

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

**POSICIONAMIENTO EN
INTERIORES BASADO EN
DISPOSITIVOS MÓVILES**

Ingeniería de Telecomunicación

Miguel Ruiz Cuesta
Abril 2012

POSICIONAMIENTO EN INTERIORES BASADO EN DISPOSITIVOS MÓVILES

AUTOR: Miguel Ruiz Cuesta

TUTOR: Pablo A. Haya Coll

Grupo de Herramientas Interactivas Avanzadas (GHIA)

Dpto. de Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Abril 2012

Resumen y Palabras Clave

Resumen

La localización del usuario empleando dispositivos móviles es una demanda creciente en muchas de las aplicaciones disponibles hoy en día, desde redes sociales que solicitan nuestra posición como puede ser Foursquare¹ en la que podemos compartir con nuestros amigos nuestra situación, los lugares o establecimientos que nos interesan y que nos informe de cuáles son los más interesantes en nuestro entorno, pasando por navegadores como los de TomTom o Google Sky² que podremos ver las constelaciones en función de nuestra posición y la fecha (aunque necesita un posicionamiento poco preciso). En concreto, este proyecto está enmarcado y motivado en la posibilidad de guiar a personas discapacitadas en su día a día fuera de su hogar.

Metro de Madrid transporta cien mil personas al día [36], convirtiéndose en el transporte público más usado en la Comunidad de Madrid. Ante la imposibilidad de obtener una señal GPS u otro medio de posicionamiento dentro de el metro, tratamos de cubrir esta necesidad con la localización en interiores a través de las señales GSM que ese mismo dispositivo es capaz de utilizar. Nuestro sistema detecta la estación y la línea en la que se encuentra la persona en ese momento. Esta información puede ser empleada por otra aplicación que sea la encargada de detectar si va en la dirección correcta o cuando debe cambiar de línea para llegar a su destino.

Esta aplicación ha sido desarrollada para la plataforma Android, la cual cuenta con gran oferta de distintos terminales de varias gamas. Estos terminales resultan idóneos para este tipo de aplicaciones dado que la práctica totalidad de ellos cuenta con acceso a Internet y GPS incorporados, además de unas altas prestaciones. La migración a otras plataformas, como iOS o Blackberry, es factible aunque se encuentra fuera de los objetivos del trabajo.

¹<https://es.foursquare.com/>, diciembre 2011

²<http://www.google.com/sky/>, diciembre 2011

Palabras Clave

Localizacion en interiores, dispositivos moviles, Android, localizacion en Metro de Madrid.

Abstract

User localisation using mobile devices is an increasing demand for many of today applications. For example social newtorks as Foursquare³) ask for our localisation and later on share it with our friends, as well as suggest cafes, shops or other business of our interest in the near or GPS systems as Tomtom or Googlesky⁴ this last one allow us to see the constellation above us only with the date and the localisation (although it only needs a coarse one). Specipally, this project is motivated in the possibility of guiding handicapped people on the outside.

Metro de Madrid transports daily one hundred thousand people [36], becoming in the most used public transport in Madrid. Givien the fact of the impossibility of receiving GPS signals or any other positioning method inside the underground, we try to cover this need thanks to GSM signals that this same device is able to use. Our system detects the station and line in which the user is. This information may be used by any other application which would be entrusted if the user is in the right direction or when he should change line to arrive his destination.

This application has been developed for Android platform, for which there is a big and increasing oPer of devices of diPerent spectrum. This devices are suitable for this kind of applications due to GPS and Internet connection capabilities, as well as several advanced features. Its possible to develop this application for other operating system as Blackberry or iOS, but is not one of the main objectives of this project.

Key words

Indoor positioning, movile devices, Android, localization in Metro de Madrid.

³<https://www.foursquare.com/>, diciembre 2011

⁴<http://www.google.com/sky/>, diciembre 2011

Agradecimientos

Agradecer en primer lugar a mi tutor, Pablo, por toda la ayuda prestada y, sobre todo, por haber tenido la paciencia de llevar mi proyecto sin poner ninguna pega a las continuas intermitencias debidas a mis otros estudios de farmacia que han retrasado tant simos meses este proyecto.

