UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA, CARTOGRÁFICA Y TOPOGRÁFICA

GRADO EN INGENIERÍA GEOMÁTICA Y TOPOGRAFÍA

PROGRAMACIÓN CON DISPOSITIVOS MÓVILES. APLICACIONES A LA GEOLOCALIZACIÓN.

TRABAJO FINAL DE GRADO

Autor: Francisco Mariño Ruiz

Tutor: Israel Quintanilla García

JULIO, 2014

TABLA DE CONTENIDO:

ÍNDICE DE IMÁGENES:

ÍNDICE DE TABLAS:

1. Introducción al proyecto.

1.1. Introducción.

En los últimos años los dispositivos móviles han pasado a ser una parte esencial de nuestra vida. Las grandes compañías han apostado fuerte en el desarrollo de estos dispositivos y esto ha significado un aumento exponencial de sus características técnicas en los últimos años. Los denominados Smartphone ya no son solo aparatos de comunicación, sino que son pequeños ordenadores de bolsillo que nos acompañan a todas partes. Lo mismo pasa con las tabletas, que aunque no de manera tan arrolladora como los Smartphone, han entrado en el mercado con la intención de desplazar a los ordenadores portátiles.

Entre todas las características que se podrían destacar de estos dispositivos, cabe destacar dos, por un lado, la disponibilidad de una conexión a internet permanente, que permite estar conectado con el resto del mundo en todo momento, haciendo accesibles a los usuarios infinidad de datos y servicios. Por otro lado están las posibilidades de geolocalización que ofrecen estos dispositivos, las cuales unidas a las rápidas conexiones a internet despliegan un gran abanico de posibilidades para usuarios y programadores.

Por todo esto se puede considerar a este tipo de dispositivos como una herramienta más de trabajo y con esa idea se va a desarrollar este proyecto. La idea principal del mismo es convertir un dispositivo móvil en una herramienta funcional para el trabajo con datos espaciales, para ello se pretende diseñar una aplicación que sea capaz de acceder al sensor GNSS del dispositivo móvil y manejarlo para que el usuario pueda controlarlo y convertirlo en un aparato de captura masiva de puntos. La aplicación que se propone tendrá como objetivo la realización de inventarios rápidos en entornos urbanos, estimaciones de superficies perimetrales previas durante la planificación de un trabajo topográfico y la lectura y almacenamiento de otros puntos que el usuario pueda considerar de utilidad.

Todo esto se desarrollará en un dispositivo móvil con Android integrado, concretamente para el desarrollo se usará un Smartphone “Samsung Galaxy S4” con la versión 4.4.2 del sistema operativo Android. El objetivo final es desarrollar la aplicación que se denominará “GTopDB”, la cual será una aplicación capacitada para cubrir todos los objetivos marcados de la manera más sencilla posible para el usuario final de la misma.

La idea de agilizar la toma de datos para inventarios surgió el pasado verano de 2013, durante el cual participé en los trabajos de medición e inventariado de una zona que había sido urbanizada recientemente, en esta zona existían conflictos entre las mediciones realizadas por parte de la empresa ejecutora del proyecto y la entidad contratante. Si bien para algunas de las entidades a medir no era posible acelerar el proceso con este método, había otros elementos puntuales de mobiliario urbano como arquetas, papeleras, farolas, señales de tráfico con los que si se podría hacer, ya que no se requería su posición espacial exacta sino que solo se necesitaba su conteo. Basándome en esa idea se me ocurrió que una buena forma de hacer rápido esa parte del trabajo podría ser usar el sensor GNNS del móvil, ya que podría obtener el conteo rápido de

estos elementos y además conseguir una posición espacial que no sería excesivamente mala. Desde ese momento investigué un poco sobre las técnicas de inventariado y resultó que no iba tan desencaminado, ya que existen aplicaciones para realizar estos trabajos, pero que generalmente están más enfocadas a uso en dispositivos móviles profesionales. Así que meses después tras realizar varios cursos de programación tanto en Java como en Android e introducirme en el mundo de los dispositivos móviles, me dispongo a realizar una aplicación que cubra estos aspectos y de paso que cubra algún aspecto más que pueda ser interesante para el desarrollo de trabajos de campo.

1.2. Objetivos del proyecto.

Como se ha comentado en la introducción, el objetivo de este proyecto es diseñar una aplicación para Android capaz de manejar el sensor GNSS del móvil para capturar las coordenadas que el usuario estime necesarias. A lo largo de este proyecto, el objetivo principal es realizar el diseño de la aplicación obteniendo una versión funcional, pero además se pretenden cubrir una serie de objetivos que se detallan a continuación:

- **Estudiar las distintas posibilidades existentes en el mercado en cuanto a dispositivos móviles.** Para ello se hará un breve estudio de los dispositivos móviles existentes y de sus capacidades, de los sistemas operativos que los hacen funcionar y las ventajas que proporcionan a la hora de enfocar el desarrollo hacia unos u otros.
- **Estudiar el estado de las distintas tecnologías que intervienen en la creación de la aplicación.** Además de estudiar los dispositivos se hablará de su funcionamiento, se intentará dar a conocer como es cada una de estas tecnologías, como funcionan en el entorno de la geolocalización, cuales son las opciones que ofrecen y decidir cuál de estas opciones es la idónea para obtener la mejor localización posible y para cubrir los objetivos del proyecto.
- **Diseño de la aplicación.** En este apartado se intentará plasmar el esquema funcional de la aplicación. Se definirá la estructura interna, la estructura de capas de la aplicación, los objetos necesarios para su funcionamiento, las relaciones entre los objetos de las distintas capas de programación y el diseño de la base de datos donde se almacenarán todos los datos que la aplicación irá generando durante su funcionamiento.
- **Programación en Android.** Se hará un pequeño análisis, a modo de introducción, de la programación en Android y el entorno de desarrollo proporcionado para esta plataforma.
- **Desarrollo de la aplicación.** Con todos los datos previos sobre el funcionamiento del entorno de desarrollo que se han recogido en los apartados anteriores se hará una revisión detallada del diseño y funcionamiento interno de la aplicación.
- **Fase de pruebas.** Una vez terminada la fase de desarrollo de la aplicación, se procederá a realizar una fase de pruebas funcionales. En estas pruebas se buscaran posibles fallos de funcionamiento, se medirán los tiempos de inicialización y

lectura de los sensores GNSS para su uso en la aplicación y finalmente se harán pruebas de lectura de puntos de coordenadas conocidas para analizar las coordenadas que es capaz de ofrecer el dispositivo móvil utilizado y la veracidad de las precisiones que el fabricante promete en las especificaciones técnicas del producto.

2. Estado del arte.

2.1. Dispositivos móviles.

Atendiendo a la definición expuesta en el libro "Mobile and Wireless Design Essentials"[1] se puede definir como dispositivo móvil todo aquel dispositivo que puede ser usado en movimiento, abarcando desde los ordenadores portátiles hasta los teléfonos móviles, es decir, todo dispositivo que no tenga una localización fijada para su funcionamiento. Sin embargo han de diferenciarse de esta definición todos aquellos dispositivos denominados inalámbricos, que si bien en su gran mayoría entran el concepto de dispositivo móviles, no todos los dispositivos inalámbricos son dispositivos móviles.

Cabe recordar que este tipo de dispositivos tienen una serie de limitaciones que vienen impuestas por su tamaño y su peso. Hace unos años se podían encontrar las siguientes limitaciones:

- Capacidad de memoria.
- CPU de bajo rendimiento.
- Pantallas pequeñas.
- Duración de las baterías.
- Mecanismos de entrada de datos limitados.
- Excesiva heterogeneidad de sistemas operativos.
- No existía soporte a aplicaciones de terceros.
- Ancho de banda limitado.
- Deficiente conexión a internet.

Con el paso de los años estos problemas se han ido solucionando y a día de hoy se puede señalar como principal problema de estos dispositivos la duración de las baterías.

En el libro anteriormente citado[1] se propone una clasificación de estos dispositivos móviles, que es a la que se hará referencia en este trabajo.

Teléfonos móviles: Creados en EE.UU a partir de radios portátiles en los años 70, tardaron en aparecer en el mercado debido a su alto costo. Comenzaron a comercializarse a gran escala en los 80 y en los 90 hicieron su gran estallido en el mercado mundial. Hace unos años dominaban el mercado, eran dispositivos diseñados principalmente para comunicación por voz y mensajes de texto. En los últimos modelos se introdujo la capacidad de conectarse a internet para realizar acciones básicas: mensajería WAP, correo electrónico y navegación por internet a bajas velocidades.

PDA's: Acrónimo de "Personal Digital Assistant", se puede definir como un ordenador de mano dotado de características básicas (agenda, editor de texto, grabadora, etc.) para el usuario móvil. Gracias a su poder de computación fue creciendo su capacidad para ejecutar programas más complejos.

Smartphone: Terminan con la batalla creada entre los móviles y las PDA's, rompen el concepto tradicional de teléfono móvil, consiguiendo un aparato a medio camino entre la PDA y el teléfono móvil. El rápido desarrollo de la tecnología hace que desparezcan las limitaciones explicadas anteriormente (aumentan las capacidades de proceso, la cantidad de memoria, el tamaño y calidad de las pantallas, la conexión a internet, etc.). Además con el interés que han suscitado estos dispositivos en la sociedad las grandes compañías de software se han volcado en el diseño de sistemas operativos que acaban con otro de los grandes problemas del mercado móvil, el hecho de que cada fabricante tuviese su propio sistema operativo cerrado, consiguiendo a su vez que por primera vez puedan introducirse aplicaciones de terceros en los teléfonos móviles.

Handhelds o PDA's profesionales: Van un paso más allá que las PDA's tradicionales, se convierten en auténticos ordenadores de mano capaces de mover sistemas operativos robustos como "Windows Mobile". Este tipo de dispositivos son muy utilizados en entornos de trabajo para la captura masiva de datos, están diseñados para adaptarse a entornos de trabajo hostiles y en función del modelo poseen diversas ranuras de expansión (lectores de códigos de barras, de tarjetas de crédito, impresoras, módems, antenas GNSS, etc.). Son muy utilizados en el ámbito de la geomática para manejo de receptores GNSS o estaciones totales, así como para la captura de datos en caso de que incorporen sensor GNSS.

Tabletas: se pueden considerar como la evolución de los Smartphone, de hecho se consideran como el paso intermedio entre los Smartphone y los ordenadores portátiles. Son dispositivos con pantallas más grandes que facilitan el trabajo y la entrada de datos, prácticamente tienen las mismas características que los Smartphone aunque sobre el papel mejoran la durabilidad de las baterías para poder trabajar durante más horas.

Ordenadores portátiles: A día de hoy siguen siendo los más capaces de ejecutar tareas complejas, aunque en teoría muchos otros dispositivos móviles han alcanzado en números a estos dispositivos, por su arquitectura son los únicos capaces de correr los sistemas operativos más completos y por tanto están capacitados para ejecutar aplicaciones más complejas.

Para ver de modo más gráfico el estado de estos dispositivos, se ha realizado una tabla comparativa en la que se exponen las características de algún modelo perteneciente a las categorías citadas anteriormente:

	Dispositivo (Modelo comparado)	Sistema operativo	Procesador	Potencia gráfica	Memoria RAM	Memoria interna	Pantalla	Batería	Tamaño Peso
	Teléfono móvil (Nokia 8310)	Propietario	-	-	-	Solo contactos	Bicolor 84x84 pixeles	350 h espera 4 h llamada	97x43x19 mm 84 g
	PDA (Acer N10)	Windows Pocket PC 2003	300 Mhz	-	64 MB	32 MB	65536 colores 240x320 px 3.5 pulgadas	10 horas	130x78x16.9 mm 168 g
	Smartphone (Samsung Galaxy S4)	Android 4.2	1.9 Ghz	Adreno 320 80 millones de triángulos	2 GB	Hasta 64 GB	16M de colores 1080 x 1920 px 5 pulgadas	350 h espera 17 h llamada	136.6x69.8x7.9 mm 130 g
	Handheld (Leica Zeno 15)	Windows CE 6.0	533 Mhz	-	512 MB	1 GB	640 x 480 px 3.5 pulgadas	9 horas	323x125x45 mm 900 g
	Tablet (Apple Ipad air)	iOS 7	1.3 Ghz	PowerVR G6430 128 millones de triángulos	1 GB	Hasta 128 GB	16M de colores 1536 x 2048 px 9.7 pulgadas	10 horas de navegación	240x169.5x7.5 mm 469/478 g
	Ordenador portátil (Toshiba C55-A-1NV) (Gama media)	Windows 8	2.5 Ghz	Nvidia GeForce 710M 324 millones de triángulos	4 GB	750 GB	1366x768 px 15.6 pulgadas	4.5 horas	380x242x30.8 mm 2.3 kg

Tabla 1: Comparativa de dispositivos móviles.

Se puede observar en la comparativa como tanto los Smartphone como las tabletas poseen capacidades técnicas similares a las de los ordenadores portátiles, aunque debido a la arquitectura de sus procesadores corren sistemas operativos que no están capacitados para ejecutar algunas de las aplicaciones que se utilizan en un sistema operativo de escritorio.

Después de analizar los datos técnicos y los tipos de dispositivos existentes, se puede afirmar que por características técnicas los Smartphone pueden suponer una herramienta muy útil. Para confirmarlo se va a investigar la tasa de mercado que estos aparatos han adquirido en los últimos meses. Según el informe "Radiografía del mercado móvil en España"[(2)] en septiembre de 2013, el 80% de los usuarios de telefonía móvil es poseedor de un Smartphone, lo que supone un crecimiento de un 21% con respecto al año anterior. En el siguiente gráfico se puede ver ese crecimiento general y el crecimiento por tramos de población, en el que se refleja una gran implantación en las capas más jóvenes de la sociedad:

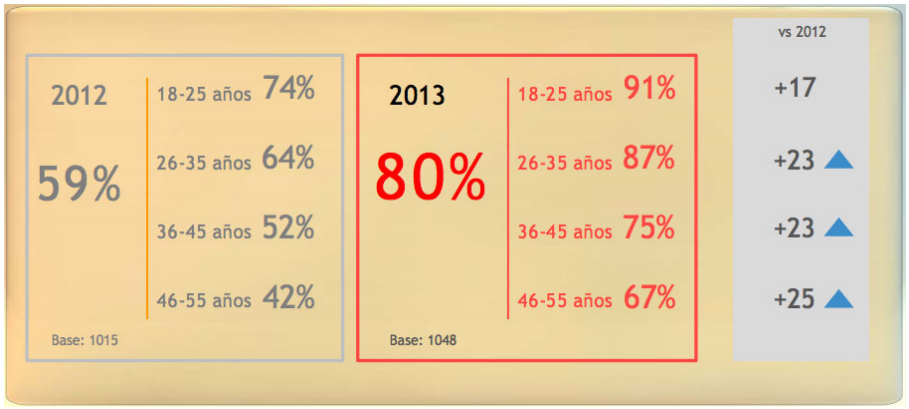


Imagen 1: Tasa de implantación de los Smartphone.

Otro dato importante que revela este informe[2] es el que habla de la conexión a internet desde los distintos dispositivos. En el siguiente gráfico se puede ver el porcentaje de conexiones diarias a internet desde diferentes dispositivos, como es evidente los ordenadores tradicionales ocupan el primer puesto, sin embargo los Smartphone ocupan el segundo lugar a muy poca distancia de los ordenadores. Se ve como los ordenadores han sufrido un pequeño descenso mientras que los Smartphone han aumentado el número de conexiones diarias un 9% con respecto al año anterior. En tercer lugar se encuentran las tabletas que se mantienen con una tasa prácticamente idéntica a la del año pasado.

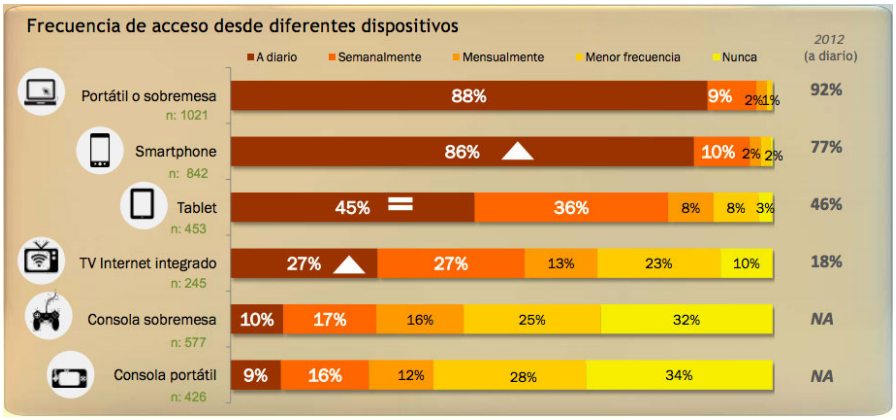


Imagen 2: Frecuencia de acceso a internet desde diferentes dispositivos.

Como último dato de este estudio[2] se puede reseñar el hecho de que un alto porcentaje de los usuarios prefiere el uso de aplicaciones para realizar las conexiones a internet o cualquier tipo de trabajo que hayan de realizar con el móvil. Según los resultados del estudio, los aspectos más valorados por los usuarios son: encuentran más fácil la navegación, el acceso al contenido es más inmediato, la navegación es más rápida, se adapta mejor a las necesidades y recuerda las preferencias de navegación. En la siguiente imagen se pueden ver los resultados de este apartado del estudio:

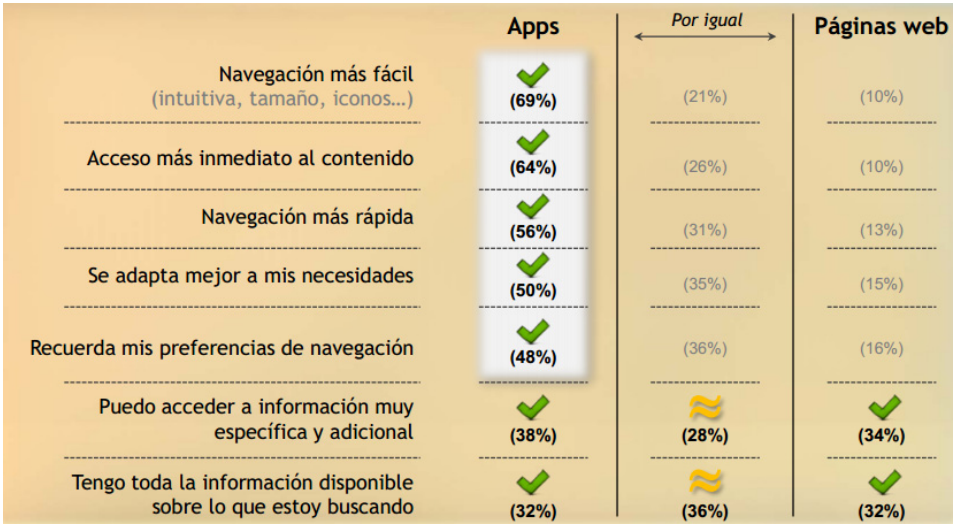


Imagen 3: Apps vs web.

Por todos estos datos que se han expuesto se puede afirmar que el Smartphone es una herramienta que un alto porcentaje de la población posee o que, atendiendo a los datos de crecimiento, se espera que posea en los próximos meses. Del mismo modo, aunque a menor escala están las tabletas, que si bien no forman una parte tan importante del mercado, poco a poco va creciendo su implantación. También se puede afirmar que a nivel de potencia de procesado son dispositivos perfectamente cualificados para realizar los trabajos que se pretenden hacer con ellos. Es por todo esto por lo que se ha decidido crear una aplicación para Smartphone, que pueda convertir un dispositivo que casi todo el mundo posee en una herramienta útil para el topógrafo. Al mismo tiempo que se diseña dicha aplicación para Smartphone se hace con la vista puesta en su uso en tabletas, puesto que por su tamaño y su configuración pueden ser una alternativa más eficaz, en términos de productividad y de durabilidad de la batería, a la hora de realizar trabajos en campo.

2.2. Sistemas operativos móviles.

Una vez se ha decidido que el Smartphone es el dispositivo para el que se va a diseñar la aplicación, falta escoger el sistema operativo sobre el que se trabajará. Para ello se va a realizar una breve introducción a los sistemas operativos para móviles existentes en el mercado.

Una primera definición de sistema operativo móvil, se puede extraer de Wikipedia donde se lee: "*Un sistema operativo móvil es el software que controla un dispositivo móvil al igual que los sistemas operativos tradicionales controlan los ordenadores de sobremesa o portátiles. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos*[(3)]", la cual viene a ser una buena aproximación a la realidad de los sistemas operativos móviles, puesto que por su diseño y la arquitectura de los componentes de hardware que se encuentran en el interior de estos dispositivos son más simples que los sistemas operativos tradicionales y tienen un enfoque totalmente diferente.

