- Logística, vehículos reparto. Una empresa de este tipo podría incorporar la aplicación en sus vehículos para ver el estado del desplazamiento y mejorar así en productividad.
- Carrera deportiva, ciclismo. En una competición de velocidad se podría tener controlado a un equipo de corredores o a la totalidad de ellos para ver posiciones o si incumplen las reglas.

Bibliografía

- [1] @shanesnow, «Web de Google Sites,» Mashable, [En línea]. Available: <https://sites.google.com/a/student.ie.edu/sistgeolocalizacion/historia>.
- [2] A. A.-D. Thamer Abulleif, «Web saudigis.org,» [En línea]. Available: http://www.saudigis.org/FCKFiles/File/SaudiGISArchive/3rdGIS/Papers/79_E_ThamerAbulleif_KSA.pdf.
- [3] D. G. Álvarez, «Web Universidad Autónoma de Madrid,» [En línea]. Available: <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20080125DavidGarcia.pdf>.
- [4] H. L. E. W. Bernhard Hofmann-Wellenhof, GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more, Springer, 2007.
- [5] S. Dawoud, «Web del Department of Telecommunication Systems, Alemania,» [En línea]. Available: http://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1112/snet-project/gnss-principles-and-comparison_dawoud.pdf.
- [6] H. A. Karimi, Advanced Location-Based Technologies and Services, CRC Press, 2013.
- [7] «Web de National Wildfire Coordinating Group del Gobierno de EE.UU.,» [En línea]. Available: http://www.nwcg.gov/pms/pubs/475/PMS475_chap5.pdf.
- [8] J. T. H. Pedro Andrade-Sanchez, «Web de Coperative Extension, Universidad de Arizona,» [En línea]. Available: <http://extension.arizona.edu/sites/extension.arizona.edu/files/pubs/az1553.pdf>.
- [9] J. R. C. Corredera, «Web de Universidad Politécnica de Madrid,» Enero 2005. [En línea]. Available: http://www.upm.es/sfs/Rectorado/Organos%20de%20Gobierno/Consejo%20Social/Actividades/tecnologias_servicios_para_sociedad_informacion.pdf.
- [10] E. R. Benito, «Web de la Universidad de Salamanca,» [En línea]. Available: <http://campus.usal.es/~comunicacion3punto0/comunicaciones/042.pdf>.
- [11] E. Hermogenes, «Web examiner.com,» San Diego Spiritual Enlightenment, 2012. [En línea]. Available: <http://www.examiner.com/spiritual-enlightenment-in-san-diego/eva-hermogenes>.
- [12] P. Albarracín, «Web de América Economía,» Social Media, 8 Enero 2014. [En línea]. Available: <http://tecno.americaeconomia.com/articulos/geolocalizacion-para-el-negocio-la-nueva-frontera-de-la-productividad>.
- [13] «Curso de Especialización en el desarrollo de aplicaciones para Android,» Universidad Politécnica de Valencia, [En línea]. Available: <http://www.androidcurso.com/>.

- [14] «Android Developers official web,» [En línea]. Available: <http://developer.android.com/index.html>.
- [15] «Universidad Nacional Abierta y a Distancia de Colombia,» [En línea]. Available: http://datateca.unad.edu.co/contenidos/233016/EXE_SAM/leccin_13_desarrollo_de_aplicaciones.html.
- [16] «Web oficial de Apple,» [En línea]. Available: <http://www.apple.com/es/ios/>.
- [17] D. Nations, «Web de About tech,» [En línea]. Available: http://ipad.about.com/od/ipad_details/ss/iOS-Version-History-Features.htm.
- [18] «Web oficial de Windows Phone,» [En línea]. Available: <https://dev.windowsphone.com/EN-US>.
- [19] kpwalkin, «Web slideshare.net,» 26 Febrero 2014. [En línea]. Available: <http://www.slideshare.net/kpwalkin/tabla-comparativa-de-sistemas-operativos-mviles>.
- [20] «Web del diario ABC de España,» 13 Febrero 2014. [En línea]. Available: <http://www.abc.es/tecnologia/moviles-telefonía/20140212/abci-android-consolidan-poder-2013-201402121927.html>.
- [21] L. Mikusiak, «Web passion4tech.com,» [En línea]. Available: <http://www.passion4teq.com/articles/ios-android-development-comparison-1/>.
- [22] E. PROTALINSKI, «The Next Web,» 12 Noviembre 2013. [En línea]. Available: <http://thenextweb.com/insider/2013/11/12/idc-android-hit-81-0-smartphone-share-q3-2013-ios-fell-12-9-windows-phone-took-3-6-blackberry-1-7/#!vif2u>.
- [23] K. Hands, «Web Whatphone,» [En línea]. Available: <http://whatphone.com.au/guide/android-vs-ios/>.
- [24] L. M. P. J. J. C. V. Juan Carlos Albarracín, «Web de la Universidad Distrital Francisco José de Caldas, Colombia,» 10 Octubre 2013. [En línea]. Available: <http://revistas.udistrital.edu.co/ojs/index.php/tia/article/view/4312/6875>.
- [25] C. F. S. S. KARLA JOHANNA CARVAJAL VALDIVIESO, «Escuela Politécnica Superior de Chimborazo,» [En línea]. Available: <http://dspace.esPOCH.edu.ec/handle/123456789/2552>.
- [26] «Web Diffeen de información,» [En línea]. Available: http://www.diffeen.com/difference/Android_vs_iOS.