Agradecer a Amilab la acogida y los cables que me han echado de vez en cuando, como Javi con Android. A pesar que ahora este algo mas vac o me llevo amistades y un buen recuerdo.

Tengo que agradecer especialmente a Patri junto con Adela y su compañero por ayudarme a recopilar informacion sobre el Metro de Madrid, que me ha facilitado y ahorrado muchas horas de trabajo. Es posible que sin esta ayuda no hubiese tenido una vision tan clara de buena parte de mi proyecto.

Muchas gracias a todos mis compañeros y amigos con los que pase las largas horas en los laboratorios, las clases y d as en la biblioteca. Las cañas en el German, las salidas nocturnas y algun pequeño viaje por Europa. A todos los que alguna vez han formado parte de mi vida universitaria como Aida que me ayudo cuando lo necesitaba, tambien a mis compañeros de Erasmus y los que me acogieron en el suyo que coincidio magicamente con el m o.

No puedo obviar a Moni y los (a veces) largos descansos en la cafeter a por la mañana que me hicieron mas amenas los d as de trabajo. A todos los que nos acompañaron esporadicamente en estos descansos.

Mi familia, en especial a mi madre y tambien a Eusebio fundamentales en poder haber empezado y terminado esta carrera. Sin ellos s que no habr a sido posible. Muchas gracias a todos.

Autor: Miguel Ruiz Cuesta
Fecha: abril 2012

Índice

1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	3
1.3. Organización de la memoria	3
2. Estado del Arte	5
2.1. Introducción	5
2.2. Tecnologías disponibles	6
2.2.1. Algoritmos de localización	6
2.2.2. Sistemas de localización en interiores	9
2.3. Localización por señales GSM	11
2.3.1. Introducción	11
2.3.2. Estudios previos realizados	12
2.3.3. Sobre GSM y datos de localización	13
2.3.4. Algunas aplicaciones móviles en Android	14
2.4. Plataformas móviles	16
2.4.1. Java 2 Micro Edition (J2ME)	16
2.4.2. Symbian	17
2.4.3. WebOS	17
2.4.4. Blackberry	18
2.4.5. Android	18
2.4.6. Windows Phone	19
2.4.7. iOS	19
2.4.8. Resumen de características	20
2.5. Conclusiones	21
2.5.1. Plataforma elegida	21
2.5.2. Algoritmo e infraestructura	22
3. Infraestructura del Metro de Madrid	25
3.1. Introducción	25
3.2. Historia del metro de Madrid	25
3.3. Infraestructura móvil	28
4. Diseño de aplicaciones	33

4.1.	Introducción a Android	33
4.1.1.	Aplicaciones	34
4.1.2.	Plataforma de desarrollo de las aplicaciones	34
4.1.3.	Bibliotecas	34
4.1.4.	Ejecución en Android	35
4.1.5.	Núcleo Linux	35
4.2.	Fundamentos de las aplicaciones de Android	36
4.2.1.	Componentes de una aplicación	36
4.2.1.1.	Activando componentes	38
4.2.2.	<i>Manifest</i>	38
4.2.2.1.	Declarar componentes	39
4.2.2.2.	Declarar capacidades de los componentes	39
4.2.2.3.	Declarar requisitos de la aplicación	39
4.2.2.4.	Context	40
4.3.	Aplicaciones desarrolladas	40
4.4.	Activity PowerMeasurement	41
4.5.	Service SampleReader	41
4.6.	Activity Reading	44
4.7.	Mapa de muestras	46
4.8.	Activity Position	48
4.9.	Clases auxiliares	50
4.9.1.	Cell	51
4.9.2.	Sample	52
5.	Algoritmo de localización en interiores	55
5.1.	Algoritmo de posicionamiento	55
5.1.1.	Mapa de muestras	56
5.1.2.	Localización a partir del mapa	57
5.2.	Estudio preliminar	59
5.2.1.	Pruebas de posición en el laboratorio	59
5.2.1.1.	Mapas de muestras realizados	60
5.2.1.2.	Resultados de localización	61
5.2.2.	Pruebas de posición del móvil	62
5.2.3.	Pérdida de intensidad sujetar el dispositivo	62
5.3.	Medidas y pruebas realizadas en el metro	63
5.3.1.	Mapas realizados	63
5.3.2.	Pruebas de localización	67
5.3.3.	Variación de potencia en estación por entrada de trenes	70
6.	Conclusiones y trabajo futuro	73
6.1.	Conclusiones	73
6.2.	Trabajo futuro	74
	Bibliografía	77