A día de hoy existen varios sistemas operativos en el mercado, cada uno con sus características y con sus consiguientes ventajas e inconvenientes. Apoyándonos en la clasificación de sistemas operativos móviles propuesta por Jesús Tomás Gironés[(4)], podemos distinguir los siguientes:

Apple iOS: Sistema operativo exclusivo para los dispositivos de Apple. Se trata de un sistema operativo que ofrece gran simplicidad de uso al usuario y que al tratarse de un sistema operativo cerrado (desarrollado a la par que el hardware y solo para ese hardware) ofrece una gran optimización que consigue un funcionamiento más liviano que otros sistemas operativos incluso con hardware menos potente que la competencia. Este sistema operativo fue lanzado en 2007 por Apple y aproximadamente una vez al año sale a la luz una versión nueva. Actualmente se comercializa la versión iOS 7 aunque la versión 8 está a punto de salir al mercado.

Android: Sistema operativo multidispositivo diseñado por un grupo de empresas encabezado por Google. Basado en los sistemas operativos abiertos de Linux, supuso la

alternativa a iOS en el momento de su salida al mercado. Se trata de un sistema abierto que utilizan múltiples fabricantes en sus dispositivos, esto hace que al querer abarcar tantos dispositivos con tantas configuraciones diferentes, no esté tan optimizado para el hardware que lo corre. El sistema operativo fue lanzado en 2008 por la alianza de empresas "Open Hanset Alliance" y al igual que iOS se actualiza aproximadamente una vez al año. Sin embargo por la política comercial de los distintos fabricantes existe una gran fragmentación en el sistema ya que no todos los dispositivos reciben las actualizaciones de sistema operativo. Actualmente la última versión es la 4.4.3 (KitKat) y aunque recientemente se ha hecho una presentación de la siguiente versión (nombre en clave L) todavía no se conoce su nombre definitivo ni su fecha de salida al mercado móvil.

Windows Phone: Creado por Microsoft, es la evolución del sistema operativo Windows Mobile hacia las plataformas móviles. Basado en el núcleo de Windows NT y con un diseño estético similar al de las versiones de escritorio es la apuesta de Microsoft para los dispositivos móviles. Fue lanzado en 2010 y la frecuencia de actualización es menor que la de sus competidores. Actualmente está en el mercado la versión Windows Phone 8.

BlackBerry OS: Sistema desarrollado por la empresa "Research in Motion" (RIM) propietaria de los dispositivos BlackBerry. Al igual que iOS se trata de un sistema cerrado desarrollado solo para ser usado con terminales BlackBerry, tuvo gran repercusión en sus primeros años en el mercado gracias al apoyo de las operadoras de telefonía, pero poco a poco ha ido descendiendo su cuota de mercado. Se lanzó en 2003 y la frecuencia de actualización es semejante a la del sistema Windows. Actualmente está en el mercado la versión BlackBerry 10 OS.

Symbian OS: Fue el resultado de la alianza de varias empresas de telefonía móvil encabezadas por Nokia. El objetivo de este sistema operativo era poder competir con las PDA's en un momento en el que los Smartphone aún no existían, era un sistema muy básico y con pocas funcionalidades que con la llegada de los sistemas actuales ha ido desapareciendo poco a poco. Fue lanzado en 1997 por el consorcio "Symbian Foundation" y la última versión que salió al mercado fue la versión Symbian 9.5.

Firefox OS: A día de hoy sigue siendo más una apuesta de futuro que una realidad, es la apuesta de la comunidad de software libre "Mozilla Corporation" para entrar en el mundo de los dispositivos móviles. Su funcionamiento está basado en HTML5 con núcleo Linux y apuesta fundamentalmente por los dispositivos móviles de gama baja y media. Fue presentado en 2013 pero por ahora su tasa de implantación en el mercado es muy baja, aunque se espera un gran crecimiento de su cuota de mercado en los próximos años gracias al apoyo de telefónica, que forma parte del consorcio de empresas que ha apoyado a Mozilla Corporation en el desarrollo de este sistema operativo.

Otros: Además de los sistemas operativos mencionados, existen otros, pero o bien su implantación es poca, o bien son sistemas que poco a poco han ido cayendo en desuso. Entre ellos podemos nombrar "Kindle" que es una versión de Android adaptada exclusivamente para los dispositivos de Amazon, "Java ME" que fue el sistema operativo que sirvió de soporte a la

transición entre los teléfonos móviles y los actuales Smartphone, “Bada” la apuesta de futuro de Samsung que no ha conseguido despegar, “Ubuntu Touch” que actualmente se encuentra en fase de desarrollo y se espera que sea una alternativa real a los sistemas existentes y en último lugar estarían otros sistemas propietario que llevaban el nombre de la compañía que los desarrollaba y que convivieron con Java ME durante la transición de los dispositivos móviles y que poco a poco han ido desapareciendo (LG, BREW, ZTE, Samsung).

A continuación se expone un gráfico a modo de resumen con las características de cada uno de los sistemas operativos:

	iOS	Android	Windows Phone 8	BlackBerry	symbian OS	Firefox OS
Compañía responsable	Apple	Open Handset Alliance	Microsoft	RIM	Symbian Foundation	Mozilla Corporation
Núcleo del S.O.	Mac OS X	Linux	Windows NT	Mobile OS	Mobile OS	Linux
Tipo de licencia de software	Propietaria	Software libre y abierto	Propietaria	Propietaria	Software libre	Software Libre
Año de lanzamiento	2007	2008	2010	2003	1997	2013
Fabricante único	Si	No	No	Si	No	No
Variedad de dispositivos	Modelo único	Muy alta	Media	Baja	Muy alta	Baja
Memoria externa	No	Si	Si	Si	Si	Si
Motor navegador web	WebKit	WebKit	Pocket Internet Explorer	WebKit	WebKit	WebKit
Soporte Flash	No	Si	No	Si	Si	No
HTML5	Si	Si	Si	Si	No	Si
Tienda de Apps	App Store	Google Play Store	Windows MarketPlace	BlackBerry App World	Ovi Store	Firefox MarketPlace
Número de Apps	1.200.000	1.200.000	160.000	130.000	100.000	4000
Coste publicar Apps	99 $/año	25 $ pago único	99 $/año	Sin coste	1 $ pago único	Variable
Actualizaciones automáticas S.O	Si	Dependiente del fabricante	Dependiente del fabricante	Si	Si	Si
Familia CPU	ARM	ARM, MIPS, Power, x86	ARM	ARM	ARM	ARM
Máquina virtual	No	Dalvik	.net	Java	No	Si
Aplicaciones nativas	Siempre	Si	Si	No	Siempre	No
Lenguaje de programación	Objective-C, C++	Java, C++	C# y muchos otros	Java	C++	HTML5, Java
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux	Windows, Mac, Linux

Tabla 2: Comparativa de sistemas operativos móviles.

Al igual que se ha hecho anteriormente con los dispositivos móviles, a continuación se hará una breve comparativa del grado de implantación en el mercado español de los diferentes sistemas operativos en porcentaje de usuarios. Según la consultora "Kantar Worldpanel ComTech[5]" en marzo del 2014 el 88.6% de los usuarios de telefonía móvil de España que usan un Smartphone para conectarse a internet lo hacen desde un Smartphone con sistema operativo Android (independientemente de la versión del sistema operativo que integre). En la siguiente imagen se puede ver el porcentaje de usuarios de cada plataforma según los datos de esta consultora:

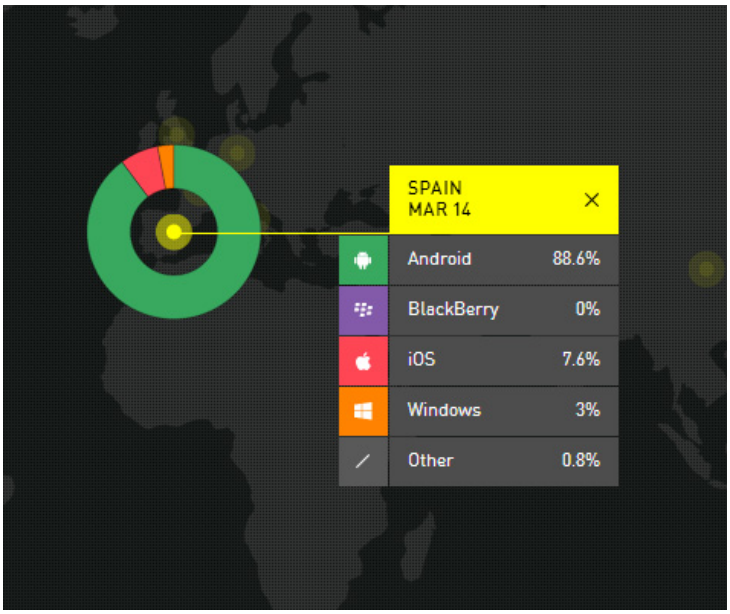
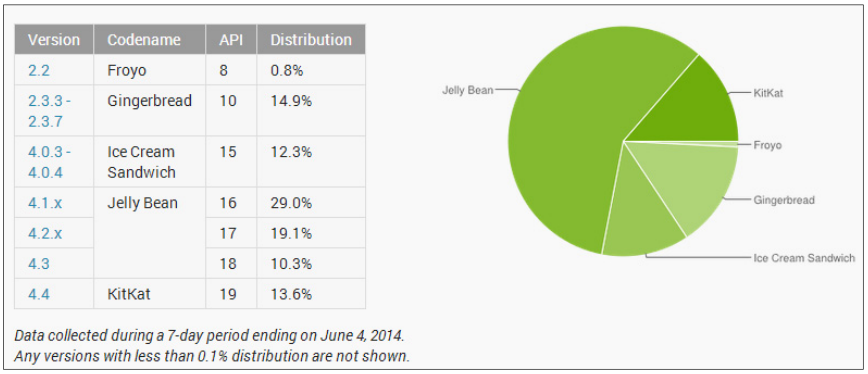


Imagen 4: Porcentaje de conexiones según consultora Kantar Worldpanel ComTech.

Vistas todas las características de las plataformas móviles disponibles en este momento, se toma la decisión de desarrollar la aplicación del presente proyecto para Android debido a su gran implantación, el bajo coste de publicación y la gran variedad de dispositivos en los que podrá ser ejecutado.

Sin embargo, dentro de Android hay que tomar una decisión importante y es la versión mínima en la que podrá ejecutarse la aplicación. Esta es una decisión importante puesto que poner una versión mínima demasiado baja implica que se perderán funcionalidades de Android que fueron implementadas en versiones posteriores y que en estas más versiones más antiguas no pueden ejecutar. A su vez, el hecho de escoger una versión demasiado nueva implica dejar a muchos usuarios fuera de la aplicación debido a la fragmentación de versiones existente en Android.

Antes de tomar esa decisión conviene investigar el grado de uso de cada una de las versiones de Android disponibles en el mercado. Según la página web de desarrollo de Android[6], actualmente la distribución de uso de las versiones de Android es la que podemos ver en la siguiente imagen:



Version	Codename	API	Distribution
2.2	Froyo	8	0.8%
2.3.3 - 2.3.7	Gingerbread	10	14.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	12.3%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	19.1%
4.3		18	10.3%
4.4	KitKat	19	13.6%

Data collected during a 7-day period ending on June 4, 2014.
Any versions with less than 0.1% distribution are not shown.

Imagen 5: Uso de las versiones de Android.

Tras analizar las funcionalidades que ofrecen las distintas versiones de Android y viendo el grado de implantación de las mismas, se toma la decisión de realizar el desarrollo de la aplicación con una versión mínima de API 13 (Android 3.0), de este modo se excluyen del mercado potencial alrededor del 15% de los dispositivos existentes, pero se permite el uso completo de la mayoría de las características de las versiones actuales de Android.

2.3. Geolocalización.

Se define la geolocalización como la identificación de la posición geográfica real de un objeto o persona, ya sea mediante un dispositivo conectado a internet (como un por ejemplo un Smartphone) o cualquier otro dispositivo que sea posible rastrear. Dicha localización hace referencia al posicionamiento que define la localización espacial de un punto en un sistema de coordenadas y un datum determinado. Existen varios métodos para determinar la posición de un aparato mediante geolocalización.

A continuación se enumeran algunos de ellos basándose en la clasificación realizada por Alan Bover Argelaga[7]:

Identidad Celular Global (CGI): Técnica desarrollada para la localización en redes GSM y posteriormente adaptada a las nuevas redes de comunicación. Basándose en la premisa de funcionamiento de las redes móviles como redes celulares compuestas por antenas que emiten a una cierta distancia, esta técnica propone que cada célula de telefonía tiene otras seis células contiguas a su alrededor y que por ello cada una de ellas ha de utilizar una frecuencia de señal diferente para que se puedan identificar en cada momento. Con esto si se conoce la célula a la que se está conectado se puede concretar que se está en un punto del radio de dicha célula. El margen de error de este sistema es variable, pues dependerá de la densidad de células disponible, siendo un margen de error de varios metros en ciudad o de cientos de metros e incluso kilómetros en zonas rurales con baja densidad de células. Esto descarta este método de localización para su uso fuera de las ciudades.

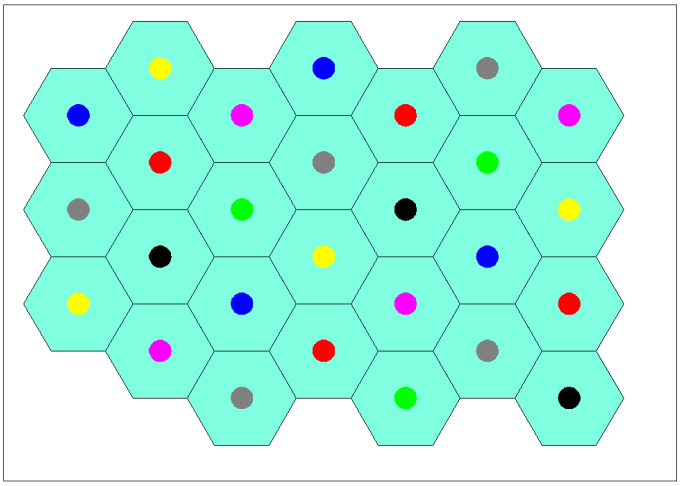


Imagen 6: Modelo de celdas de telecomunicaciones.

Identidad celular perfeccionada (CGI-TA): Esta técnica permite conocer la distancia entre celda y dispositivo a través del tiempo de retraso de la señal. Combina la detección de las celdas con la medición del tiempo de llegada produciendo un posicionamiento más preciso del dispositivo, pero que aun así, sigue siendo poco preciso en zonas rurales.

Ángulo de llegada (AOA, DOA): Este sistema se basa en el ángulo con el que las señales de las antenas llegan al dispositivo móvil. Para conocer la posición se necesita conocer al menos el ángulo de llegada desde dos antenas diferentes, lo cual es suficiente en entornos con poca densidad de antenas pero para conseguir un cálculo más preciso se necesitan al menos tres antenas. El gran inconveniente de este método es que se necesitan antenas de telefonía con unas características determinadas que las hacen más caras que las normales, por lo que en la práctica es menos usado.

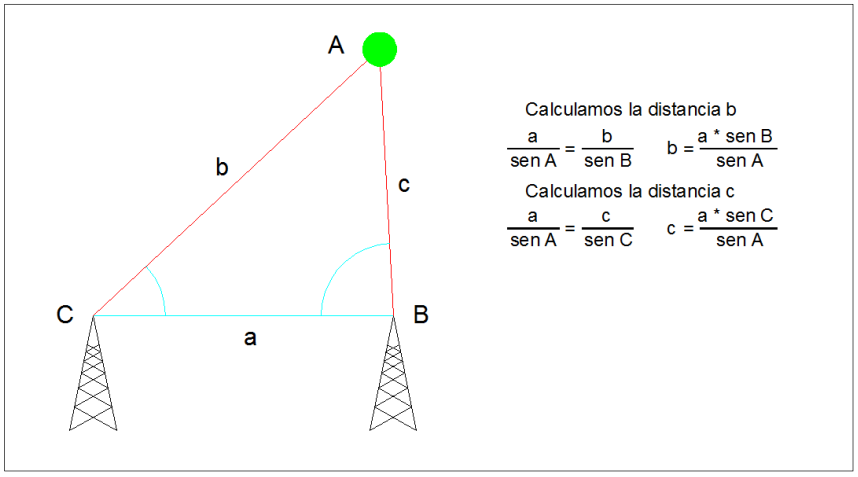


Imagen 7: Posicionamiento por ángulo de llegada.

Tiempo de llegada (TOA): Se basa en el tiempo que tarda en llegar una señal a las antenas cercanas. Partiendo de la base de que la señal viaja a la velocidad de la luz, si se conoce el tiempo que tarda en recorrer la distancia entre dispositivo y antena se podrá calcular la distancia a la que se encuentra. Con las distancias de al menos tres antenas se trazarán tres

circunferencias (una desde cada antena) y en el lugar que se corten se encontrará el dispositivo móvil. Las interferencias existentes y el hecho de necesitar comunicación y respuesta entre antenas y dispositivo crean errores de respuesta que dificultan el cálculo preciso de la posición.

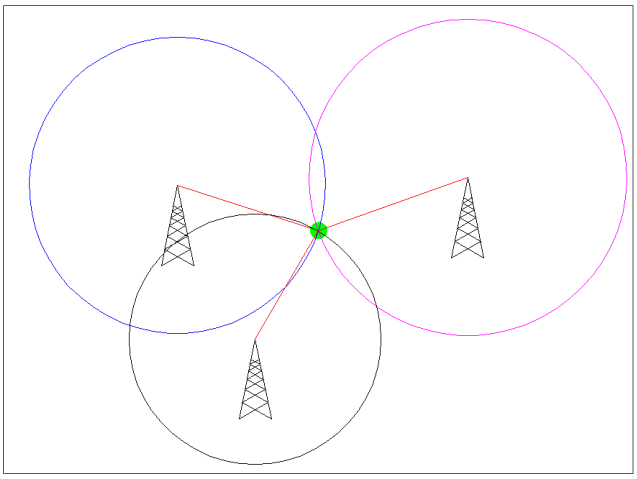


Imagen 8: Posicionamiento por tiempo de llegada.

Diferencia de tiempo de llegada (TDOA): Es un método semejante al anterior, pero en lugar de medir el tiempo en el que se recorre la distancia entre dispositivo y antena, se mide la diferencia de tiempo de llegada de la señal desde el dispositivo a un par de antenas. Una vez se conozca la diferencia de tiempo, si se marcan todos los puntos en que la distancia de cada punto a las antenas este definida por la diferencia de llegada se generará una hipérbola. Repitiendo el proceso con todos los pares de antenas disponibles se consigue que todas las hipérbolas se crucen en un punto, en el cual estará situado el dispositivo móvil. Este sistema al estar basado en mediciones a dos antenas en lugar de una, reduce los errores a causa de la reflexión. Es muy importante para que funcione que todos los relojes estén perfectamente sincronizados, lo que en la práctica no siempre está asegurado.

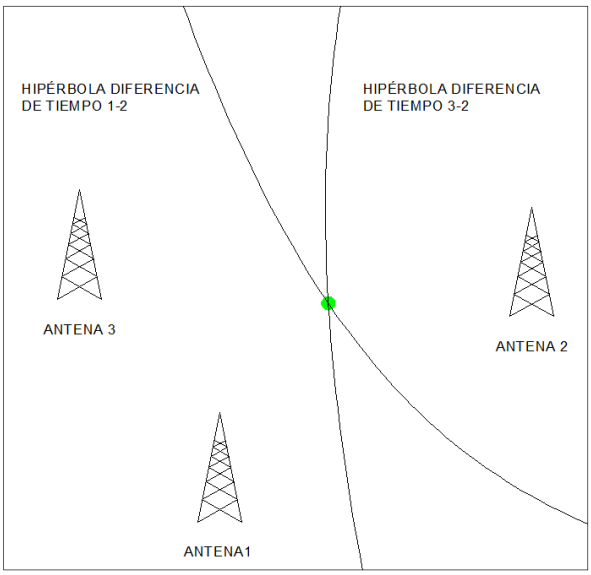


Imagen 9: Posicionamiento por diferencia de tiempo de llegada.

Huella multitrayecto: Esta técnica aprovecha el fenómeno de refracción de las ondas del dispositivo al llegar a la antena. Durante el trayecto de la señal esta rebota en obstáculos como edificios, casas, etc hasta llegar a la antena, lo que provoca que la antena reciba varias veces la misma señal con diferentes tiempos. Para que este sistema funcione ha de existir un entrenamiento previo donde se estudien los rebotes de las ondas en los distintos obstáculos y se cree una base de datos de señales con la que luego comparar los tiempos de llegada de las ondas rebotadas.

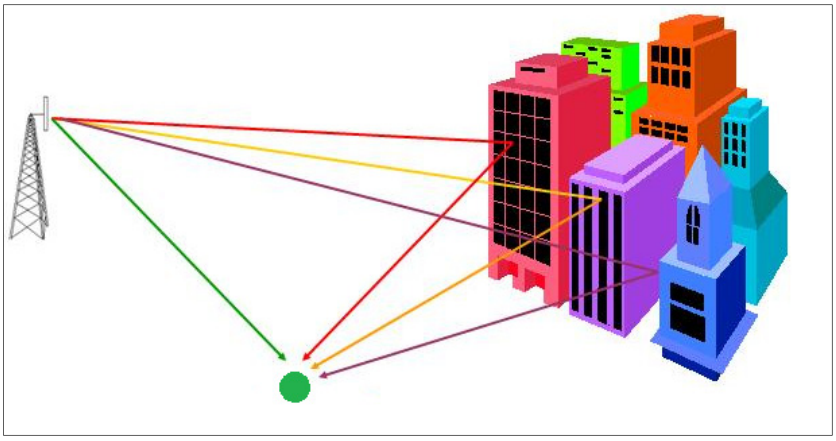


Imagen 10: Posicionamiento por huella multitrayecto.

Diferencia de tiempo de llegada con terminal modificado: La base de funcionamiento es la misma que la del método de tiempo de llegada, con la diferencia de que el dispositivo es capaz de saber el momento exacto en el que la antena envió la señal por lo que solo tendrá que calcular la diferencia de tiempo de envío y llegada para conocer la distancia. El problema de este método radica en la necesidad de sincronización de los relojes del dispositivo móvil y la antena.

Diferencia de tiempo de llegada perfeccionada (E-ODT): Basada en el concepto de diferencia de tiempo de llegada, pero siendo el dispositivo el que calcula la distancia y no el operador de la antena. También requiere sincronía de relojes. Para corregir los posibles desajustes de relojes, se añaden técnicas que comparan tiempos de reloj entre antenas para perfeccionar dichos tiempos.

Sistema global de navegación por satélite (GNSS): Es el término que engloba a los sistemas de navegación por satélite que proporcionaN un posicionamiento geoespacial con cobertura global, tanto de forma autónoma, como con sistemas de aumentación. El principio básico de funcionamiento del GNSS se basa en una constelación de satélites artificiales que emiten ondas electromagnéticas desde el espacio y que llegan al dispositivo móvil. Gracias a las efemérides que los satélites transmiten en su señal se puede conocer la posición exacta(x, y, z) de los satélites en cada momento. Con estas posiciones hay que descifrar las señales emitidas, para las cuales habrá un determinado tiempo de salida y un determinado tiempo de llegada. Recibiendo la señal de al menos cuatro satélites se realizarán trilateraciones con el objeto de despejar las cuatro incógnitas que presenta la localización (x, y, z, t). Para poder trabajar con estas distancias se requieren unas sincronías de reloj muy preciso, por lo que los satélites van equipados con relojes atómicos muy precisos con estabilidades del orden 10^{-13} - 10^{-14}

segundos/día. Los receptores utilizan relojes de cuarzo menos precisos pero ese inconveniente se supera sincronizando tiempos durante la etapa de inicialización del sistema[8].

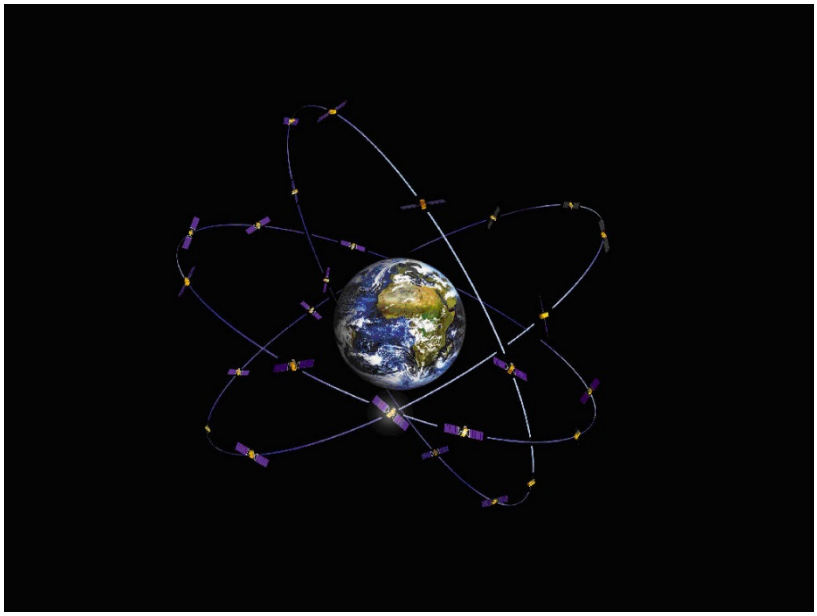


Imagen 11: Sistema GNSS.

El sistema tiene dos métodos de medida, por un lado la medición por código en la que el satélite emite una determinada marca de tiempo en la señal que el receptor repite simultáneamente de forma que cuando le llega la señal que ha emitido el satélite las compara y determina el tiempo que ha tardado en recibirla. Por otro lado está la medición de fase que se basa en que se controla en fase una emisión radioeléctrica hecha desde el satélite con frecuencia conocida y desde una posición conocida. Al controlar en fase, lo que se hace es observar continuamente la evolución del desfase entre la señal recibida y la generada por el receptor, es decir, el observable es el desfase entre ambas y éste cambia según lo hace la distancia entre satélite y receptor. Correlacionando continuamente ambas portadoras a partir del momento de conexión con el satélite, el receptor podrá contar la cantidad de ciclos enteros debido a los cambios de distancia entre satélite y receptor y medir la fracción de ciclo entre ambas señales. La diferencia entre ciclos observados y la cantidad total será la ambigüedad inicial N, que será invariable para toda la sesión, siempre que no se produzcan cortes en la señal, ya que en ese momento se perderá la cuenta de ciclos enteros y aparecerá una nueva ambigüedad. El problema que se plantea en la medida de fase es la dificultad que implica la obtención del número inicial de ciclos enteros en la portadora (N) contenidos en la distancia D que hay entre satélite y receptor[8].

Los principales errores que afectan a estas medidas son los provocados por la propagación de la señal por ionosfera y troposfera y los errores multipath por rebotes de la señal en su llegada a tierra.

En el caso de los Smartphone el método de medición usado es el de medición en código. Adicionalmente suelen usar sistema de referencia absoluto, aunque los últimos modelos

ya incorporan la capacidad de usar sistemas de aumentación como EGNOSS para trabajar con posicionamiento diferencial.

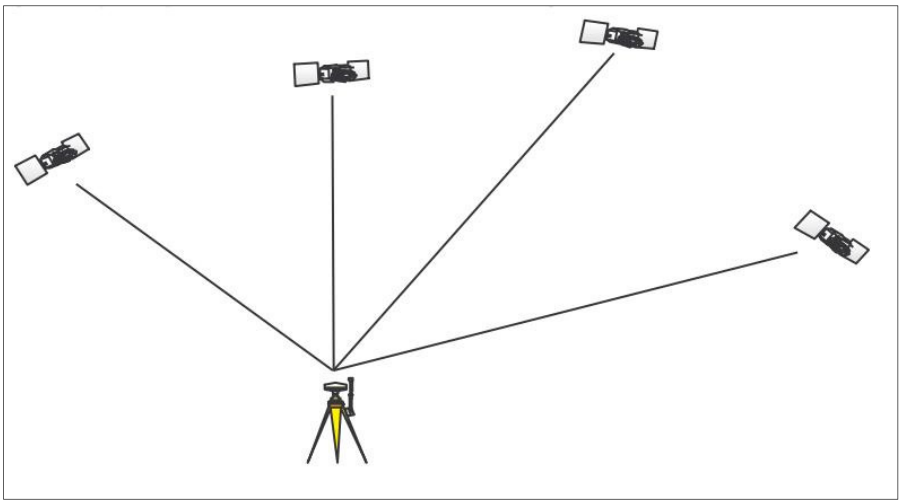


Imagen 12: Posicionamiento por GNSS.

A-GPS: Es una solución GNSS especialmente diseñada para dispositivos móviles. Básicamente trata de solucionar los problemas de tiempo de inicialización del GNSS en las ciudades recibiendo la señal de posición de los satélites GNSS con algún otro tipo de señal de datos (WiFi, 3G). Tiene dos sistemas de funcionamiento, el modo online, donde el A-GPS necesita una conexión activa y constante a la red de teléfono de la cual recibe una primera posición basándose en la celda y con ella información de los satélites que se encuentran en su entorno y las condiciones ionosféricas presentes para aumentar la precisión. En el modo offline, la conexión a la red telefónica no es constante, por lo que el A-GPS descarga un fichero con información de su celda o posición de satélites, etc., mientras disponga de esta conexión a internet para más tarde usar esta información cuando no disponga de dicha conexión.

Conocidas todas las técnicas de geolocalización disponibles se opta por el A-GPS, ya que es la más fiable y la que en teoría mejores coordenadas nos ofrecerá. Por tanto las coordenadas a partir de las que se trabajará serán coordenadas geográficas en el sistema de referencia WGS84.

2.4. WMS de cartografía móvil.

El "Open Geospatial Consortium[(9)]", organismo encargado de definir los estándares abiertos e interoperables de los Sistemas de Información Geográfica (SIG) y de la World Wide Web (WWW), define los Web map service (WMS) como una interfaz estándar HTTP para proveer imágenes georreferenciadas de una o más bases de datos geoespaciales. Una petición WMS define las capas geográficas y el área de interés a procesar y recibe como respuesta uno o más mapas georreferenciados en formato imagen que pueden ser mostradas en un navegador de internet o cualquier otro programa preparado para ello. La interfaz también soporta la capacidad para dar transparencia a las capas que se devuelven y de este modo combinar capas de diversos servidores en un solo mapa.

A la hora de desarrollar una aplicación para dispositivos móviles existen varias opciones en lo que a servidores de cartografía se refiere, a continuación se van a exponer algunos de estos servidores y sus características:

Google Maps: Fue lanzado el 8 de febrero de 2005 y es probablemente el servidor de cartografía online más conocido del mundo. Ofrece imágenes de mapas desplazables, así como fotografías por satélite de todo el mundo, está capacitado para ofrecer rutas a los usuarios y gracias a la gran inversión realizada por Google en este servicio dispone de imágenes a pie de calle de gran parte de las zonas habitadas del planeta. Su integración en Android es nativa, puesto que es el servidor de mapas que usa el sistema operativo para todas sus necesidades. El uso de sus API en el desarrollo de aplicaciones para Android es gratuito (exclusivamente en Android en otros entornos hay limitaciones de uso en función de las conexiones mensuales).

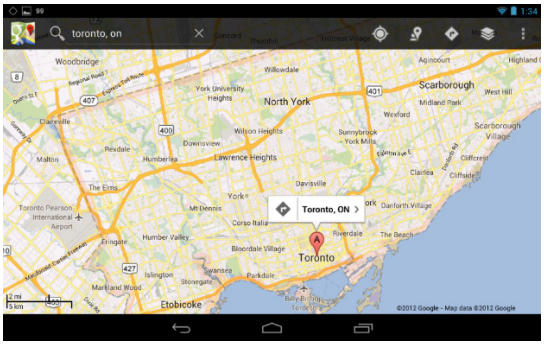


Imagen 13: Captura de Google Maps en Android.

Bing Maps: Lanzado por Microsoft en julio de 2005 como respuesta a la gran acogida de Google Maps. A día de hoy los servicios que ofrecen ambos servidores son muy semejantes, diferenciándose los mapas de Bing en las opciones de visualización 3D. Se puede usar en Android gracias a la API que Microsoft ha liberado, aunque su uso gratuito está restringido a 125.000 conexiones mensuales.

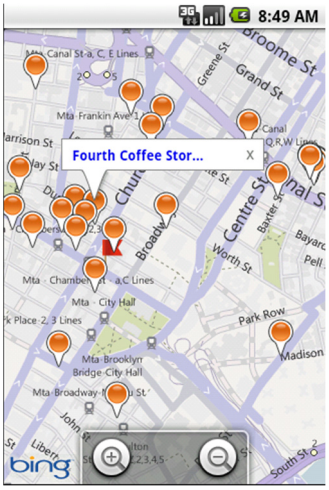


Imagen 14: Captura de Bing Maps en Android.

Yahoo! Maps: A pesar de ser el más antiguo de los servicios es probablemente el más desconocido, fue lanzado en marzo de 2002 por Yahoo!. Los servicios que ofrecen son

semejantes a los del resto de competidores pero carece de visualizaciones 3D. Se puede usar en Android mediante la correspondiente API, aunque es de pago desde la primera conexión.

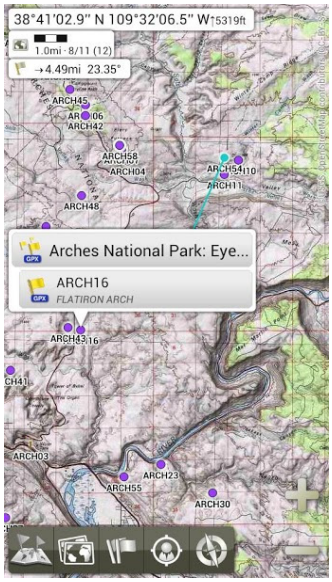


Imagen 15: Captura de Yahoo! Maps en Android.

OpenStreetMap: Nació en julio de 2004 de la mano del inglés Steve Coast, cuando era estudiante. Debido a la carencia de datos cartográficos libres en ese momento y al alto precio que las agencias gubernamentales le pedían, decidió crear sus propios mapas. Poco a poco OpenStreetMap ha ido creciendo hasta convertirse en una fundación. A día de hoy es un proyecto de cartografía libre a nivel mundial que ofrece cartografía para su uso libre de todo el mundo. Se puede usar en Android gracias a la API que facilitan y no tiene coste de uso (debido a la licencia de propiedad intelectual que usan tan solo es necesario citarlos).

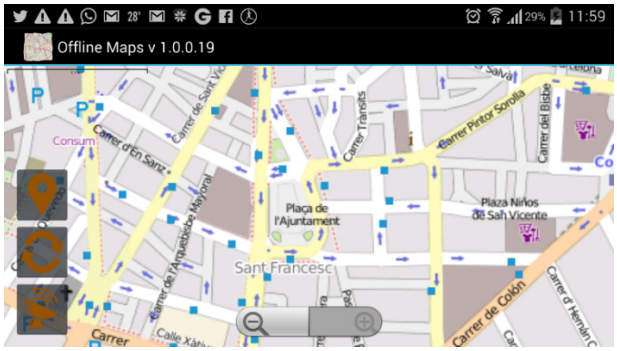


Imagen 16: Captura de OpenStreetMap en Android.

Otros servidores: Existen otros servidores de cartografía online que puede ser integrables en Android, pero que son menos usados por diferentes motivos como los altos precios de uso o la dificultad de integración. Entre ellos estarían "World Wind" de la NASA, "Ovi Maps" de Nokia, "MapQuest" de AOL o la propia API de ESRI que permite conectar con servidores WMS que no poseen API para utilizarlos en Android.

En el siguiente cuadro tenemos un pequeño resumen de las características de cada uno de los servidores WMS:

	Google Maps	bing	YAHOO!	OpenStreetMap FOUNDATION
Compañía responsable	Google	Microsoft	Yahoo!	OpenStreetMap Foundation
Tipo de licencia	Propietario	Propietario	Propietario	Open DataBase
Navegadores soportados oficialmente	IE, Firefox, Safari, Mozilla, Opera, Chrome	IE, Firefox, Safari	IE, Firefox, Opera, Safari	IE, Firefox, Chrome, Safari
Vistas	Vertical, horizontal, relieve, Rotación, 360 panorámica (Street View), Modo 3D	Vertical, horizontal, relieve, 360 panorámica (Streetside) Modo 3D (Tilt, Pan, Rotate)	Vertical, horizontal, relieve, 360 panorámica (Street View)	Vertical, horizontal, relieve
Niveles de zoom	19	23	19	19
Tipos de mapas	6: Mapa, satélite, híbrido, calle, tráfico, 3D	10: Camino, vía, satélite, híbridos, ojo de pájaro, tráfico, 3D, calle, Callejero de Londres, ordenación, lugares	4: Mapa satélite, Híbrido, calle, tráfico (EE.UU)	5: Estándar de mapas, transportes, bicicletas, mapa de exploración, mapa humanitario
Formato de intercambio de datos	JSON	.NET	JSON, XML	XML
Edad imágenes de satélite	1-3 años	1-3 años	1-3 años	1-4 años
Proveedores de datos	MapIT, TeleAtlas, DigitalGlobe, MDA Federal	NAVTEQ, Intermap, Pictometry International, NASA	NAVTEQ, TeleAtlas, i al Cubo, dominio público	Contribuciones de usuarios
Ubicaciones	Código postal, nombre de calle, ciudad, barrio, Longitud/latitud	Código postal, nombre de calle, ciudad, barrio, Longitud/latitud	Código postal, nombre de calle, ciudad, barrio, Longitud/latitud	Código postal, nombre de calle, ciudad, barrio, estado, región, Longitud/latitud
Instrucciones de viaje	Si	Si	Si	Si
Invertir direcciones	Si	Si	Si	No
Tipos de indicaciones	Coche, transporte público, a pie, bicicletas	Coche, transporte público (metro Londres), a pie	Coche, transporte público, a pie	Coche, transporte público, a pie, bicicletas, sillas de ruedas
Imprimibles	Si	Si	Si	No directamente

Tabla 3: Comparativa servidores WMS.

Debido a todo lo comentado se ha decidido que por la naturaleza del proyecto que se va a realizar y por la plataforma de desarrollo elegida, la mejor opción de WMS para este proyecto es la de Google Maps, puesto que está perfectamente integrado en el sistema operativo y se puede usar de modo gratuito.

3. Diseño de la aplicación.

3.1. Requisitos funcionales de la aplicación.

Una vez decidido el dispositivo para el que se va a programar, el sistema operativo en el que correrá la aplicación, el sistema de geolocalización a utilizar y el servidor WMS que se utilizará de base para visualizar los puntos, se procede a diseñar la aplicación.

La idea consiste en programar una aplicación, que haciendo uso del sensor GNSS del dispositivo móvil, pueda leer coordenadas en un punto y almacenarlas con unas características determinadas para generar inventarios de mobiliario urbano, hacer estimaciones de superficies en preparaciones de levantamientos topográficos o cualquier otro trabajo que se desee realizar, siempre teniendo en cuenta las limitaciones de precisión que este tipo de dispositivo nos pueden ofrecer.

Para que la aplicación pueda ser funcional, se piden los siguientes requisitos:

- Posibilidad de almacenar diferentes trabajos y gestionarlos con múltiples opciones.
- Posibilidad de almacenar puntos en el dispositivo sin que se pierdan al cerrar la aplicación.
- Posibilidad de decidir los tiempos de lectura que el usuario considere para aproximar mejor las coordenadas.
- Posibilidad de agregar comentarios y fotografías a los puntos que el usuario decida.
- Posibilidad de continuar el último trabajo de manera rápida.
- Posibilidad de exportar las coordenadas obtenidas a un ordenador tradicional de la manera más sencilla posible.
- Posibilidad de visualización de los puntos almacenados sobre un mapa en tiempo real.
- Posibilidad de calcular superficies en tiempo real discriminando los puntos que no sean relevantes para dicha superficie.

Con estos requisitos en mente se procede al diseño de la aplicación.

3.2. Arquitectura de tres capas.

Siguiendo los conceptos básicos de programación por capas descritos en bibliografía[10,11], se realizará un diseño de la aplicación en tres capas, que consiste en separar el funcionamiento de la aplicación en tres capas diferenciadas, presentación, negocio y datos. Esto hace que en caso de modificaciones del programa los datos no se alteren en ningún momento y que pueda modificarse cualquiera de las capas sin necesidad de que se modifiquen las otras. Las capas que se suelen usar en este tipo de diseños son las siguientes:

- Capa de presentación: es la que ve el usuario, es la que ejerce las labores de presentación de la información y de captura de datos. Se suele procurar que este

tipo de capa no haga cálculos complejos y que se límite a comprobaciones básicas de formato para evitar errores en los datos que se muestran o se recogen.