A. Vocabulario técnico, siglas y lenguaje técnico	83
B. Presupuesto	85
C. Pliego de condiciones	87

Índice de Figuras

2.1.	Posicion a partir de los angulos medidos.	8
2.2.	Captura de pantalla de la aplicacion <i>Antennas</i>	14
2.3.	Capturas de pantalla de la aplicacion OpenSignal.	15
3.1.	Ampliaciones de metro entre 1955 y 1966. En todas estas l neas las paredes estan recubiertas de ladrillo en lugar de hormigon.	27
3.2.	L neas de metro con cobertura movil. Las estaciones base estan en las estaciones numeradas. Las l neas verdes son con cobertura del exterior.	29
3.3.	Detalle del mapa de cobertura de telefon a sobre la l nea 10.	30
3.4.	Perdidas de propagacion de cable radiante (RCT7-CPUS) en un tunel circular de hormigon armado en la banda PCS (1800-1900MHz).	31
3.5.	Perdidas de propagacion de una antena en un tunel circular de hormigon armado en la banda PCS (1800-1900MHz).	32
3.6.	a) Antena bidireccional. b) Antena unidireccional. c) Antena omnidireccional.	32
4.1.	Ciclo de vida de una actividad.	37
4.2.	Ejemplo de declaracion de la actividad <i>PowerMeasurement</i> en el manipesto	39
4.3.	Diagrama UML de todas las clases de la aplicacion. <i>PowerMeasurement</i> es la encargada de lanzar cada una de las actividades de uno u otro programa.	42
4.4.	Captura de la vista de la actividad <i>PowerMeasurement</i> . El diagrama de UML se puede ver en el completo (pgura 4.3).	43
4.5.	Diagrama UML en detalle de <i>SampleReader</i>	43
4.6.	Captura de <i>Reading</i>	45
4.7.	Diagrama UML en detalle de <i>Reading</i>	46
4.8.	Ejemplo de una muestra con dos <i>fingerprints</i>	48
4.9.	Diagrama UML en detalle de la actividad <i>Position</i>	49
4.10.	Diagrama de ujo simplificado para la actividad <i>Position</i>	50
4.11.	Dos muestras de una localizacion. Debajo de los botones aparece el texto con todas las distancias calculadas de cada uno de los puntos.	51
5.1.	Estabilidad de la se#al en dBm en funcion del tiempo para se#ales GSM y Wip [55].	56

5.2. Ejemplo de un mapa en una casa pcticia. Cada zona tiene las huellas de seal de un color diferente	58
5.3. Distancia de la potencia medida a una huella de cada habitacion. Azul, rojo y verde las habitaciones, en magenta el vector medido en la prueba y en negro el vector distancia.	60
5.4. Tres posiciones perpendiculares entre s en las que se ha medido la variacion de potencia.	62
5.5. Variacion de potencia de seal al cambiar de posicion el dispositivo movil. Evento A: cambio de tumbado a canto. Evento B: segundo cambio de posicion.	63
5.6. Variacion de potencia al depositar el movil en la mesa. Cada color representa una seal distinta.	64
5.7. Esquema de donde se han tomado las huellas de seal en las estaciones de metro.	65
5.8. Potencia en recibida en dBm en funcion del tiempo donde se puede observar el efecto de los trenes en la seal. Tren A es el tren en el mismo anden en el que estamos, tren B el tren que circula en el sentido contrario	72