- Capa de negocio: Es donde residen los programas propiamente dichos. En esta capa se suelen recibir las peticiones realizadas a través de la capa de presentación y es donde tienen lugar los procesos más complejos. Es la capa que se encarga de hacer las peticiones a la capa de datos.
- Capa de datos: Es donde se almacenan los datos y es la encargada de gestionarlos. Estará formada por uno o varios gestores de bases de datos que realizan todo el almacenamiento de los mismos, reciben las solicitudes de almacenamiento o recuperación de información de capa de negocio.

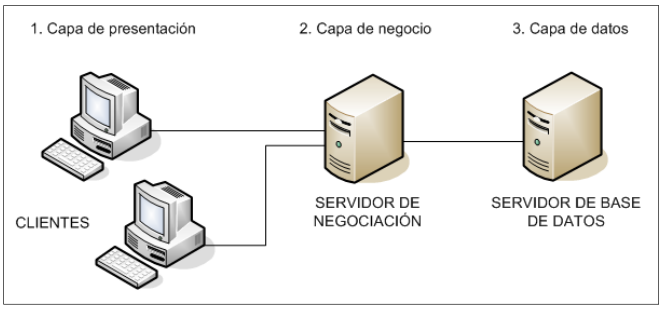


Imagen 17: Ejemplo de funcionamiento de la arquitectura de 3 capas.

3.3. Entorno de desarrollo.

Como ya se comentó en el aparatado 2.2 de sistemas operativos móviles, Android permite programación en Java o en C++, en este caso se ha optado por desarrollar en Java, ya que es un lenguaje de programación muy potente y es el que se estudia en el "Grado en ingeniería Geomática y Topografía". Para programar para Android en Java es necesario instalar en el ordenador el entorno de desarrollo eclipse, en este caso se hizo uso de la última versión disponible, la versión Kepler.

Imagen 18: Eclipse, versión Kepler.

En teoría ya se podría programar solo con eclipse y Java, pero Google ha puesto al servicio de la comunidad de desarrolladores el "Software Development kit" (SDK) de Android, que se compone de un conjunto de herramientas y APIS para facilitar el desarrollo de software específico para Android. El SDK consiste en plugin que se instala sobre la distribución de eclipse y gestiona los paquetes de Android instalados en el propio Eclipse para que el desarrollador

decida en cada momento que paquetes necesita para su aplicación y así resulte mucho más sencillo el proceso.

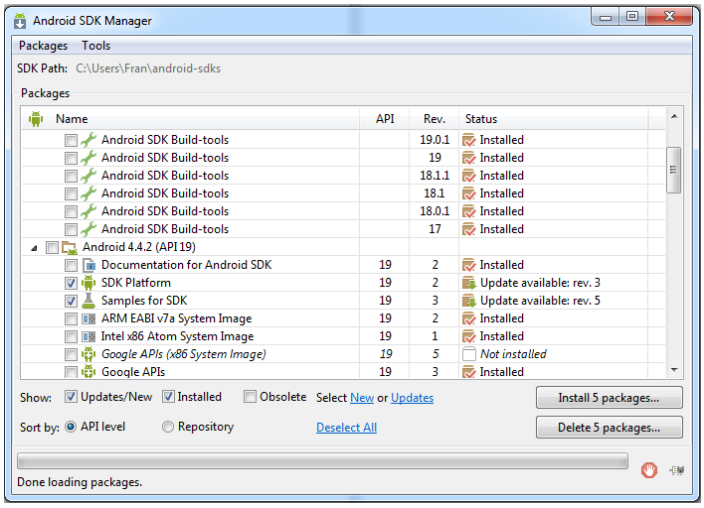


Imagen 19: SDK de Android.

Por otro lado, además de un gestor de paquetes, integra en eclipse un emulador de dispositivos con Android para que en cualquier fase del desarrollo el programador pueda ir probando su aplicación y conocer en todo momento si esta cumple con todo lo que se le solicita.

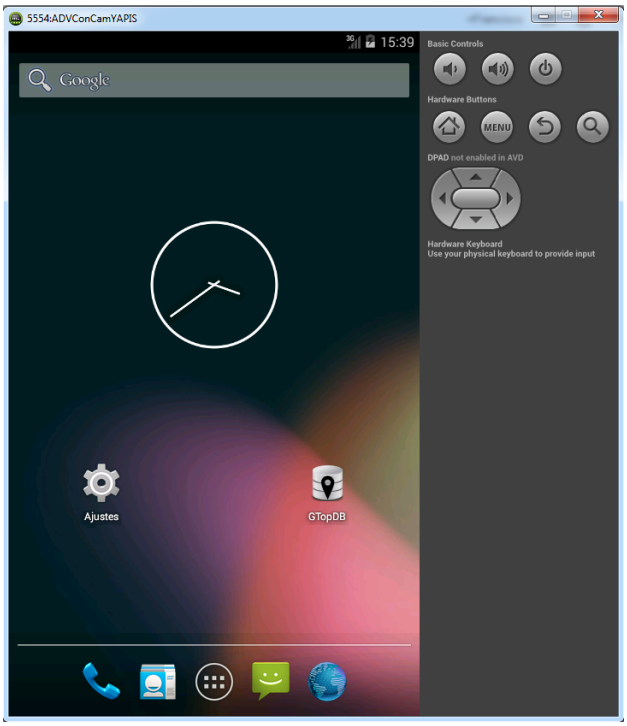


Imagen 20: Emulador de dispositivos Android.

Con todas estas herramientas y con las funciones del programa definidas, ya se puede comenzar a desarrollar la aplicación.

3.4. Estructura del programa.

3.4.1. Base de datos.

El primer paso que se ha seguido al desarrollar este programa es el diseñar la base de datos en la que se almacenaran los trabajos y los puntos que se deseen guardar. Para ello se ha diseñado una única base de datos que constará de dos tablas. El diseño de la base de datos lo podemos ver en la siguiente tabla:

Base de datos: trabajos					
Tabla: "trabajos"			Tabla: "Puntos"		
Nombre del campo	Tipo de dato	Otros	Nombre del campo	Tipo de dato	Otros
_id	INTEGER	Clave primaria	_id	INTEGER	Clave primaria
nombre	TEXT	No nulo	idtrabajo	INTEGER	Calve ajena
tipo	INTEGER	No nulo	numero	INTEGER	No nulo
entidad	TEXT		tipo	INTEGER	No nulo
persona	TEXT		longitud	REAL	No nulo
direccion	TEXT		latitud	REAL	No nulo
telefono	INTEGER		precision	INTEGER	
mail	TEXT		observaciones	TEXT	
comentario	TEXT		foto	TEXT	
fechaini	LONG	No nulo	fecha	LONG	No nulo
fechamod	LONG				

Tabla 4: Estructura de la base de datos.

Se define una primera tabla para los trabajos cuyos elementos son los siguientes:

- "_id": se define como un número entero que se autoincrementará automáticamente y será la clave primaria de esta tabla de modo que nunca se repetirá y a nivel interno será el identificador único de cada trabajo.
- "nombre": Nombre del trabajo, el único requisito será que no sea nulo, ya que en tal caso el programa no guardará el registro.
- "tipo": se han definido los tipos de trabajo mediante una lista enumerada de Java. De este modo bastará con que la base de datos almacene el número entero que corresponda al tipo de trabajo. Se requiere que no sea nulo, pero como veremos más adelante, por el propio diseño del programa nunca podrá serlo.
- "entidad": hace referencia a la entidad que pide el trabajo, puede ser nulo ya que no necesariamente todos los trabajos tendrán un peticionario concreto. El tipo de dato es texto.
- "persona": hace referencia al cliente del trabajo o a la persona de contacto en caso de ser dependiente de una entidad. Al igual que en el caso anterior es de tipo texto y se permite que el campo esté vacío.
- "direccion": será la dirección del cliente. Formato texto y puede ser nulo.
- "telefono": número de contacto del cliente. Formato de número entero, puede ser nulo.
- "mail": dirección de correo electrónico del cliente. Formato texto, puede ser nulo.
- "comentario": campo reservado para posibles anotaciones que se quieran realizar sobre el trabajo y no tengan cabida en el resto de apartados. Formato texto, puede ser nulo.

- "fechaini": Guardado como un solo número largo en el formato interno de Android es la fecha y hora de inicio del trabajo y se graba automáticamente al grabar el trabajo.
- "fechamod": Al igual que el campo anterior se trata de un número largo en formato interno que contiene fecha y hora de la última modificación. Este campo se mantiene sin intervención del usuario.

La segunda tabla que se define es la tabla "puntos" que consta de los siguientes campos:

- "_id": Del mismo modo que se hizo en la tabla de trabajos se define como un número entero que se autoincrementará automáticamente y será la clave primaria de esta tabla. A nivel interno será el identificador único de cada punto.
- "idtrabajo": Campo mantenido por el propio programa se trata de un número entero que hace referencia al trabajo al que pertenece y que se corresponde con el campo "_id" de la tabla "trabajos". Es la clave ajena.
- "numero": Es el número de punto, será un número entero que en la propia estructura del programa se impide que pueda ser nulo.
- "tipo": Al igual que para los trabajos, se ha definido una clase enumerada de Java en la que definen todos los puntos posibles. Solo es necesario almacenar el número entero que se corresponde con el orden del tipo en la clase enumerada. La estructura del programa evita que pueda ser nulo.
- "longitud": Se trata del campo que almacena la longitud leída por el sensor GNSS, es un número de tipo real y no puede ser nulo.
- "latitud": Idéntico al anterior pero para la latitud leída.
- "precision": Almacena la precisión que el sensor GNSS ha estimado durante la lectura de coordenadas. Se trata de un entero y no puede ser nulo.
- "observaciones": Cualquier comentario o anotación que se quiera realizar sobre el punto leído. Se trata de un campo de texto y puede ser nulo.
- "foto": En este campo se guarda una dirección de tipo "Uri" que hace referencia a la ruta en la que se ha guardado la fotografía del punto dentro del dispositivo móvil (si existe). Es un campo de texto y puede ser nulo.
- "fecha": Tiene las mismas características que los campos de fechas de la tabla trabajo y al igual que esos campos es mantenido por el propio programa.

3.4.2. Objetos definidos en el programa.

Se podría decir que la orientación a objetos es la base de estructuración del lenguaje Java. Java, al contrario que otros lenguajes, obliga a que toda la lógica de programación esté encapsulada en objetos. La programación orientada a objetos es un paradigma de la programación (reglas que rigen una forma de programar) que permite descomponer el código en pequeñas unidades lógicas llamadas objetos[12]. Un objeto puede ser un trabajo o un punto y cada uno de estos objetos tendrá unas determinadas características:

- Poseerá propiedades. Todos los puntos poseerán coordenadas, las cuales serán únicas para cada punto, pero que sin embargo su existencia es una propiedad común a todos los puntos. Para definir un objeto y sus propiedades será necesario definir la clase de objeto a la que pertenecerá.
- Pueden realizar acciones. En el caso de un punto podremos realizar acciones limitadas sobre el cómo crearlo, actualizar alguna de sus propiedades o hacer algún calculo con ellas. Para definir estas acciones habrá que definir los métodos de la clase a la que pertenecen los objetos.
- Pueden relacionarse entre sí. Un objeto de tipo trabajo puede contener muchos objetos de tipo punto, por ello no es necesario que se definan todos los datos del trabajo en cada punto, bastará con hacer referencia en una propiedad del punto a algún identificador del objeto trabajo en cada punto.
- Pueden formar parte de una organización. A menudo un objeto habrá objetos que puedan acoger varios tipos de objetos más específicos, por ejemplo dentro de un objeto trabajo, podríamos crear varios objetos que tuviesen propiedades no comunes al resto de trabajos, es decir podríamos tener una subclase de trabajos en cuevas que requieran propiedades especiales, para ello el nuevo objeto más específico tan solo tendría que heredar las propiedades del objeto general sin necesitar volver a definirlas.

Debido a la filosofía de Java que se acaba de explicar se van a tratar los datos como objetos y para ello se necesitan definir las siguientes clases:

"Trabajos.class": La clase que se ha definido para los trabajos consta prácticamente de los mismos atributos que se han definido en la base de datos a excepción del atributo "id" que es exclusivo de la base de datos y no es necesario utilizarlo para gestionar los trabajos. Como se comentó durante la descripción de la base de datos, para esta clase se ha creado una lista enumerada de tipos de trabajo, la podemos ver en la siguiente tabla:

Valor ordinal (BD)	Clave interna	Texto a mostrar al usuario	Imagen que define el tipo de trabajo
0	OTROS	Otros	
1	INVENTARIO	Inventario	
2	LEVANTAMIENTO	Planificación de levantamiento	

Tabla 5: Tipos de trabajo.

Dentro de la propia clase se han definido los constructores para el objeto, con tres opciones de constructor, la primera que define el objeto con todas sus propiedades, la segunda opción que lo define solo con el nombre de trabajo y el tipo de trabajo y la tercera opción que solo será usada en ciertos procesos internos que define un objeto sin datos pero al que el constructor asigna los datos necesarios para poder manejarlo. Del mismo modo se han definido los métodos necesarios para poder actualizar el valor de cualquiera de las propiedades del objeto o para leerlas cuando sea necesario.

"Punto.class": El la clase que define los objeto de tipo punto, sus propiedades son iguales a las de la base de datos salvo por el campo "id" que no se utiliza en el objeto y los campos "latitud" y "longitud" que se agrupan en una propiedad denominada "GeoPunto" y que hace referencia a otra clase que será la siguiente que se verá. Al igual que en el caso de la clase trabajo, para la clase de puntos se ha creado una lista de posibles valores que podrán tomar los puntos. Para definirlos se ha tomado como referencia un trabajo de inventariado de urbanización realizado con anterioridad. En la siguiente tabla podemos ver los tipos de punto que se han definido:

Valor ordinal (BD)	Clave interna	Texto a mostrar al usuario	Imagen que define el tipo de punto	Imagen que define el grupo de tipos de puntos
0	OTROS_PUNTOS	Otros puntos		
1	SUMIDERO	Sumidero		
2	POZO_PLUVIALES	Pozo aguas pluviales		
3	ACOMETIDA_PLUVIALES	Acometida red pluviales		
4	CANALETA_PLUVIALES	Canaleta aguas pluviales		
5	POZO_FECALES	Pozo aguas fecales		
6	ACOMETIDA_FECALES	Acometida red fecales		

7	HIDRANTE_ANTIINCENDIOS	Hidrante red contra incendios		
8	ARQUETA_INCENDIOS	Arqueta red contra incendios		
9	BOCA_RIEGO	Boca de riego		
10	ACOMETIDA_ABASTECIMIENTO	Acometida red abastecimiento		
11	POZO_ABASTECIMIENTO	Pozo red agua potable		
12	ARQUETA_TELEFONICA_TIPO_D	Arqueta red telefónica tipo D		
13	ARQUETA_TELEFONICA_TIPO_H	Arqueta red telefónica tipo H		
14	ARQUETA_TELEFONICA_TIPO_M	Arqueta red telefónica tipo M		
15	ARQUETA_TELECOMUNICACIONES	Arqueta red telecomunicaciones		
16	ACOMETIDA_TELECOMUNICACIONES	Acometida redes telecomunicaciones		
17	ARMARIO_TELECOMUNICACIONES	Armario red telecomunicaciones		
18	CENTRO_TRANSFORMACION	Centro distribución eléctrica		
19	ARQUETA_ELECTRICIDAD_CUADRADA	Arqueta red eléctrica cuadrada		

20	ARQUETA_ELECTRICIDAD_REDONDA	Arqueta red eléctrica redonda		
21	ACOMETIDA_ELECTRICIDAD	Acometida red eléctrica		
22	FAROLA	Farola		
23	ARQUETA_ALUMBRADO	Arqueta alumbrado público		
24	ARQUETA_CONEXION_ALUMBRADO	Arqueta de conexión de farolas a red		
25	SEMAFORO	Semáforo		
26	ARQUETA_CONEXION_SEMAFORO	Arqueta de conexión de semáforos a red		
27	CAMARA_SEGURIDAD	Cámara de seguridad		
28	ARMARIO_ALUMBRADO	Armario de alumbrado público		
29	ARMARIO_SEMAFORO	Armario de semáforos		
30	ARQUETAS_GAS	Arquetas red de gas		
31	ACOMETIDA_GAS	Acometidas red de gas		
32	ALIVIADERO_GAS	Aliviadero red de gas		

33	ARBOL	Árbol		
34	BANCO	Banco		
35	ALCORQUE	Alcorque		
36	FUENTE	Fuente de agua potable		
37	PAPELERA	Papelera		
38	ESCULTURA	Escultura		
39	ESCALERA	Escaleras		
40	BARANDILLA	Barandilla		
41	SEÑAL_PELIGRO	Señal de tráfico: peligro		
42	SEÑAL_PROHIBIDO	Señal de tráfico: prohibido		
43	SEÑAL_OBLIGACION	Señal de tráfico: obligación		
44	SEÑAL_INFO_CUADRADA	Señal de tráfico: informativa cuadrada		
45	SEÑAL_INFO_RECTANGULAR	Señal de tráfico: informativa rectangular		

46	BORDILLO_ACERA_NORMAL	Bordillo acera convencional	BN	
47	BORDILLO_ACERA_PEATON	Bordillo o límite de acera en zona peatonal	BP	
48	PERIMETRO	Perímetro del trabajo	P	
49	RIGOLA	Rigola	RG	
50	OTROS_LINEAS	Otras elementos lineales	OL	

Tabla 6: Tipos de punto.

Al igual que en la clase trabajo se han definido los constructores con tres opciones de constructor. La primera opción define el objeto con todas sus propiedades, la segunda opción lo define sin comentarios y la tercera opción, que al igual que en el caso anterior solo es usado por el sistema para procesos internos, crea el punto sin datos. Dentro de los constructores hace llamadas al constructor de la clase GeoPunto. Por último se han definido los métodos necesarios para actualizar el valor de las propiedades del objeto o para leerlas.

"GeoPunto.class": Es una clase específicamente generada para trabajar con las coordenadas. Esta clase es llamada cuando se crea un nuevo punto y a través de su constructor toma los valores de latitud y longitud leídos por el sensor GNSS. Tras leer las coordenadas se llama al método "completaPunto()" utilizando las ecuaciones de Coticchia-Surace[13], se transforman las coordenadas geográficas del punto a coordenadas UTM, completando las propiedades del objeto "x", "y" y "huso". A mayores se crean los métodos necesarios para actualizar o leer las propiedades del objeto.

3.4.3. Clases de negociación.

Ya se han definido los objetos que se utilizarán en la aplicación y la base de datos que almacenará todos los datos que sean leídos en campo. En este apartado se van a describir las clases que negocian entre la base de datos y las capas de visualización para mostrar la información al usuario o recoger la información que el usuario introduzca.

"Trabajos.class": Es la clase encargada de gestionar los objetos de tipo trabajo. Su único atributo es un objeto de la clase "TrabajosBD" que es con el trabajará, es una clase de tipo estático por lo que otras clases podrán trabajar con sus métodos sin necesidad de definirla un objeto de clase "Trabajos". En esta clase se han diseñado los siguientes métodos para atacar a la base de datos:

- *"inicializaBD":* Método imprescindible para el funcionamiento de la base de datos en cualquier clase desde la que se vaya a llamar a cualquier otro método de esta clase. Recibe el contexto de la actividad de Android sobre la que se usará e inicializa la base de datos para que pueda ser utilizada.
- "*elemento"*: Se llama a este método enviándole como parámetro un dato de tipo entero que será el identificador del trabajo que se está buscando. El método atacará a la base de datos en modo lectura y seleccionará el elemento de la tabla trabajos que posea el identificador que se le ha pasado. Si lo encuentra creará un objeto "trabajo" vacío e irá actualizando todos los atributos del objeto con los datos que ha recogido de la base de datos. Una vez completado el proceso devuelve el objeto de la clase "trabajo".
- *"actualizaTrabajo"*: Para llamar a este método hay que pasarle como parámetro un dato de tipo entero que será el identificador del trabajo y un objeto de tipo "trabajo" con los datos actualizados. A continuación ataca a la base de datos en modo escritura y en función de los atributos que se le hayan asignado al objeto que se ha enviado en la llamada a la función, actualizará de un modo determinado la base de datos. En caso de funcionamiento correcto, el método no devuelve nada y en caso de que no encuentre el identificador en la base de datos devolverá un error.
- *"nuevo"*: Este método no requiere parámetros para su llamada, simplemente atacará a la base de datos en modo escritura para crear un registro solo con identificador y fecha inicial. Devuelve un entero que es el identificador del trabajo creado. Este método siempre es utilizado en la aplicación acompañado en la línea siguiente por el método "actualizaTrabajo" de modo que aunque se cree una entrada vacía este suceso es momentáneo y enseguida se corrige, ya que antes de llamar al método la propia aplicación se asegura de tener los datos necesarios para poder actualizar el registro en la línea siguiente.
- *"borrar"*: Recibe como parámetro de entrada un entero que es el identificador del trabajo, ataca a la base de datos en modo escritura y como su nombre indica borra la entrada de la base de datos con ese identificador.
- *"listado"*: Método que no recibe ningún parámetro. Ataca a la base de datos en modo lectura para conseguir un listado de todas las entradas de la tabla de trabajos. Devuelve un objeto de tipo "cursor" que es un objeto propio de SQL con un listado en formato texto de las entradas de la base de datos que resultan de la consulta. La actividad que ha llamado al método se encargará de utilizar los datos del objeto "cursor" para formar los objetos que necesite o mostrarlos como sea conveniente.
- *"ultimoTrabajo"*: Este método tampoco necesita parámetros, simplemente ataca a la base de datos en modo lectura para conseguir un listado de trabajos ordenados por fecha de modificación, recoge el último trabajo modificado y

devuelve un entero que es el identificador del trabajo. Cuando se usa este método a continuación se hace una llamada al método "elemento".

- *"buscarNombre"*: Recibe como parámetro una cadena de texto con el nombre de un trabajo, a continuación ataca a la base de datos en modo lectura para buscar algún registro con ese nombre. En caso afirmativo devuelve el identificador del trabajo y en caso negativo devuelve "-1".

***"Puntos.class"*:** Se encarga de la gestión de los objetos de tipo "Punto". Es prácticamente igual a la clase "Trabajos", de hecho tiene el mismo único atributo y también es una clase estática que puede ser usada por el resto. La mayoría de métodos implementados son semejantes:

- *"inicializaBD"*: Inicializa la base de datos para poder trabajar con ella.
- *"elemento"*: Tiene la misma función y la misma estructura que el método de la clase "Trabajos" pero en lugar de utilizar objetos de tipo "trabajo" usa objetos de tipo "punto".
- *"actualizaPuntos"*: Es el método equivalente a "actualizaTrabajos".
- *"nuevo"*: Método equivalente al método "nuevo" de "Trabajos".
- *"borrar"*: Idéntico al método de "Trabajos".
- *"listado"*: Al igual que en la clase "Trabajos" devuelve un cursor con el listado de todos los puntos almacenados en la base de datos.
- *"listadotrabajo"*: Método semejante al anterior pero que recibe un valor entero, el cual es el identificador del trabajo que se está consultando. Devuelve un listado con todos los puntos almacenados en la base de datos que correspondan al trabajo seleccionado.
- *"ultimopuntotrabajo*": Este método tiene las mismas características que el método "ultimoTrabajo".
- *"buscarNumero"*: Método funcionalmente idéntico al método "buscarNombre" de la clase "Trabajos".
- *"buscarNumeroEnTrabajo"*: Método semejante al anterior, pero que además de recibir el valor entero del número de punto a buscar, recibe un segundo valor entero correspondiente al identificador del trabajo. De este modo el método busca si existe algún punto con ese número que pertenezca al trabajo indicado en la base de datos.
- *"listadoPerimetro"*: Este método recibe un entero, que es el identificador del trabajo sobre el que se hace la consulta y ataca a la base en modo lectura para seleccionar puntos que pertenezcan al trabajo seleccionado y cuyo tipo de punto haya sido definido como perímetro. Devuelve un objeto de tipo "Cursor" con el listado de puntos.

3.4.4. Clases de presentación.

Las clases de presentación son aquellas con las que interactuará el usuario y a las que, en algunas ocasiones, se les requieren procesos poco complejos que serán realizados de forma

volátil ya que nunca accederán directamente a la base de datos. A cada clase de presentación se le asigna un fichero XML, que es un fichero de texto con leguaje de marcas que define la estructura visual que se mostrará al mostrar la clase. A continuación se van a repasar las distintas clases de presentación existentes en el programa, que a su vez son las pantallas con las que el usuario tendrá que interaccionar. Se pretende hacer una pequeña descripción de cada una de ellas:

"MainActivity" (1): Es la ventana principal de la aplicación. Cuando se abre la aplicación esta es la primera pantalla desde la que se arranca y es la que hace posible llamar al resto de actividades de la aplicación. En la siguiente imagen podemos ver una captura de esta pantalla:

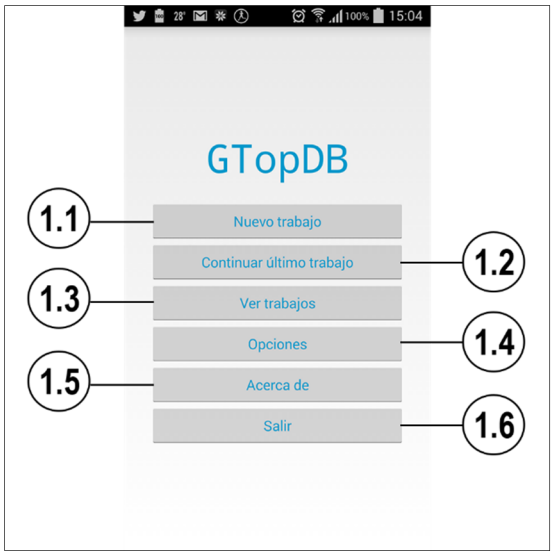


Imagen 21: Pantalla principal de la aplicación.

Como se puede ver en la imagen se ha diseñado con una interfaz de manejo sencillo, siendo a su vez muy básica a nivel estético. Se pueden observar varios botones que llaman a las diferentes funciones de la aplicación. Hay un botón para crear un nuevo trabajo (1.1), otro botón que permite continuar el último trabajo que en el que se realizaron modificaciones (1.2), otro botón para ver los detalles de los diferentes trabajos almacenados (1.3), otro para entrar a las opciones generales del programa (1.4), otro para obtener información básica sobre el programa (1.5) y un último botón que cierra el programa (1.6).

"NuevoTrabajo" (2): Una vez se pulsa el botón de nuevo trabajo en la pantalla principal (1.1), se abre esta clase, que contiene la pantalla para crear un nuevo trabajo:

Imagen 22: Pantalla de creación de trabajo.

Como se puede ver en la imagen, esta pantalla se compone de un formulario para rellenar los datos del trabajo. En primer lugar hay una caja de texto para rellenar el nombre del trabajo, en caso de que no se introduzca ningún carácter en esa caja de texto el programa no permitirá continuar, con esto se evita que se intenten introducir en la base de datos objetos con el nombre nulo. A continuación hay un "spinner", se trata de un menú desplegable que al pulsarlo muestra los tipos de trabajos existentes para que se pueda elegir el que más se adecúe al trabajo que se va a realizar. Este tipo de control de Android obliga a que siempre haya definido un tipo de trabajo, que por defecto será de tipo "Otros". Las siguientes cajas de texto que se pueden apreciar en la imagen son para rellenar el resto de datos que se le pueden atribuir a un trabajo, no existen restricciones en cuanto a que los campos estén vacíos, pero en caso de escribir en alguno de ellos el sistema solo permitirá escribir el tipo de dato que corresponde a ese atributo. En cuanto a los botones de acción, se han implementado 3, el primero (2.1) tiene como finalidad crear el nuevo trabajo, para ello toma los datos de todos los campos, comprueba que el campo nombre no está vacío y que no exista otro trabajo con ese nombre, en caso de que se cumpla almacenará el trabajo, mientras que si no se cumple no lo guardará y enviará un mensaje de error al usuario.

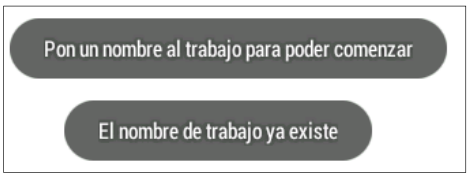


Imagen 23: Ejemplos de mensajes de error o aviso.

Si el trabajo se crea correctamente se abrirá la pantalla de lectura de puntos (3). Por otro lado están el botón para limpiar (2.2) que reinicia todas las cajas de texto dejándolas en blanco y el botón cancelar (2.3) que cancela las acciones que se puedan realizar en esta pantalla y la cierra.

"LeerPunto" (3): Una vez creado el trabajo se abrirá una nueva ventana, desde ella se accede a la funcionalidad de lectura de puntos.

Imagen 24: Pantalla de lectura de puntos.

En esta pantalla existen pocos campos editables por el usuario, tan solo puede insertar en cajas de texto editable el número de punto y los comentarios. Mediante un menú desplegable podrá escoger el tipo de punto que va a leer (por defecto "otros puntos") y con los 3 iconos que aparecen con un círculo en la imagen (3.0) podrá añadir una foto al punto. El primero de los tres iconos es una cámara fotográfica, al pulsarlo se llama a la cámara del dispositivo para realizar una fotografía, el segundo icono es representa una imagen, al pulsarlo se llama al explorador de la galería de imágenes del dispositivo para seleccionar una fotografía tomada con anterioridad y el tercer icono con forma de aspa elimina la foto que haya en ese momento.

En cuanto a los botones, sus funcionalidades son las siguientes, el primer botón (3.1) activa la captura de coordenadas, la cual se realizará en función de las opciones de tiempo de lectura que se hayan establecido en las opciones generales del programa. Tras activar el sensor comienza a leer coordenadas y espera a mostrarlas hasta que las coordenadas leídas tengan al menos una precisión menor a 4 metros, cuando consigue una primera coordenada con ese valor mínimo se activa un contador de tiempo interno que permite seguir aproximando las coordenadas hasta que se llegue al tiempo de lectura establecido en las preferencias. Cuando alcanza el objetivo de tiempo de lectura y precisión mínima muestra las coordenadas en pantalla. El siguiente (3.2) es el encargado de almacenar las coordenadas del punto, cuando se pulsa se realizan tres comprobaciones, en primer lugar comprueba que se haya asignado un número al punto, en segundo lugar comprueba que este número de punto no haya sido asignado previamente y para terminar comprueba que se hayan leído coordenadas. Si alguna de estas comprobaciones no se cumple, se mostrará un mensaje de error en pantalla, si se superan todas las condiciones el punto se almacenará, se limpiaran todos los campos menos el campo del número que se autoincrementará en una unidad. El siguiente botón (3.3) se usa para poder detener la lectura de coordenadas en caso de que el usuario así lo decida. El botón de limpiar

formulario (3.4) vacía todos los campos del formulario y el último botón (3.5) cierra la pantalla y vuelve a la pantalla de inicio del programa.

"UltimoTrabajo" (4): Si desde la pantalla principal se pulsa el botón de continuar último trabajo (1.2) y se busca el trabajo con la última fecha de modificación, el cual se abre en una nueva pantalla:

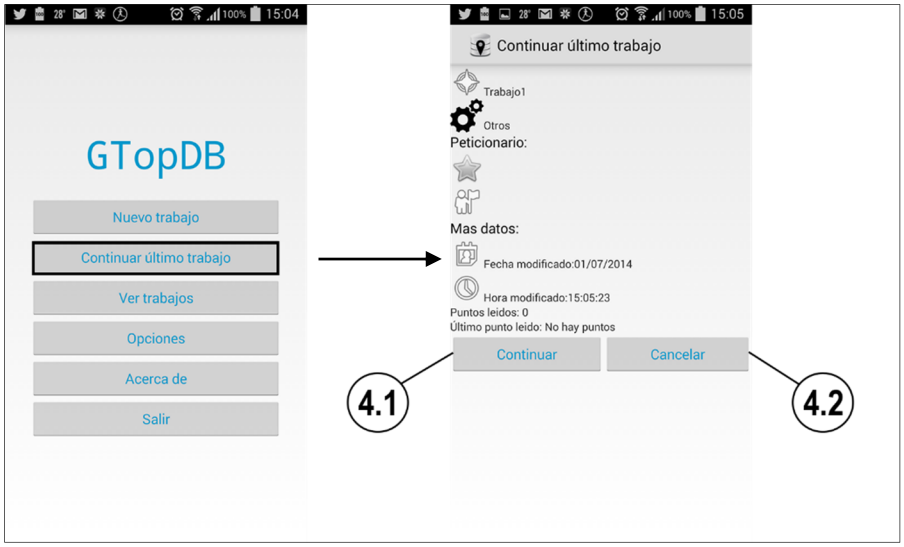


Imagen 25: Pantalla de continuar último trabajo.

Una vez abierta esta pantalla se mostrarán los datos del último trabajo modificado. Desde esta pantalla el usuario tan solo tendrá dos opciones, o bien pulsa el botón de continuar (4.1) y accede a la pantalla de lectura de puntos (3), o por el contrario pulsa el botón de cancelar (4.2) y sale a la pantalla anterior. En caso de acceder a la pantalla de lectura de puntos el programa tomará el número del último punto leído, lo incrementará una unidad y con el cubrirá el campo de número de punto para seguir leyendo (el usuario podrá modificarlo si así lo desea).

"ListarTrabajos" (5): Se accede a ella desde "Ver trabajos" (1.3) en la pantalla principal:

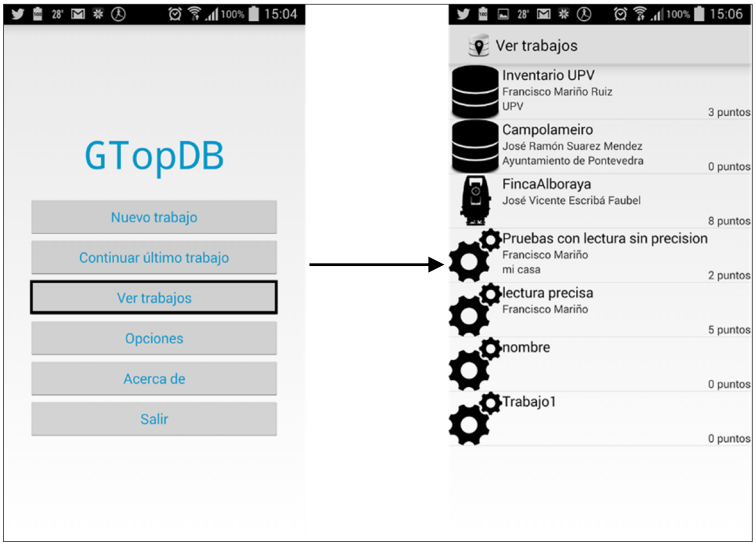


Imagen 26: Pantalla de lista de trabajos.

La clase con la que se implementa esta pantalla es especial ya que lo que realmente se está viendo es un listado de elementos de otra clase visual denominada "AdaptadorCursorTrabajos", esta clase forma cada uno de los objetos que vemos en el listado:

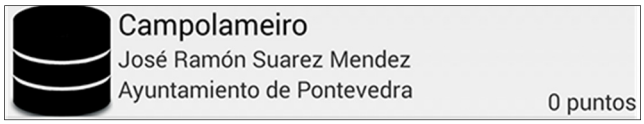


Imagen 27: Detalle de lista de trabajos.

El funcionamiento es el siguiente, la clase "ListarTrabajos" llama a la clase "AdaptadorCursorTrabajos", la cual a su vez llama a la clase de negociación "Trabajos" que solicita a la base de datos el listado de trabajos. El Adaptador va creando diferentes objetos de tipo vista en el que aparecen el nombre del trabajo, la persona de contacto y la entidad peticionaria si los hubiese, a continuación se los devuelve a la clase de ListarTrabajos que los muestra ordenados en pantalla. En caso de no haber ningún trabajo en la base de datos se mostraría la pantalla vacía con un mensaje avisando de la no existencia de trabajos.

A su vez, cada uno de estos objetos que forman la pantalla es seleccionable, de modo que pulsando en cualquiera de ellos se accede a sus detalles.

"DetallesTrabajo" (6): Al pulsar cualquier trabajo de la lista anterior se abre esta pantalla en la que se muestran todos los detalles del trabajo seleccionado:

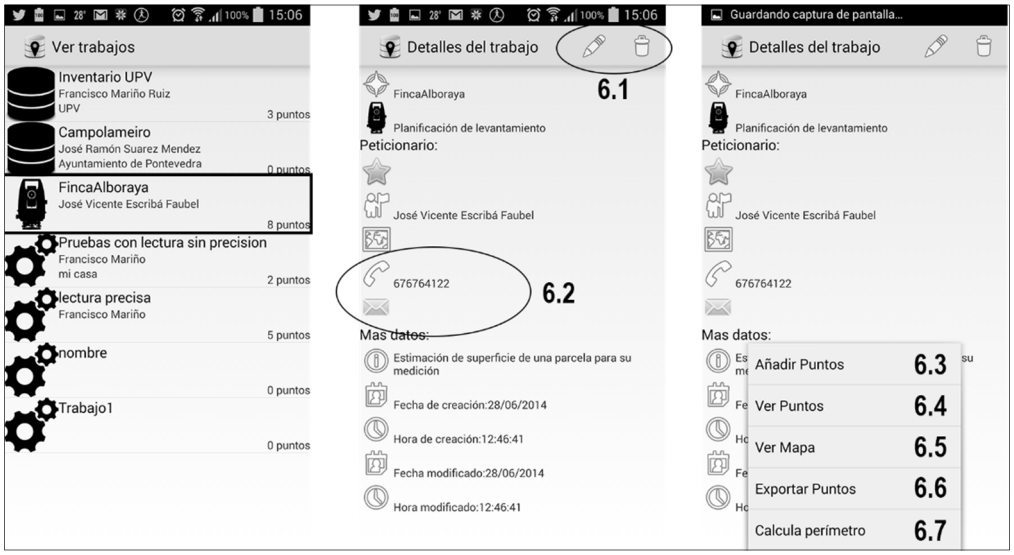


Imagen 28: Detalles de trabajo.

Una vez se abre la pantalla se muestran todos los datos del trabajo. Dentro de la imagen se han resaltado algunas secciones. En la sección 6.1 hay dos iconos, el primero al pulsarlo abrirá la opción de editar los datos del trabajo y el segundo es el icono de borrar el trabajo, que al pulsarlo lanzará un mensaje de alerta de borrado para evitar borrados accidentales. En la sección 6.2 están dos campos de los datos del trabajo que se corresponden al teléfono y al correo electrónico, pulsando el campo del teléfono se podrá realizar una llamada

telefónica al cliente y pulsando el del correo se abrirá la aplicación de correo electrónico del teléfono para poder enviar un correo al cliente.

Además de las opciones ya vistas si se pulsa el botón menú del dispositivo se desplegará el menú que se puede ver en la tercera parte de la imagen (acciones 6.3 a 6.7). Este menú será diferente en caso de que el trabajo que se esté visualizando no sea de tipo "Planificación de levantamiento" ya que no se mostrará la opción de calcular el perímetro (6.7).

Las opciones del menú que se pueden usar son la opción "Añadir puntos" (6.3), que vuelve a abrir la pantalla de "LeerPuntos" (3), la opción de "Ver puntos" (6.4) que muestra los puntos del trabajo, la opción "Ver mapa" (6.5) que muestra los puntos leídos proyectados sobre el WMS de Google Maps, la opción "Exportar puntos" (6.6) que llama a la actividad de exportar los puntos y por último la opción "Calcula perímetro" (6.7) que como ya se ha comentado solo está disponible para los trabajos de tipo "Planificación de levantamiento".

Esta última opción realiza el cálculo de la superficie de la parcela aplicando el método de cálculo de superficies por coordenadas[14]. Aunque la base del cálculo es sencilla, el proceso para realizar el cálculo es bastante laborioso. En primer lugar el concepto teórico del cálculo de superficie por coordenadas. Se parte de un polígono del cual se conocen las coordenadas de todos los vértices:

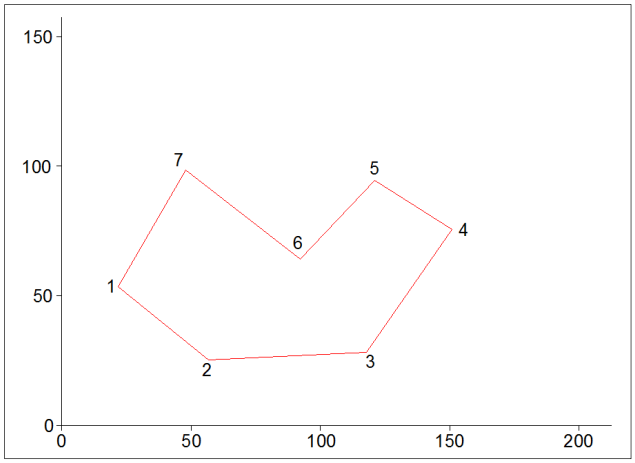


Imagen 29: Cálculo de superficies por el método de coordenadas (1).