Índice de Tablas

2.1. Tabla con las características principales de cada una de las plataformas disponibles en el mercado.	21
2.2. Tabla con las características principales de cada una de las plataformas disponibles en el mercado.	22
4.1. Tabla con los campos disponibles en el mapa de muestras.	47
5.1. Tabla con los resultados de los experimentos en el laboratorio. El punto más lejano, los ascensores es el que menor error presenta.	61
5.2. Índice de acierto si tenemos en cuenta únicamente los servicios y los ascensores.	61
5.3. Primer grupo de estaciones en las que se indica por cada celdid el número de veces registrado con el número de huellas de señal en las que aparece, además de la potencia media (rss) en dBm.	66
5.4. Grupo de nuevos ministerios de nuevo, se muestra el número de veces que aparece el celdid por huellas y la potencia media.	66
5.5. Tercer grupo formado por las estaciones de Alonso Martínez y Tribunal con número de muestras por huellas y su potencia media.	67
5.6. Celdid con su frecuencia por número total de huellas y potencia para la estación de Plaza de España.	67
5.7. Resultados del primer grupo. Presenta un acierto del 36 %, es decir casi completamente aleatorio entre las tres estaciones.	69
5.8. Resultados del segundo grupo. Presenta un acierto del 50 %, de nuevo aleatorio entre ambas estaciones. Aunque en este caso hay errores con una estación del primer grupo.	69
5.9. Resultados del tercer y cuarto grupo (formado solo por plaza de España). Un acierto del 65 % y el 100 % respectivamente.	69
5.10. Resultados clasificados por zonas. Hay un 100 % de aciertos.	70
5.11. Las potencias medias de las antenas comunes de cada zona con sus respectivas desviaciones estándar de cada una.	71

Capítulo 1

Introducción

La localización, el conocer en que lugar estamos, ha sido siempre una necesidad para desenvolvernó en nuestro día a día. La tecnología siempre ha jugado un papel importante en esta tarea. Empezamos confeccionando mapas de las estrellas y desarrollamos el sextante y la brújula. Pero, no ha sido hasta la llegada de sistemas como el Sistema de Posicionamiento Global (*Global Positioning System* o GPS) hasta que la navegación sencilla y omnipresente ha llegado a manos de todo el mundo.

No obstante el problema de la localización en interiores todavía está por resolver. Aunque en esta memoria se muestran diversos sistemas y métodos para resolver este problema, todavía no hay ninguna sistema tan ampliamente extendido y tan popular como es el GPS hoy en día.

La demanda en la localización está creciendo en los últimos años de manera exponencial gracias a los dispositivos móviles, en concreto los teléfonos inteligentes. El empleo de la tecnología GPS por dispositivos portátiles e integrados en los coches nos acercaron la localización y, sobre todo, el guiado que tanto éxito ha tenido (y el por supuesto ahorro de tiempo) allá donde fuésemos. Pero, ahora han tomado el relevo los "smartphones", móviles con GPS e Internet integrados que nos dan mapas dinámicos (Google ya construye con su navegador la ruta óptima en función del tráfico que haya en ese momento) que han relegado al resto de dispositivos GPS a un segundo plano (las acciones de Tomtom descendieron cuando Google anunció su navegador gratuito). Ahora bien, un problema que presentan los GPS es que solo tenemos cobertura de los satélites cuando hay línea de visión directa, con lo que no tendremos cobertura en interiores.

Falta por tanto, esta última frontera para la localización ubicua y precisa que es en los interiores de edificios, en el interior del metro o en cualquier punto en el que la señal de GPS no alcance. Propuestas no faltan como mostraremos, pero todo apunta a que si hay algún sistema que no requiera desplegar nueva infraestructura, este contará con una gran ventaja. De nuevo, Google ha avanzado en este sentido. Cuando lanzas una de sus aplicaciones solicita activar la conexión inalámbrica de red local (WLAN) de nuestro dispositivo para ayudar a su localización y es que, sus coches de Streetview, han

realizado un mapa con las señales inalámbricas que reciben de al menos Wif y GPS, aunque utiliza también las antenas de teléfono móvil para localizar, allí por donde pasaban aumentando la precisión de localización cuando no se dispone de GPS.

De hecho en la última actualización de Google Maps para móviles¹, han incluido en Japón y en EE.UU. varios mapas de centros comerciales y aeropuertos por los que Google te da la opción de guiarte hasta tu puerta de embarque, tu tienda de interés o simplemente a la salida.

Es pues, un problema con un perfil cada vez más cercano y, quizá la solución sean los mapas de señales ya desplegadas (lo que requeriría o bien almacenarlos en memoria o bien una conexión a internet). Una solución de bajo coste al estar basada en infraestructuras existentes y al alcance de las nuevas tecnologías.

1.1. Motivación

Una persona con discapacidad cognitiva es aquella que está limitada en al menos dos de las siguientes áreas: comunicación, aptitudes sociales y de convivencia, uso de recursos comunitarios, autocuidado, aptitudes funcionales académicas, trabajo, tiempo libre, salud y capacidad de guiarse. Muchos desórdenes se expresan en un espectro y además, muchas veces aparecen combinados definiendo y caracterizando al individuo de manera única[11]. Estos desórdenes se manifiestan a la hora de enfrentarse a la movilidad en un entramado tan complejo como el transporte público en: leer y entender direcciones, coger el vehículo, salir en la estación correcta y entendiendo los avisos que dan por megafón o por paneles.

Este PFC se enmarca en el proyecto de ASIES (*Adapting Social and Intelligent Environments to Support people with special needs*), en el que la aplicación WAI-routes[34] se encarga de dirigir a la persona gracias a la localización por GPS. No obstante, las señales GPS no llegan al Metro de Madrid, donde se está desplegando infraestructura para dar servicios móviles a través de GSM. La idea es a través de la realización de un mapa de las señales GSM (Global System for Mobile Communications, originalmente Groupe Special Mobile) ser capaz de dar la posición a aplicaciones como WAI-routes que se encargan de guiar al individuo.

¹<http://googleblog.blogspot.com/2011/11/new-frontier-for-google-maps-mapping.html>, diciembre 2011

1.2. Objetivos

Este proyecto se enmarca dentro del sistema WAI (Where Am I) que pretende facilitar la movilidad de personas con discapacidad cognitiva. El transporte suburbano es muy sencillo de utilizar con un unico tren por v a y muy esquematizado de por s pero presenta el problema que los sistemas de localizacion en exteriores como el GPS no reciben señal en el subsuelo obligandonos a buscar otros medios para lograr la localizacion.

Por tanto el objetivo es analizar la viabilidad de desarrollar un sistema posicionamiento de bajo coste para el interior del metro de Madrid e implementar un prototipo que muestre las posibilidades del mismo. Es preciso que el sistema tenga unos coste de despliegue y mantenimiento bajos para poder ampliar hasta abarcar la mayor cantidad de l neas posibles y que pueda ser usado por el mayor numero de usuarios posibles sin aumentar mucho los costes.

As , la solucion que hemos estudiado parte de la idea de emplear estructuras ya existentes en dispositivos de relativo bajo coste y extendidos:

Dispositivos móviles: telefonos inteligentes (smartphones), en nuestro caso Android, con mucha cuota de mercado y sencillos de desarrollar sobre ellos. Ademas Android cuenta con una gama media de dispositivos mas economicos.

Infraestructura existente (GSM): desplegada en bastantes l neas de la red del metro de Madrid y utilizada por todos los telefonos moviles.

Algoritmos de bajo coste computacional: que consuma el m nimo de recursos, que pueda ser ejecutado en el mismo dispositivo sobre el que corre WAI y que de una posicion lo subcientemente precisa como para poder guiar a una persona a traves del metro de Madrid.

1.3. Organización de la memoria

En el primer cap tulo se presenta el proyecto, la motivacion y los objetivos del mismo.

En el segundo cap tulo se hace un estudio sobre el estado del arte de los campos asociados a este proyecto tanto referente a algoritmos de localizacion como de dispositivos moviles.

En el tercer cap tulo se presenta un breve estudio sobre la infraestructura movil de la red de Metro de Madrid y sus posible potencial y limitaciones.

En el cuarto cap tulo hay una introduccion al sistema operativo Android y el desarrollo de los dos programas en este sistema operativo.

El quinto cap tulo trata sobre el desarrollo del algoritmo empleado, sus partes y ejemplos. Ademas en este cap tulo estan explicadas las distintas pruebas de recepcion de señal y los resultados de las pruebas dentro del metro de Madrid.