Según la teoría de este método, se han de tomar los puntos en el sentido contario a las agujas del reloj, conocidas las coordenadas de cada uno de los puntos, se puede descomponer la figura en una serie de trapecios proyectados sobre el eje de las X, de modo que la suma o resta total de estas figuras dará la superficie de la figura, en primer lugar se restan todos los trapecios que quedan fuera de la figura (los que se forman entre los puntos 1-2, 2-3 y 3-4) y en segundo lugar se suman todos los trapecios que tienen una parte dentro del polígono (los que se forman entre los puntos 4-5, 5-6, 6-7 y 7-1). En la siguiente imagen se puede ver de modo más claro:

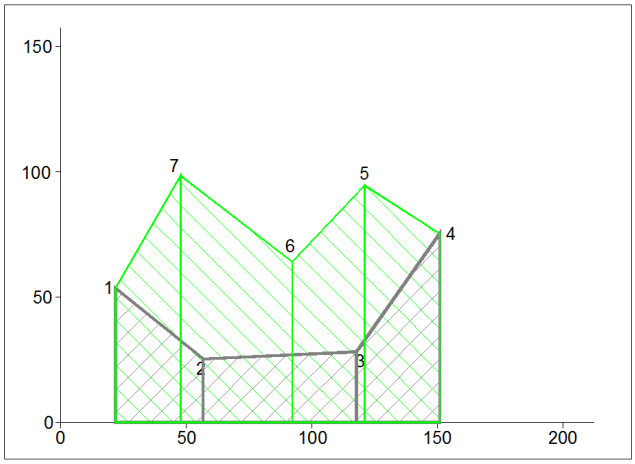


Imagen 30: Cálculo de superficies por el método de coordenadas (2).

Se puede ver que si a la suma de los trapecios verdes se le resta la suma de los trapecios grises se obtiene la superficie del polígono.

A continuación se verá desde el punto de vista del algoritmo diseñado. En primer lugar se hace una consulta a la base de datos pidiendo un listado de puntos que tengan como tipo "perímetro" ordenados por la hora de lectura. El sistema siempre entenderá que los puntos de contorno se han leído ordenados, esto quiere decir que empezando desde cualquiera de los puntos y en cualquier dirección se seguirá el perímetro tal como es y no se realizarán saltos.

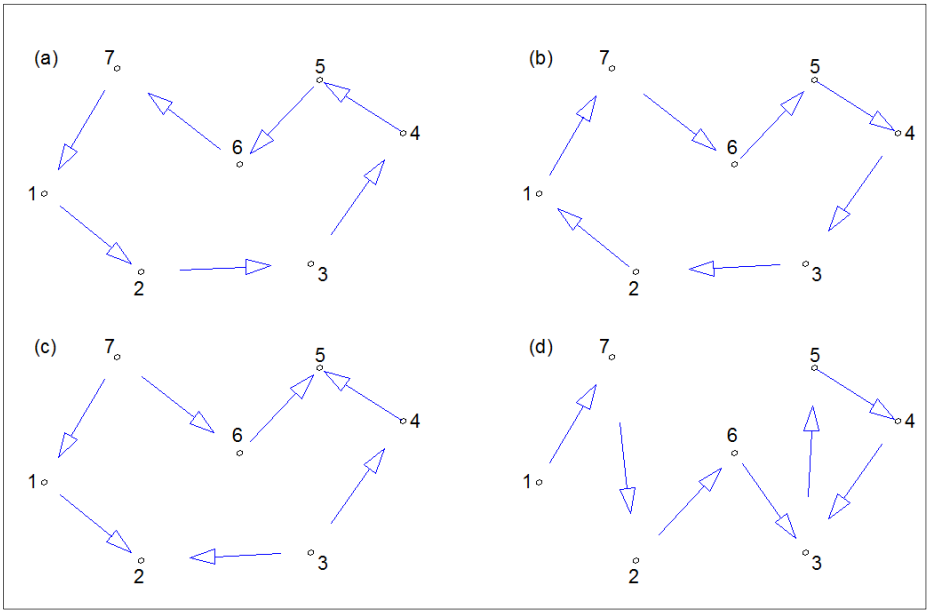


Imagen 31: Ejemplos de lectura de contornos bien leídos (a), (b) y contornos mal leídos (c), (d).

Una vez el programa ha recibido los puntos de la base de datos, los guarda en un vector de puntos para trabajar con ellos. Como es lógico el usuario no siempre habrá leído los puntos en el sentido contrario a las agujas del reloj y comenzando por el punto situado más al oeste, así que el siguiente paso es reordenar los puntos, para ello en primer lugar el programa busca los puntos con mayor y menor coordenada X y almacena en variables su posición en el vector y su número de punto. Partiendo de la base de que el punto de menor coordenada está en la posición

"i" del vector, a continuación el programa busca los puntos de las posiciones "i+1" e "i-1" (si el punto "i" fuese el último del vector cogería como "i+1" el primer punto del vector y si el punto "i" fuese el primero cogería como "i-1" el último del vector). De estos dos puntos tomará las coordenadas Y, y las comparará de modo que el programa decidirá que el punto que tenga menor coordenada Y, será el punto que definirá la dirección del polígono, con lo cual si el punto con menor coordenada Y es el punto que de la posición "i+1", durante el trabajo de campo se leyeron los puntos del perímetro en la dirección contraria de las agujas del reloj (que es lo correcto para el método de cálculo de superficies), mientras que si es el punto que se encuentra en la posición "i-1" se habrá leído en la dirección de las agujas del reloj.

Conocida la posición del punto de menor coordenada X y la dirección de lectura, el vector de puntos pasa por una serie de bucles concatenados que definen un nuevo vector ordenado de tal forma que el primer punto del vector sea el de menor X y el orden de los puntos sea contrario a las agujas del reloj, siendo este vector el que se pasa a la parte de cálculo de superficies. Empieza desde el primer punto haciendo trapecios que almacena con signo negativo hasta que llega a la posición en la que está el punto cuyo número fue almacenado como punto de mayor coordenada X, a partir de donde empieza a generar trapecios con signo positivo. Finalmente se suma y se obtiene la superficie que se muestra al usuario mediante un cuadro de dialogo.

Imagen 32: Superficie calculada.

"EdiciónTrabajo"(7): Pulsando el icono de edición de la vista de trabajo (6.1) se accede a la pantalla de edición del trabajo en la que se pueden modificar los datos básicos del proyecto, es decir nombre, tipo, entidad, persona de contacto, dirección, teléfono, correo electrónico y comentarios. Los campos de fechas de modificación y de inicio no serán accesibles aunque el campo de fecha de modificación se actualizará automáticamente al pulsar el botón guardar (7.1), si se pulsase el botón cancelar (7.2) la pantalla de edición se cerraría sin guardar nada.

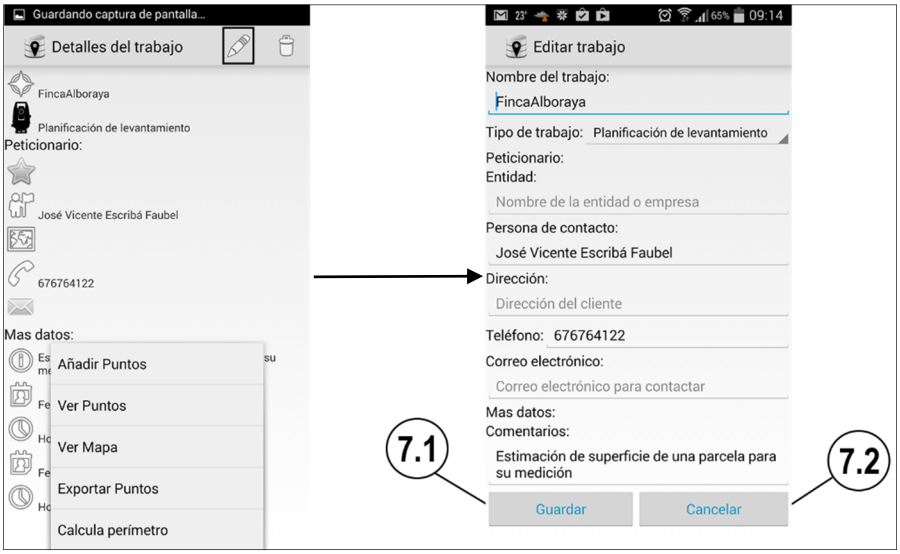


Imagen 33: Pantalla de edición de trabajo.

"ListarPuntos" (8): En el menú de la pantalla en la que se muestran los detalles de un trabajo (6), pulsando la opción de "ver puntos" (6.4) se accede a esta pantalla:

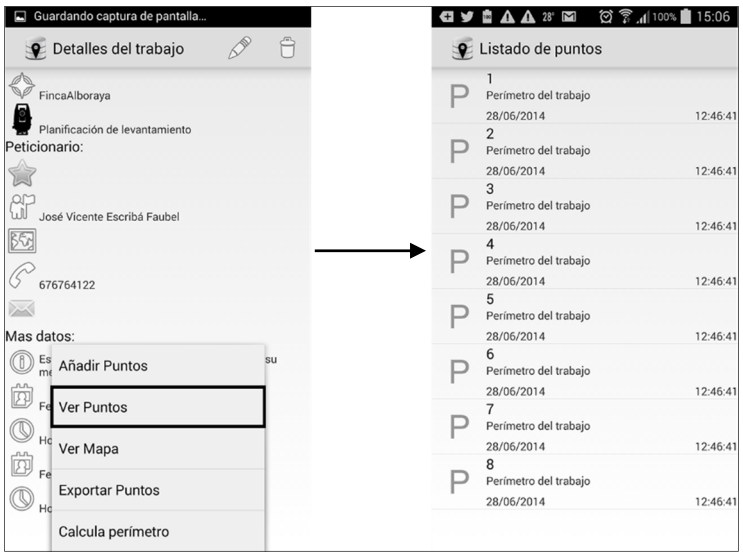


Imagen 34: Pantalla de listado de puntos.

Esta clase es semejante a la clase "ListarTrabajos", es un listado de pequeños objetos de otra clase denominada "AdaptadorCursorPuntos". A través de la clase "Puntos" se hace una petición a la base de datos de todos los puntos que pertenecen al trabajo que se estaba viendo, estos se le pasan al adaptador que se ha creado para definir cada punto. El adaptador construirá los objetos en los que mostrará el icono del tipo de punto, el número de punto, el texto de tipo, la fecha y la hora de lectura. A continuación los devuelve a la clase "ListarPuntos" para que los muestre en forma de lista. En caso de no haber puntos se mostraría un texto de no existencia de datos.

Imagen 35: Detalle de lista de puntos.

Al igual que en el caso del listado de trabajos, pulsando cualquiera de los elementos del listado se puede acceder a los detalles de ese punto.

"DetallesPunto" (9): Una vez pulsada cualquiera de los puntos del listado anterior se abre una nueva ventana en la que se pueden ver todos los detalles del punto.

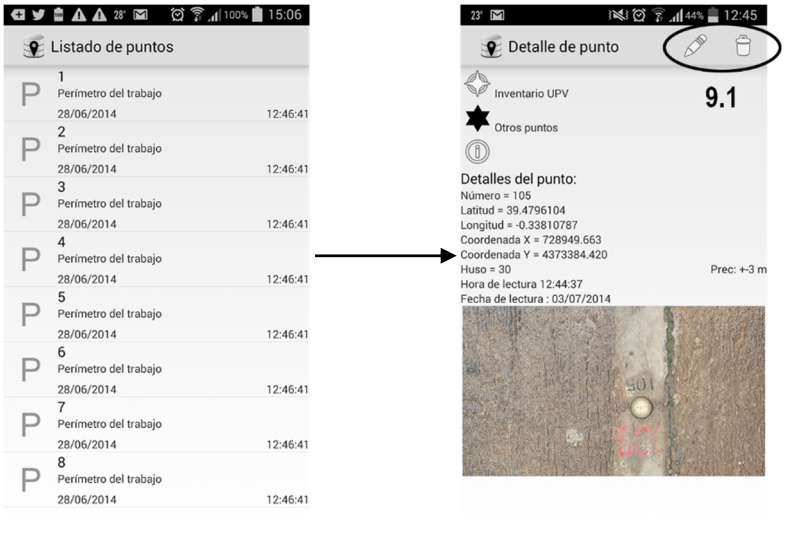


Imagen 36: Pantalla de detalles de punto.

Como se puede ver en la imagen adjunta, se muestran todos los detalles del punto incluida la imagen en caso de que la haya. Esta pantalla solo implementa dos controles (9.1), el primero de ellos lanza la pantalla de edición del punto y el segundo de ellos lanza un mensaje de confirmación para borrar el punto de la base de datos.

"EdicionPunto" (10): Es lanzada desde el control de edición de "Detalles Punto".

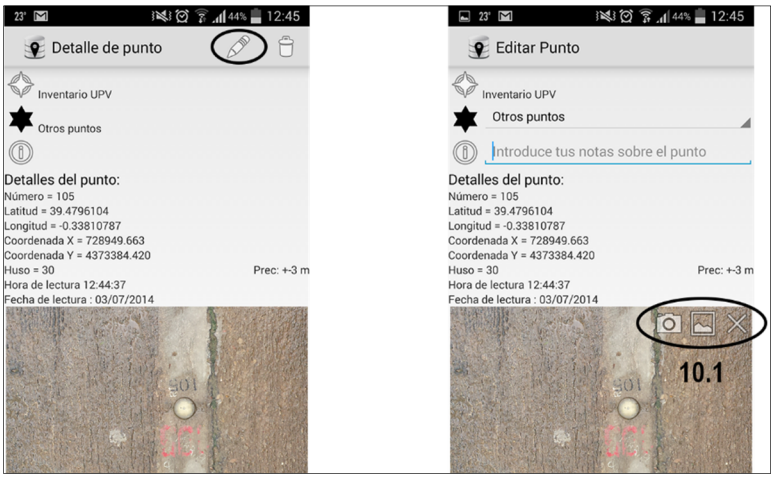


Imagen 37: Pantalla de edición de punto.

Como se puede observar en la imagen es prácticamente idéntica que la pantalla desde la que es lanzada salvo por pequeñas variaciones. El campo de texto en el que está el tipo de punto pasa a ser un menú desplegable para poder cambiar el tipo de punto si así se desea, el campo de comentarios se vuelve editable y sobre la foto aparecen controles para cambiarla o eliminarla (10.1). Desde el botón de menú de esta pantalla se guardan o cancelan los cambios.

"Mapa" (11): Se lanza desde el menú de la pantalla "DetallesTrabajo" y se trata de una pantalla en la que se proyectan los puntos del trabajo sobre el WMS de Google Maps.

Imagen 38: Pantalla de mapas.

El mapa se inicializa en vista de imágenes de satélite, pero se ha colocado un botón en la parte superior (11.1) para que el usuario pueda cambiar a la vista de mapa si así lo decide. este botón cuando se cambia de vista cambia su texto y pasa a ser el botón que hace la función para volver a vista satélite (11.2). Además de proyectar los puntos como marcadores sobre el mapa, da la posibilidad de pulsarlos para obtener información sobre ellos, se trata del número de punto y del tipo de punto. Si se pulsa una segunda vez el programa llamará a la ventana "DetallesPunto" para ver todos los detalles del punto. Los marcadores que se ven sobre el mapa se basan en los marcadores de grupo que se definen en la tabla 6 de esta memoria.

Desde la tecla menú del dispositivo se accede a un menú con una única opción "Más información" que al pulsarla mostrará sobreimpresionada una pantalla que explica que significa cada uno de los marcadores (tabla 6).

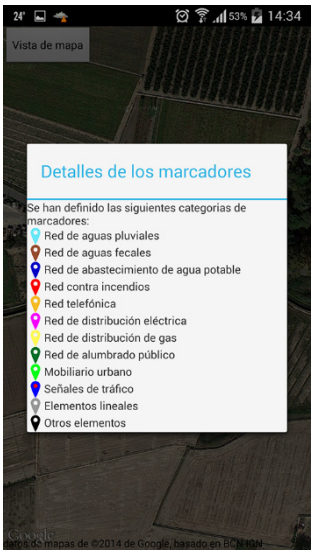


Imagen 39: Detalles de los marcadores.

"Exportar" (12): Es la última pantalla a la que se puede acceder desde la vista de detalles de trabajo.

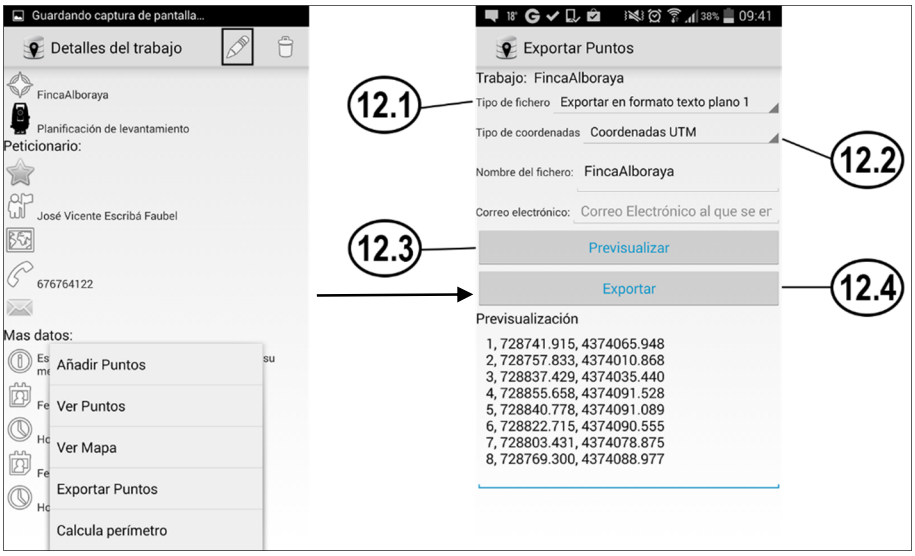


Imagen 40: Pantalla de exportación de puntos.

En esta pantalla se aprecian varios controles, en primer lugar un menú desplegable (12.1) que permite seleccionar el tipo de exportación que se va a realizar, se dispone de los siguientes:

- *Texto plano 1*. Es un fichero de texto separado por comas que contiene el número de punto, la coordenada X y la coordenada Y. Un ejemplo sería:

  1, 728741.915, 4374065.948

- *Texto plano 2*. Fichero de texto separado por comas que contiene el número de puntos, la coordenada X, la coordenada Y, el huso y tipo de punto. Un ejemplo sería:

  1, 728741.915, 4374065.948, 30, Perímetro del trabajo.

- *Texto plano 3.* Fichero de texto separado por comas que contiene todos los datos del punto, es decir número de punto, coordenada X, coordenada Y, huso, precisión de la lectura, tipo de punto, fecha de lectura, hora de lectura, comentario y Uri de la foto. Un ejemplo sería:

  1, 728741.915, 4374065.948, 30, 10, Perímetro del trabajo, 28/06/2014, 12:46:41, Perímetro de parcela, file:///storage/emulated/0/img_1404384264.jpg

- *XML 1.* Fichero con formato de lenguaje de marcas XML, contiene los mismos datos que el formato "texto plano 1". Un ejemplo:

  <trabajo name = FincaAlboraya>
  <punto id = 1>
  <x>728741.915</x><y>4374065.948</y>
  </punto>
  …
  </trabajo>

- XML 2. Contiene los mismos datos que el fichero "texto plano 2", pero en formato XML. Un ejemplo:

  <trabajo name = FincaAlboraya>
  <punto id = 1>
  <x>728741.915</x><y>4374065.948</y><huso>30</huso><tipo>Perímetro del trabajo</tipo>
  </punto>
  …
  </trabajo>

- XML 3. Es el equivalente al fichero "texto plano 3". Un ejemplo:

  <trabajo name = FincaAlboraya>
  <punto id = 1>
  <x>728741.915</x><y>4374065.948</y><huso>30</huso><precision>10</precision><tipo>Perímetro del trabajo</tipo><dia>28/06/2014</dia><hora>12:46:41</hora><comentario>Perímetro parcela</comentario><foto>file:///storage/emulated/0/img_1404384264.jpg</foto>
  </punto>
  …
  </trabajo>

Debajo del menú desplegable para seleccionar el tipo de fichero a exportar, hay otro menú desplegable (12.2) para seleccionar el tipo de coordenadas, por defecto están preseleccionadas las coordenadas UTM, pero el usuario tiene la opción de seleccionar las coordenadas geográficas. Si se seleccionan las coordenadas geográficas los tipos de exportación serán los mismos, pero se sustituirá la coordenada X por la latitud y la coordenada Y por la longitud.