Finalmente las conclusiones estan en el ultimo y sexto cap tulo as como una posible continuacion o mejoras del trabajo realizado.

El Apendice A es una lista con el vocabulario tecnico y sus traducciones al castellano.

El Apendice B es el presupuesto aproximado de los costes de este proyecto.

En el Apendice C aparecen las condiciones legales que se han seguido en la realizacion del proyecto.

Estado del Arte

2.1. Introducción

Los sistemas de posicionamiento en interiores a través de infraestructuras inalámbricas gozan hoy en día de gran popularidad [33]. El avance de la tecnología tanto en comunicaciones como en computación han permitido el desarrollo de redes de sensores [47], así como de dispositivos portátiles como Smartphones y PDAs de gran capacidad de computación que permiten la utilización de algoritmos que antes estaban reservados a equipos fijos [8].

Este crecimiento de la oferta de distintos sistemas de localización en interiores ha producido la aparición de distintas aplicaciones en los campos más variados como sistemas de transporte inteligente [16], facturación dependiendo de la localización [16], computación ubicua [57], servicios de información dependientes de la ubicación o del seguimiento del usuario [21, 27] por mencionar unas pocas. Muchas de estas requieren una precisión relativamente alta, tanto en exterior como en interior [23, 44].

Diferentes aplicaciones requieren diferentes tipos de localización, en este sentido se pueden clasificar los sistemas como [23, 33]:

- Localización física: que identifica un punto en un mapa, ya sea 2-D o 3-D. El más extendido son las coordenadas físicas de grados/minutos/segundos (DMS y UTM).
- Localización simbólica: la posición se describe mediante una etiqueta que hace referencia a una posición descrita en lenguaje natural. Por ejemplo cocina, despacho B-403 o servicio de caballeros de la tercera planta.

En función de la referencia que se emplee podemos distinguir [23]:

- Posición absoluta: que utiliza una malla de referencia para localizar todos los posibles objetos. Este sería el caso, por ejemplo, del GPS. Todos los dispositivos se localizarían con respecto a los mismos puntos.
- Posición relativa: en este caso cada dispositivo puede tener su propia referencia. Un ejemplo consistiría en la localización de víctimas de una avalancha en el que cada dispositivo de rescate localiza a las víctimas con respecto a sí mismo.

Siempre las coordenadas absolutas pueden transformarse a relativas y viceversa.

También se pueden clasificar las distintas tecnologías en función del algoritmo que usen o su variable de medida (véase en la sección 2.2.1), en función de la infraestructura que empleen (véase en la sección 2.2.2), tipo de entorno ya sea interior como exterior (dominado fundamentalmente por GPS) [23], la granularidad y precisión, la movilidad de los nodos o balizas o bien en función de si es el dispositivo móvil el que realiza el cálculo de la posición [23] (GPS es un buen ejemplo de este último tipo) o si el dispositivo envía las medidas y es un servidor central el que calcula la posición[8].

2.2. Tecnologías disponibles

A parte de la clasificación detallada en la introducción de este capítulo, una clasificación bastante empleada es elegir un sistema en función del algoritmo que empleen [23]. Este nos determinará el hardware a emplear y, por tanto, la escalabilidad de la opción elegida debido a la infraestructura necesaria de antenas o balizas y el número máximo de dispositivos que es posible localizar, por ejemplo GPS tendrá un número ilimitado de usuarios [24]. Segundo, debemos considerar si es necesaria una clasificación de los distintos objetos en el sistema y su diferenciación, como podría ser un sistema de transporte de maletas en un aeropuerto. Tercero, los costes, los cuales estarán más fuertemente ligados al hardware utilizado. Y por último deberemos tener en cuenta otras limitaciones como la granularidad, precisión o alcance del sistema.

2.2.1. Algoritmos de localización

En este apartado se hace hincapié en los distintos algoritmos desarrollados y las propiedades de las señales sobre los que trabajan [33, 47].

En primer lugar mencionamos los algoritmos basados en la triangulación. Es decir, midiendo las distancias que hay entre las balizas y el usuario:

Tiempo de llegada (TOA): es el tiempo que transcurre desde que se emite la señal en el dispositivo a localizar hasta que llega al dispositivo de medida, la distancia se calcula en forma de función euclídea:

$$f_i(x) = c(t_i - t) - \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

(donde c es la velocidad de la luz). En general, se precisan tres puntos de referencia para una localización en dos dimensiones [18]. Necesita una gran precisión en la sincronización y en el tiempo de referencia. Así en el caso de GPS, para cumplir estos requisitos, cada satélite cuenta con su respectivo reloj atómico [24].

Este tiempo puede ser medido usando diferentes técnicas como espectro ensanchado (DSSS) [10, 32] o *Ultra Wide Band* (UWB) [2], además de otros algoritmos basados en TOA tales como vecino más cercano (closest-neighbor) o peso residual (residual weighting) [26]. El primero aproxima la situación del usuario en función a la estación base más cercana mientras que el segundo es una forma ponderada de mínimos cuadrados.

Tiempo de llegada diferencial (TDOA): se basa en determinar la posición relativa del usuario midiendo la diferencias de tiempo de la señal recibida en distintos dispositivos, en lugar de el tiempo de llegada absoluto. Para cada medida, en lugar de distancia euclídea quedan expresadas como función hiperbólica

$$R_{i,j} = \sqrt{(x_i - x)^2 + (y_i - y)^2} + (z_i - z)^2 - \sqrt{(x_j - x)^2 + (y_j - y)^2} + (z_j - z)^2$$

donde (x_i, y_i, z_i) y (x_j, y_j, z_j) representan las señales recibidas y (x, y, z) las coordenadas del objetivo [16]. Normalmente se simplifican las ecuaciones empleando un desarrollo en serie de Taylor y creando un algoritmo iterativo [53].

Tiempo de vuelo (RTOF): este método consiste en medir el tiempo de vuelo de la señal desde el transmisor hasta el usuario y de vuelta. Requiere una sincronización relativamente alta. Este método está presentado en [19] con un error de unos pocos metros. El mecanismo de funcionamiento es el mismo que TOA y es muy parecido al de un radar ya que el dispositivo en el fondo responde a una señal interrogatoria de radar.

Ángulo de llegada (AOA estimation): la posición con este método se puede lograr conociendo la intersección de varias líneas de dirección. Si conocemos el ángulo incidente de las ondas y la localización y distancia entre las estaciones base que emiten estas señales, podemos establecer la posición del objetivo [46] como podemos ver en la figura 2.1 [12]. Con este sistema, bastan con tres señales de medida para poder posicionarse en tres dimensiones o tan solo dos para un posicionamiento en dos dimensiones. El problema es que necesita un hardware muy complejo y voluminoso, incrementando el precio y la escalabilidad. Además sufre problemas con las sombras, el multirayecto dando ángulos equivocados o con objetivos en movimiento [1].

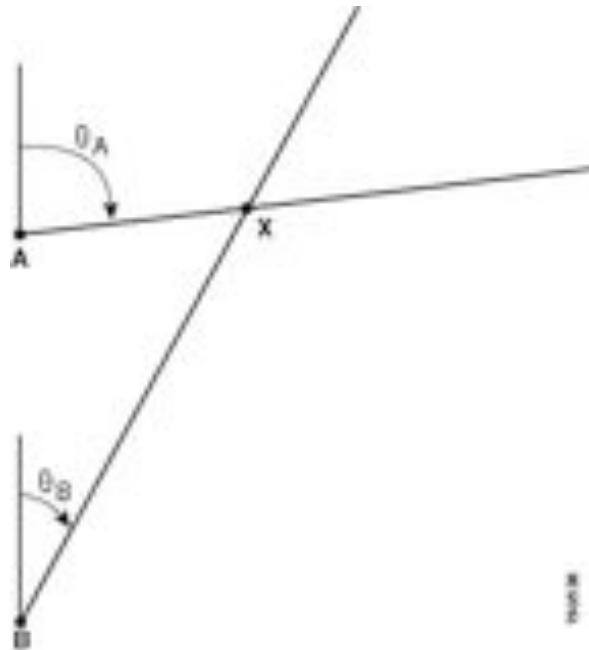


Figura 2.1: Posición a partir de los ángulos medidos.