Otro control que posee la pantalla es el botón "Previsualizar" (12.3), con este botón el usuario podrá ir viendo los distintos formatos de exportación para ver el que más se adecue a sus necesidades antes de exportarlo. El último botón es el botón "Exportar" (12.4), en un primer momento se había pensado en la idea de exportar los datos a un fichero de texto plano y este enviarlo al correo electrónico que el usuario decidiese, pero por las características de seguridad de Android no se permite el envío de archivos adjuntos desde programas de manera directa y confiable, por lo tanto se decidió que una buena manera de enviar los datos sería escribiéndolos en el cuerpo del mensaje y enviarlos como mensaje, una vez en el ordenador, el usuario ya los

trataría según sus necesidades. Por tanto se ha puesto una caja de texto editable en la que se pondrá una dirección de correo electrónico, si el usuario ha definido una dirección en las preferencias generales del programa esa será la dirección que aparecerá por defecto para que se le envíen los datos exportados, aunque también podrá editarla sin problema. Al pulsar el botón de exportar, si se hay definida una dirección de correo electrónico se lanzará el gestor de correo del dispositivo móvil para enviar el mensaje, si no está definida dará un aviso de que falta dicha dirección.

"Opciones" (13): Son las preferencias generales del programa y se accede a ellas desde la pantalla principal del programa.

Imagen 41: Pantalla de opciones.

Tan solo se han generado dos opciones, por un lado la opción donde se escoge el tiempo de lectura de punto que realizará el programa cuando esté en la pantalla de lectura (3), se trata de un menú desplegable donde los valores que se pueden escoger son (en segundos): 0, 1, 2, 3, 4, 5, 10, 15, 20, 30 y 60. La segunda opción es el correo electrónico que se usará como correo por defecto en la pantalla de exportación (12).

En principio se pretendía integrar en estas preferencias los tipos de ficheros de exportación, pero tras definir todos los tipos de ficheros, se llegó a la conclusión de que la mejor opción era permitir al usuario escoger el formato deseado desde la propia pantalla de exportar (12) puesto que así podría previsualizarlos y decidir cuál es el que se ajusta más a sus necesidades.

"AcercaDe" (14): Es una pantalla meramente informativa a la que se accede desde la pantalla principal del programa (1). La podemos ver en la siguiente imagen:

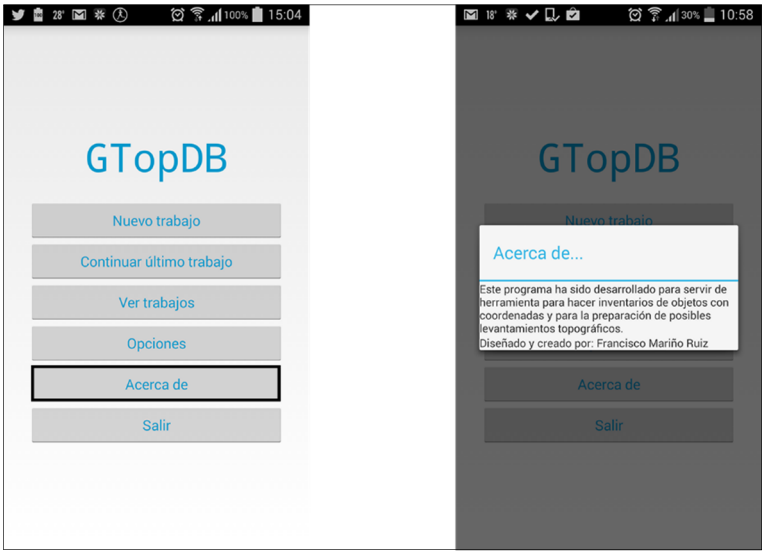


Imagen 42: Pantalla "Acerca de".

Se puede ver que tan solo se trata de un mensaje informativo sobreimpresionado en la pantalla principal.

3.5. Esquema general del programa.

En la siguiente página se adjunta una imagen en la que se ha intentado mediante un gráfico hacer un esquema general de las interacciones del programa. En dicho esquema se pueden apreciar las conexiones entre las pantallas que se han explicado en el apartado anterior y las llamadas que hacen a los distintos métodos de las clases de negociación para que estos métodos trabajen con la base de datos.

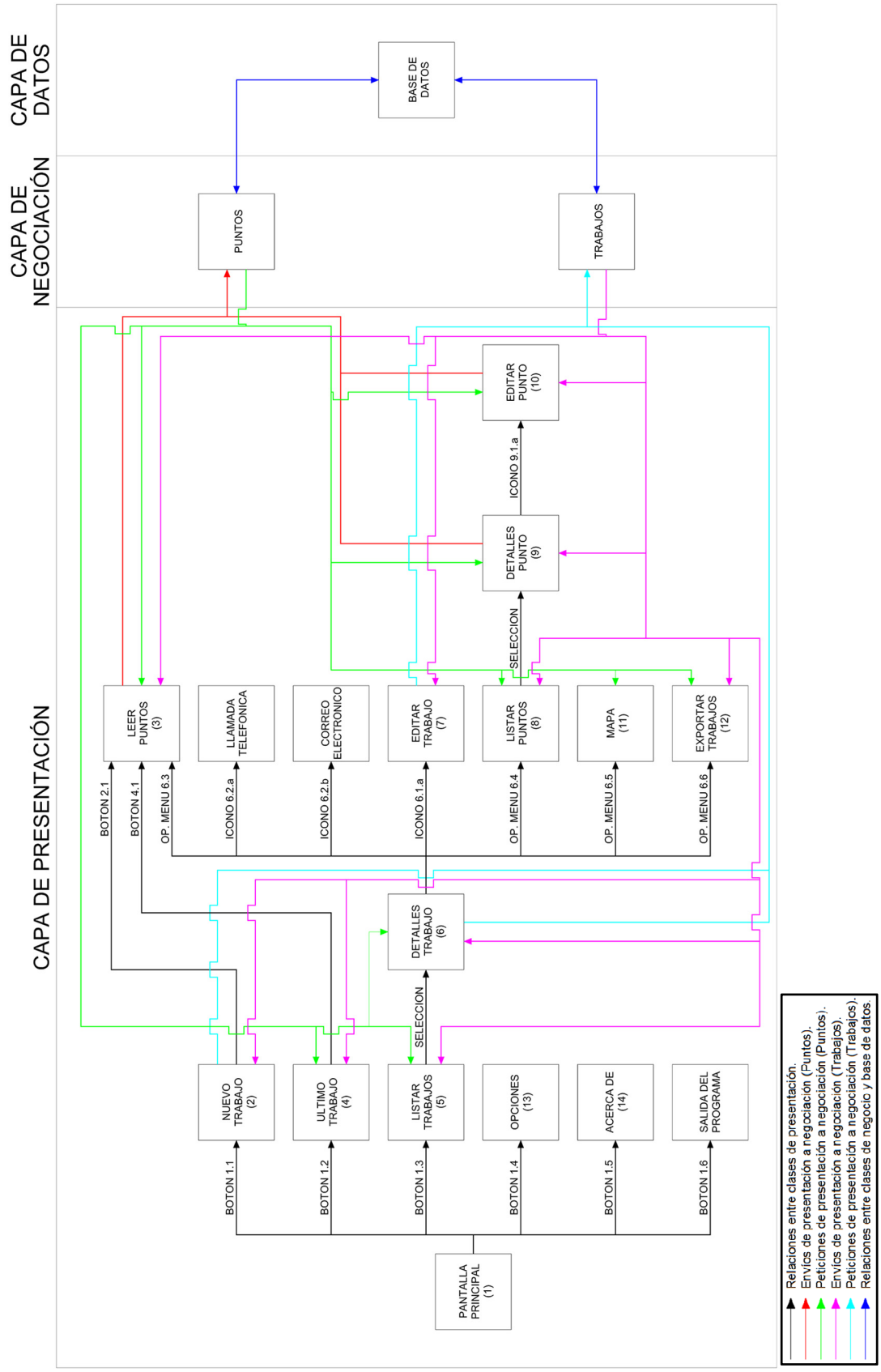


Imagen 43: Esquema general del programa.

4. Fase de pruebas.

Una vez terminada la fase de diseño y programación de la aplicación llegó el momento de realizar algunas pruebas de funcionamiento. Estas pruebas se basaron en primer lugar en la funcionalidad de los algoritmos del programa y en las lecturas del sensor GNSS, en la segunda fase se pretendía comprobar la calidad de las coordenadas obtenidas en las lecturas realizadas por el sensor GNSS.

4.1. Primera fase: Funcionamiento del programa.

Las primeras pruebas fueron realizadas en el emulador que proporciona eclipse, este entorno de desarrollo proporciona un completo simulador de dispositivos móviles con Android en el que se pueden hacer múltiples pruebas, una de las grandes ventajas es poder simular un sensor GNSS que envíe coordenadas al dispositivo para que este las reciba y las procese adecuadamente, esto permite poder ir haciendo pruebas a medida que se desarrolla la aplicación sin necesidad de hacer salidas al campo para probar cada mejora. Estas pruebas fueron realizadas durante todo el proceso de desarrollo y en ellas se buscaba probar que todos los algoritmos programados funcionaban tal como se habían diseñado, que el procesado y transformación de las coordenadas era correcto y que no había errores que cerrasen la aplicación. De este modo se fueron haciendo las pruebas en paralelo al desarrollo y así se corrigieron todos los problemas que fueron surgiendo.

Tras probarlo en el simulador de Eclipse, era necesario probarlo en un dispositivo real que confirmase la usabilidad y la fluidez del programa. Para las pruebas se ha utilizado un Smartphone cuyo modelo es Samsung Galaxy S4. Las primeras pruebas de funcionamiento real fueron realizadas el día 28 de junio de 2014 en la azotea de un edificio del centro de Valencia. Además de la funcionalidad del programa se hicieron pruebas en las que se medían los tiempos de inicialización del sensor GNSS, tiempo necesitado por el sensor para alcanzar las precisiones requeridas y el resultado de proyectar estas coordenadas sobre Google Maps, puesto que la funcionalidad de mapas no es posible probarla sobre el simulador.

Se ha hecho un primer test de inicialización y aproximación con 12 muestras. La metodología ha sido la siguiente, en primer lugar se activaba el GNSS, a continuación se pulsaba el botón “Leer coordenadas” (3.1) de la pantalla “LeerPuntos”, se cronometraba el tiempo de obtención de la primera posición y el tiempo en realizar la lectura con precisiones menores de 4 metros, una vez leídas las coordenadas se desactiva el GNSS desde las opciones del móvil y se reinicia el proceso. Se observa que en algunas ocasiones el GNSS deja de funcionar y de leer coordenadas, aunque pasados unos segundos suele volver a inicializar y leer, estos intentos aparecen en la tabla como puntos sin tiempo de aproximación ya que al reiniciar la inicialización se decidió no tener en cuenta el tiempo. Los resultados obtenidos se pueden ver en la siguiente tabla:

Número de muestra	Hora de lectura	Tiempo de inicialización (s)	Tiempo de aproximación a la precisión (s)	Tiempo entre contacto y aproximación (s)
1	17:39	4	-	-
2	17:41	3	14	11
3	17:42	3	13	10
4	17:44	10	-	-
5	17:46	5	14	9
6	17:50	4	20	16
7	17:51	4	36	32
8	17:52	2	7	5
9	17:53	4	19	15
10	17:54	3	14	11
11	17:55	3	13	10
12	17:56	7	22	15
Media	-	4,33	17,2	13,4

Tabla 7: Pruebas de tiempos de lectura.

Se puede ver en la tabla que el tiempo de inicialización es muy bueno, en torno a los 4 segundos, esto en gracias a la integración del A-GPS que acelera mucho este parámetro. Una vez obtenida la inicialización la media de tiempo necesario para obtener una precisión aceptable es de 13,4 segundos, si bien la mayoría de los tiempos se sitúan cercanos a esa media se encuentran algunos tiempos dispares como son el tiempo de 5 segundos que es muy rápido y el de 32 que aunque no es excesivamente lento es más del doble de la media.

En esa misma sesión se hizo otro test, en esta ocasión se trata de un test de tiempo de exposición. La metodología fue la siguiente, se activa el sensor GNSS desde las opciones del móvil, se inicia el programa y en las opciones se establece el tiempo de lectura, se sitúa el dispositivo en un punto de coordenadas desconocidas en el que permanecerá durante toda la sesión y se procede a la lectura, tras leer cada punto se sale al menú principal para ir a las opciones y cambiar el tiempo de lectura. Los resultados son los siguientes:

Número de punto	Hora de lectura	Tiempo de exposición (s)	Coordenada X (m)	Coordenada Y (m)	Distancia entre lecturas (m)
1	19:24:04	5	724835,934	4371959,125	-
2	19:25:38	10	724836,102	4371958,431	0,714
3	19:27:48	20	724835,661	4371958,314	0,456
4	19:29:44	30	724836,250	4371957,620	0,910
5	19:32:24	60	724835,809	4371956,844	0,892

Tabla 8: Lecturas a distintos tiempos de exposición.

Se puede ver en la tabla como van variando las coordenadas entre las distintas lecturas moviéndose todas ellas en un radio de apenas dos metros. Como en este caso no había coordenadas con las que comparar las lecturas simplemente se han proyectado sobre el WMS de Google Maps, con esto se cumplirá un doble objetivo, por un lado se verá el funcionamiento de la integración de los mapas en la aplicación y por otro lado se verá los puntos en torno a la azotea desde la que se estaba leyendo. En la siguiente imagen, a pesar de que el color de los marcadores no se distingue bien se puede ver como los puntos parecen converger a medida que se aumenta el tiempo de lectura, pero sin embargo no se acercan al punto real que estaría más adentrado en la azotea.

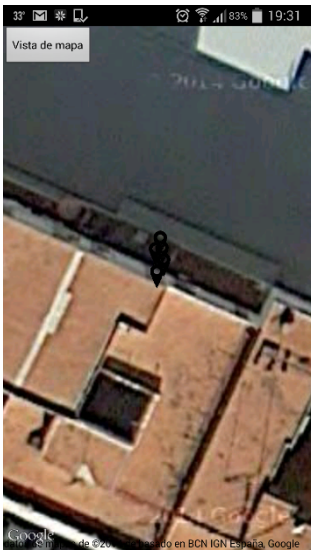


Imagen 44: Pruebas de tiempos de exposición.

4.2. Segunda fase: Calidad de las coordenadas.

En esta segunda fase de pruebas se busca estimar la calidad de las coordenadas ofrecidas por el sensor GNSS del dispositivo móvil comparándolas con puntos de coordenadas conocidas. Para realizar estas lecturas se necesitaba leer puntos de coordenadas conocidas, por ello se decidió leer algunas de las bases de calibración de la Universidad Politécnica de Valencia. Se hicieron las lecturas el día 2 de julio pero las bases de calibración estaban en uso, por lo que solo se pudieron leer dos de ellas. Para completar el test, se leyeron dos vértices de la red de control del Departamento de Ingeniería Cartográfica y Fotogrametría en la zona misma zona. En concreto se han leído las bases de calibración 1 y 2 y los vértices 201 y 209 de la red de control. En la siguiente imagen se puede ver la situación de estos puntos:

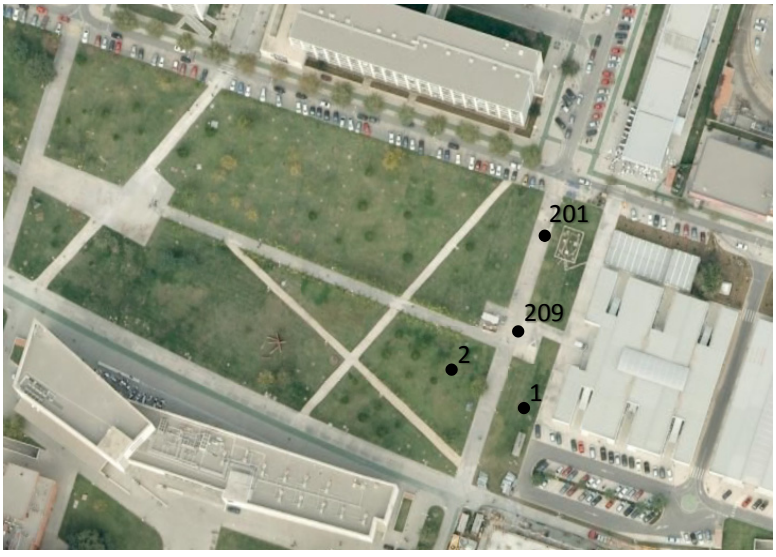


Imagen 45: Distribución de los puntos leídos.

Cabe mencionar que las coordenadas de las que se dispone están en el sistema de referencia ETRS89, mientras que las que lee el sensor GNSS están en el sistema de referencia

WGS84, sin embargo la diferencia entre ambos sistemas de coordenadas es mínima, en torno a un centímetro, por lo que a pesar de tratarse de coordenadas en sistemas de referencia diferentes, pueden ser comparadas en este estudio en el que solo se busca una estimación de la proximidad entre coordenadas reales y coordenadas leídas. La metodología fue la siguiente, en primer lugar se inicia el GNSS desde las opciones del móvil, se abre la aplicación y en las opciones se establece el primer tiempo de lectura, a continuación se crea un trabajo y se comienza la lectura de puntos. Se han leído puntos con 0, 1, 5, 10, 30 y 60 segundos de exposición. El primer punto leído fue el punto 201:

Número de punto	Hora de lectura	Tiempo de exposición (s)	Coordenada X (m)	Coordenada Y (m)	Distancia a las coordenadas reales (m)
Coordenadas reales			729003,405	4373391,069	-
2010	11:49:43	0	729002,679	4373391,975	1,161
2011	11:54:04	1	729005,238	4373394,179	3,610
2012	11:55:33	5	729005,554	4373393,578	3,304
2013	11:57:35	10	729005,268	4373393,812	3,316
2014	12:03:03	30	729004,933	4373393,632	2,984
2015	12:06:28	60	729004,291	4373394,074	3,133

Tabla 9: Lecturas del punto 201.

El siguiente fue el punto 209:

Número de punto	Hora de lectura	Tiempo de exposición (s)	Coordenada X (m)	Coordenada Y (m)	Distancia a las coordenadas reales (m)
Coordenadas reales			728997,012	4373372,126	-
2090	12:08:56	0	728998,636	4373368,579	3,901
2091	12:09:57	1	728997,341	4373369,334	2,811
2092	12:11:40	5	728996,800	4373370,386	1,753
2093	12:13:27	10	728996,403	4373371,028	1,256
2094	12:17:11	30	728996,364	4373371,847	0,706
2095	12:20:43	60	728996,898	4373373,005	0,886

Tabla 10: Lecturas punto 209.

A continuación se leyó el punto 2:

Número de punto	Hora de lectura	Tiempo de exposición (s)	Coordenada X (m)	Coordenada Y (m)	Distancia a las coordenadas reales (m)
Coordenadas reales			728974,065	4373360,165	-
3000	12:25:28	0	728978,510	4373361,734	4,714
3001	12:27:05	1	728978,159	4373361,797	4,407
3002	12:27:59	5	728978,020	4373361,146	4,075
3003	12:28:58	10	728977,657	4373361,184	3,734
3004	12:30:51	30	728977,397	4373361,231	3,498
3005	12:32:43	60	728979,102	4373361,574	5,230

Tabla 11: Lecturas punto 2.

Se terminó con el punto 1:

Número de punto	Hora de lectura	Tiempo de exposición (s)	Coordenada X (m)	Coordenada Y (m)	Distancia a las coordenadas reales (m)
Coordenadas reales			728999,070	4373346,727	-
3010	12:35:59	0	729001,319	4373349,533	3,596
3011	12:37:42	1	728999,730	4373346,309	0,781
3012	12:38:50	5	728999,783	4373346,203	0,885
3013	12:41:52	10	728998,931	4373343,796	2,934
3014	12:43:29	30	728998,671	4373343,578	3,174
3015	12:45:41	60	728997,980	4373344,302	2,659

Tabla 12: Lecturas punto 1.

A continuación, antes de interpretar los resultados de la prueba, se expondrán unas imágenes en las que se ve la distribución espacial de los puntos leídos por el sensor GNSS del dispositivo. Visualización del punto 201:

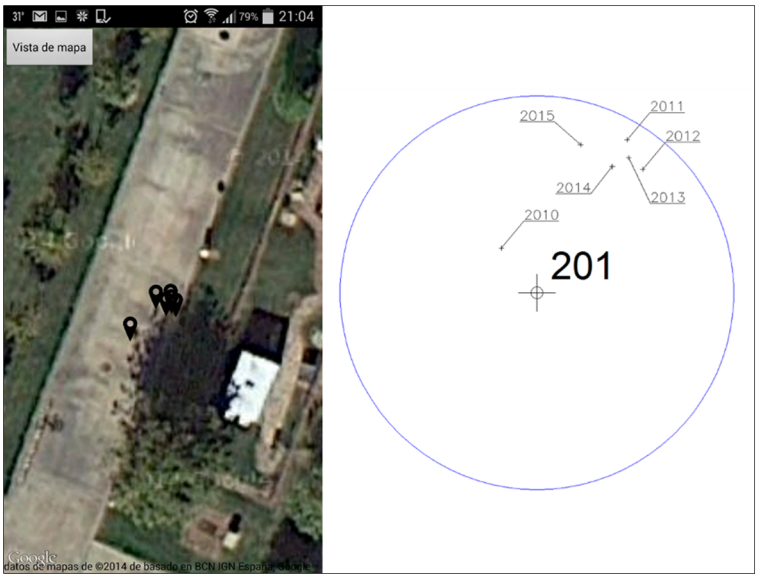


Imagen 46: Punto 201.

A continuación el punto 209:

Imagen 47: Punto 209.

El punto 2:

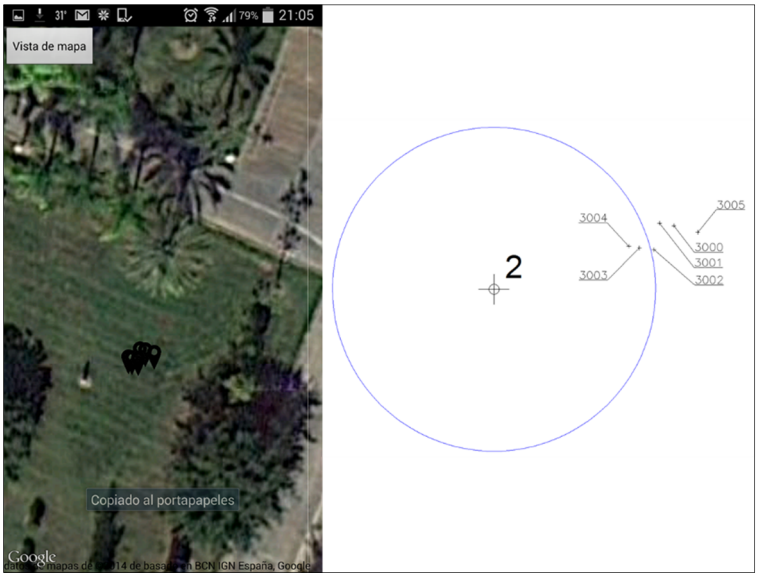


Imagen 48: Punto 2.

Y el punto 1:

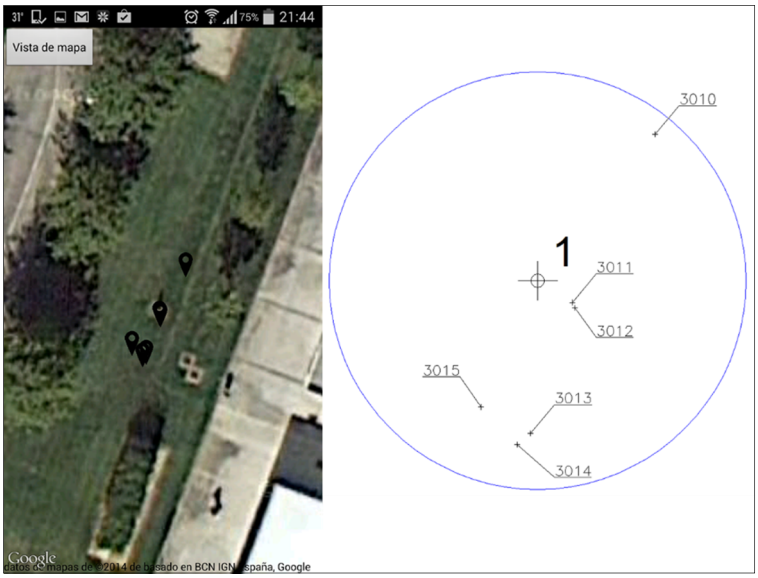


Imagen 49: Punto 1.

Cuando se explicó el funcionamiento del programa, más concretamente cuando se explicó el algoritmo de la pantalla “LeerPunto” se comentó que el programa comenzaba tener en cuenta las lecturas cuando la precisión ofrecida por el sensor GNSS era menor de cuatro metros, cabe decir que según la documentación proporcionada por el fabricante del dispositivo la precisión del GNSS es ± 3 metros con una fiabilidad del 68%. Como se han pedido lecturas cuando las precisiones hayan bajado de cuatro metros, esa es la referencia que se ha tomado para medir la distancia entre el punto real y los leídos. En las imágenes, además de la vista sobre Google Maps, vemos un detalle del punto real con los puntos leídos proyectados sobre una circunferencia de 4 metros. En el punto 201 se ve una lectura bastante precisa pero poco exacta

pues los puntos se distribuyen en una sola zona de la circunferencia alejada de la zona del punto real, sin embargo todos los puntos están por debajo de los cuatro metros con lo que se puede dar como buena la lectura. El punto 209 representa lo que en teoría se esperaba, que a medida que se fuese iterando durante más segundos la posición esta fuese acercándose más al punto real, en este caso vemos que las últimas lecturas realizadas son con una diferencia de menos de un metro con respecto al punto real, lo cual está indicando una gran calidad de los puntos leídos. En el punto 2 vemos una serie de lecturas precisas pero poco exactas y bastante alejadas del punto real, de hecho la mayoría de los puntos están a más de cuatro metros estando el más alejado está a 5,230 metros. Recordando las especificaciones del dispositivo, tenía una precisión de ± 3 metros al 68% de fiabilidad, por tanto para una fiabilidad del 95% habría que aumentar la precisión a ± 6 metros por lo que se podrían considerar buenas las coordenadas que obtenemos del sensor. Por último, en el punto 1 vemos que las lecturas no son ni exactas ni precisas, hay puntos con buenas coordenadas pero no son necesariamente los de más tiempo de exposición.

Si se promedian todas las distancias que se han obtenido en el estudio, se obtiene una media de 2,854 metros de diferencia entre coordenadas. Sin embargo este dato por sí mismo no es muy significativo, pues se están mezclando diferentes tiempos, así que a continuación se expone una tabla con las distancias medias en función del tiempo de exposición:

Tiempo (s)	Punto 201 (m)	Punto 209 (m)	Punto 2 (m)	Punto 1 (m)	Media (m)	Desv. Típica (m)
0	1,161	3,901	4,714	3,596	3,343	1,529
1	3,610	2,811	4,407	0,781	2,902	1,557
5	3,304	1,753	4,075	0,885	2,504	1,448
10	3,316	1,256	3,734	2,934	2,810	1,086
30	2,984	0,706	3,498	3,174	2,590	1,274
60	3,133	0,886	5,230	2,659	2,977	1,786

Tabla 13: Comparativa por tiempos de exposición (1).

Si se da un primer vistazo a esta tabla se podría llegar a la conclusión de que aumentando el tiempo de exposición en un punto no se consiguen mejoras significativas en la distancia, en la siguiente imagen podemos ver el resultado gráficamente:

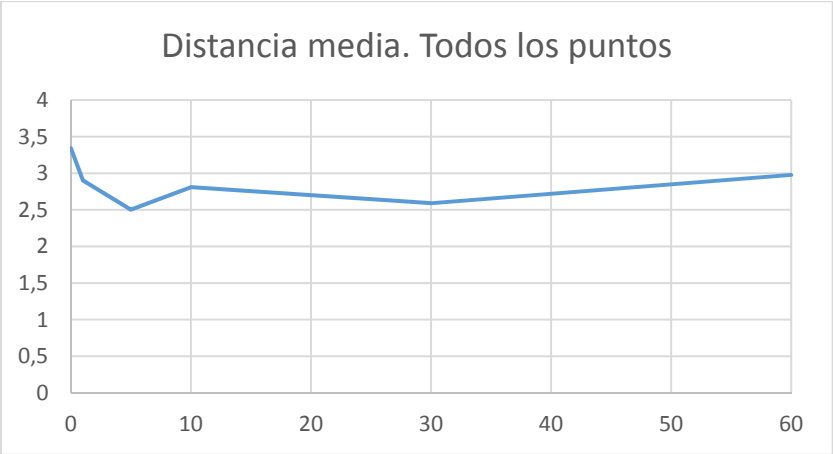


Imagen 50: Evolución distancias en función del tiempo (1).

Antes de sacar conclusiones con estos datos se va a eliminar de la serie el punto 2, ya que es el menos exacto y podría estar introduciendo algún error en el estudio. En primer lugar se recalcularán las medias para los distintos puntos:

Tiempo (s)	Punto 201 (m)	Punto 209 (m)	Punto 1 (m)	Media (m)	Desv. Típica (m)
0	1,161	3,901	3,596	2,886	1,502
1	3,610	2,811	0,781	2,401	1,458
5	3,304	1,753	0,885	1,980	1,225
10	3,316	1,256	2,934	2,502	1,096
30	2,984	0,706	3,174	2,288	1,374
60	3,133	0,886	2,659	2,226	1,184

Tabla 14: Comparativa por tiempos de exposición (2).

En esta ocasión ya se puede observar como a medida que aumenta el tiempo de exposición disminuye la distancia. En la gráfica también se puede observar este comportamiento:

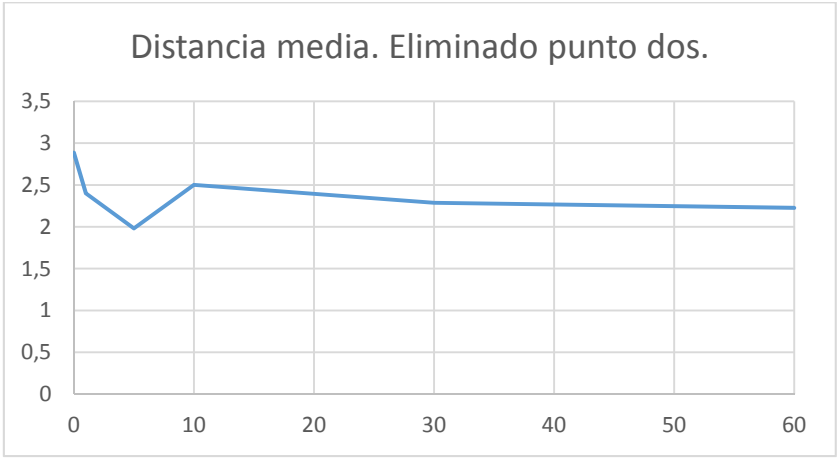


Imagen 51: Evolución distancias en función del tiempo (2).

A la vista de los resultados obtenidos se puede observar una rápida convergencia dentro de los primeros cinco segundos y tras ello una pérdida de proximidad y finalmente una tendencia de convergencia lenta. Esto es debido a que existe una lectura a que en el punto 1, las lecturas a 1 y 5 segundos son anormalmente buenas. Como la muestra que se ha tomado no es muy grande no se pueden sacar conclusiones deterministas pero se puede decir que el tiempo de exposición mejora la calidad de las coordenadas pero que sin embargo la mejora no es tan alta como cabría esperar. Esto tiene una explicación lógica, cuando se diseñó el algoritmo de lectura de coordenadas, se hizo para que no tuviese en cuenta las coordenadas hasta que las precisiones que el sensor consigue no bajasen de 4 metros, con esto ya se está consiguiendo un tiempo de convergencia previo a que se inicie el contador interno de tiempo de exposición en el punto. Este tiempo previo es variable como ya se ha visto en las pruebas de tiempo de inicialización realizadas anteriormente. Por todo esto se puede decir que si bien las coordenadas mejoran con los tiempos de exposición, esto implica una ralentización del trabajo que se está realizando, con lo cual se ha de buscar un tiempo de exposición que proporciones un buen compromiso entre las mejores coordenadas posibles y el tiempo de trabajo.

5. Líneas de mejora.

En este apartado se pretende comentar las posibles líneas de mejora que se han establecido para la aplicación y que se intentarán implementar en futuras fases de desarrollo y actualización. En este momento, la aplicación desarrollada es funcional, pero hay aspectos que es necesario pulir y funcionalidades que sería bueno incorporar. A continuación se describen algunas de ellas:

- ***Interfaz***: Se ha diseñado una interfaz muy básica para el programa, sería conveniente que en próximas versiones se intentase conseguir un rediseño de la interfaz más atractivo para el usuario. Aunque un nuevo diseño puede mejorar el atractivo de la aplicación, también puede conseguir que la experiencia de usuario se dificulte, por lo que además del diseño ha de cuidarse que no se pierda en ningún momento la facilidad de funcionamiento e interacción con el usuario que da una interfaz básica.
- ***Sistema operativo***: Una vez el programa se haya consolidado sería conveniente realizar versiones para otros sistemas operativos con la finalidad de hacer el programa accesible a la mayor cantidad de usuarios posible.
- ***Mapas***: En la aplicación se usa la cartografía de Google Maps para proyectar los puntos, lo cual en algunas ocasiones no es todo lo útil que a priori puede parecer, en ocasiones las imágenes que proporciona este servidor no son de buena calidad, existen zonas que a medida que se hace zoom aparecen muy pixeladas y otras en las que las imágenes son muy oscuras, por lo que al proyectar los puntos no se aprecia correctamente lo que los marcadores señalan. Por otro lado la vista de mapa de Google no ofrece buena cartografía fuera de los grandes núcleos de población, donde a menudo solo están representadas las vías de comunicación y no existen muchos detalles del resto de elementos. Por ello en próximas versiones podría integrarse algún servidor WMS más a las opciones de visualización, aunque lo ideal sería integrar las funcionalidades de ESRI para que el usuario pueda usar sus propios mapas adaptados a sus necesidades en cada momento.
- ***Gestión de datos***: A la hora de visualizar los datos en la versión actual existen ciertas limitaciones, por ejemplo, cuando se están viendo los puntos existentes en un trabajo se visualiza una lista con todos los puntos que se han leído. Una posible mejora sería dar la opción al usuario de establecer filtros o herramientas de búsqueda para que pueda ver solo los puntos del tipo que le interese o pueda ver solamente un rango de puntos determinado. A su vez sería interesante la posibilidad de disponer de este tipo de filtros a la hora de exportar los puntos desde un trabajo.
- ***Coordenadas***: Las coordenadas que se obtienen están limitadas por la precisión del dispositivo, en el caso del utilizado durante el desarrollo del proyecto ofrece, según las especificaciones del fabricante, precisiones de entre 3 y 4 metros. Una línea de investigación muy interesante sería la integración en el dispositivo móvil

de una antena GPS externa. En 2012 se presentó en la conferencia TopCart el dispositivo “Posify”[(15)].

Imagen 52: Posify.

Este dispositivo consiste en un sistema de posicionamiento y medición GPS con soporte para postproceso, con el que se pueden obtener posiciones centimétricas. Según los fabricantes este aparato puede proporcionar precisiones de 20 a 30 centímetros para medidas en modo cinemático y de 2 a 3 centímetros para medidas en estático, lo que representaría una mejora de calidad impresionante en las medidas de coordenadas y abriría el abanico de posibilidades funcionales de la aplicación. Además de la precisión que promete, “Posify” es totalmente compatible con Smartphone, como se vio en su presentación:

Imagen 53: "Posify" en un Smartphone.

Otra característica que hace que este aparato sea muy interesante para el proyecto que se ha desarrollado es su precio, pues según dijeron los fabricantes durante la presentación del producto saldría al mercado por un precio de alrededor de 395 €.

Resulta evidente que no todas estas mejoras son igual de importantes, sin duda la línea de mejora más importante sería la que se basa en mejorar las coordenadas. Lo siguiente sería ampliar el catálogo de mapas utilizables y la gestión de datos. Las mejoras de interfaz y la migración a otros sistemas operativos son mejoras secundarias.

6. Conclusiones finales.

Para poner punto final a este proyecto se han extraído unas conclusiones finales del trabajo realizado. En primer lugar es conveniente señalar que se han cubierto todos los objetivos planteados al inicio del proyecto.

Durante la realización de este trabajo final de grado, se han adquirido los conocimientos necesarios para desarrollar aplicaciones para dispositivos móviles con sistema operativo Android, se han perfeccionado los conocimientos previos de programación en lenguaje Java y se han desarrollado los conceptos referentes al posicionamiento en el espacio mediante técnicas GNSS.

Gracias a todos estos conocimientos se ha podido desarrollar una aplicación, que haciendo uso de los métodos de geolocalización en dispositivos móviles es capaz de realizar inventarios rápidos de objetos que no necesiten grandes precisiones en su posicionamiento. Se ha logrado una versión funcional de la misma que se ha podido probar para tener una idea de la calidad real de las coordenadas que puede proporcionar.

Por otro lado se ha conseguido adentrarse en el mundo de la programación y de las nuevas tecnologías sin perder la perspectiva geomática, entendiendo la programación como una herramienta para el desarrollo de este trabajo y no como una finalidad. Con ello se ha conseguido una herramienta de trabajo basada en los dispositivos móviles que ya poseemos para facilitar y agilizar los trabajos de campo.

Siguiendo el espíritu geomático del proyecto se han intentado conseguir siempre los mejores resultados en cuanto a la precisión del posicionamiento y se han buscado las mejores soluciones en cada momento, abriendo también posibles futuras líneas de trabajo que puedan mejorar los resultados obtenidos hasta ahora.

Por último, señalar que a medida que el desarrollo ha ido avanzando la idea inicial ha ido variando y creciendo, todo ello siempre con la idea de obtener la mayor eficacia posible con los mínimos recursos. De este modo se han podido cubrir todos los objetivos marcados al inicio e incluso alguno más que al inicio del proyecto no se había planteado.

7. Bibliografía.

(1) Mallick M. Mobile and Wireless Design Essentials. Primera ed. Indianápolis, Indiana (USA): Wiley Publishing, Inc.; 2003.

(2) Interactive Advertising Bureau (IAB). V Estudio Anual IAB Spain Mobile Marketing. 2013 25/09/2013;V.

(3) Wikipedia. Sistema operativo móvil. 2009; Available at: http://es.wikipedia.org/wiki/Sistema_operativo_móvil. Accessed 05/29, 2014.

(4) Gironés JT. Comparativa con otras plataformas. 2014; Available at: http://www.androidcurso.com/index.php/tutoriales-android-fundamentos/31-unidad-1-vision-general-y-entorno-de-desarrollo/98-comparativa-con-otras-plataformas. Accessed 04/15, 2014.

(5) Kantar Worldpanel ComTech. Smartphone OS market share. 2014; Available at: http://www.kantarworldpanel.com/smartphone-os-market-share/. Accessed 06/01, 2014.

(6) Android Developer. Dashboards. 2014; Available at: http://developer.android.com/about/dashboards/index.html. Accessed 06/04, 2014.

(7) Argelaga B. Aplicación de gestión de información geolocalizada en Android. Proyecto final de carrera presentado en la Facultad de Informática de Barcelona de la Universidad Politécnica de Cataluña. 2011.

(8) Berné JL, Anquela AB, Garrido N. GPS : fundamentos y aplicaciones en geodesia y topografía. Valencia: Universitat Politècnica de València; 2013.

(9) Open Geospatial Consortium. Web Map Server. 2013; Available at: http://www.opengeospatial.org/standards/wms. Accessed 06/10, 2014.

(10) Pastor O, Pelechano V, Insfrán E, Gómez J. Generación automática de prototipos en entornos Internet/Intranet a partir de modelos conceptuales. Computación y Sistemas 1999;3:38-49.

(11) Reynoso CB. Introducción a la Arquitectura de Software. Artículos seleccionados. Universidad de Buenos Aires 2004.

(12) Roberto MM. Java 7. Madrid: Anaya Multimedia; 2011.

(13) Juan MB. Geodesia superior. Madrid: Instituto Geográfico Nacional; 2008.

(14) Porres MJ. Cálculo de superficies. Método de coordenadas cartesianas. 2009 2009-12-21T13:12:39Z.

(15) GeoFumadas. Posify, precisión centimétrica GPS de bajo costo. 2012; Available at: http://www.geofumadas.com/posify-precisin-centimtrica-gps-de-bajo-costo/. Accessed 06/28, 2014.