Potencia de señal recibida (RSS): Todos los métodos anteriores no muestran el rendimiento esperado si no tienen una línea de visión directa con el emisor de la señal [33]. Utilizando la atenuación de la señal emitida nos da una precisión más alta en estas situaciones, pero a distancias cortas. La incertidumbre crece con la distancia debido a problemas como el desvanecimiento, las sombras o múltiples trayectorias dando lugar a una precisión bastante pobre [4]. Los algoritmos basados en la atenuación de la señal calculan el camino que ha recorrido la señal a través de la atenuación sufrida, por lo tanto la diferencia entre la potencia emitida y la recibida nos dará una idea de la distancia hasta la antena [33].

Proximidad: simple y ligero, asume que el usuario está en el área de la antena de la que recibe la mayor potencia. Cuanto mayor sea la densidad de antenas, más preciso será el método.

Otros métodos disponibles son a través del análisis de escena, en los cuales se miden ciertas características del medio en el que se localiza el usuario:

Huellas de señal (fingerprints): Este es el método usado en este proyecto. Consiste en la realización de un mapa de la escena donde se realizará la localización de una característica. En todos los casos estudiados, los fingerprints (o muestras) serán la potencia medida de la señal (RSS). Una vez realizado el estudio de la escena se podrán usar distintos algoritmos probabilísticos eligiendo cuál será la

posición más probable [51]; redes neuronales [33]; algoritmos de clasificación vectorial [5] junto con algoritmos de aprendizaje como el SVM [54]; o simplemente calculando la distancia euclídea al mapa de huellas de señal [55, 43].

Easy Living: método desarrollado por Microsoft en el que una serie de cámaras se encarga de localizar al individuo [28]. Véase la sección 2.2.2.

2.2.2. Sistemas de localización en interiores

En este apartado se analizan distintos sistemas desde el punto de vista de hardware que necesitan, una de las características más a tener en cuenta a la hora de elegir. En esta sección por tanto, se basa en los algoritmos ya descritos (véase la sección 2.2.1) y exponemos algunas de sus versiones comerciales.

Active Bat: utiliza una red de receptores ultrasonicos en el techo que recibe la señal del dispositivo a localizar a través de TOA. La sincronización viene por parte de una señal de radio. Tiene una precisión de unos 10 cm en el 95% de los casos [22].

Cricket: utiliza, como el anterior, tanto una red de balizas en el techo como señales de radio para la sincronización y ultrasonicas para la localización que alcanza una precisión de unos 2 cm. Emplea TOA y un filtro Kalman para mejorar el comportamiento dinámico. Presenta una escalabilidad descentralizada y una alta privacidad, no obstante tiene altos requisitos de computación por la necesidad de precisos sincronismos y emplear dos señales de distinta naturaleza con sus respectivos receptores [48].

Hexamite: tiene tres tipos de transmisores: emisores, receptores y controladores. Utiliza el algoritmo de TDOA y, por lo tanto, hace uso de la función hiperbólica pero, a cambio, no requiere sincronización. Obtiene una precisión de 1cm, pero necesita también una alta densidad de antenas [8].

AHLos: la ventaja principal de este sistema es su instalación dinámica, la red se despliega haciendo uso de infraestructura ya existente como GPS (funciona tanto en exteriores como en interiores) para localizar a sus propias balizas. A continuación todos los nodos estiman su posición a los vecinos a partir de los nodos que ya conocen su posición, los denominados nodos baliza, tras lo que comunican su posición a todos los demás y a broadcast. Para la localización utilizan TDOA a través de una señal ultrasonica y una señal de radio. Como se puede deducir, el mayor problema de este sistema son los altos costes de computación [39].

D-Locus: la particularidad de este sistema es que funciona en el rango de señales acústicas dentro del espectro audible por el ser humano, el cual presenta unas

buenas propiedades de atenuación y rango. Utiliza TOA y el sistema es bidireccional, es decir que todos los nodos son tanto emisores como receptores. Obtiene una precisión de 1 cm. Su mayor desventaja es que requiere gran potencia de cálculo [8].

RADAR: sistema muy referenciado en la literatura [26, 55, 23, 52, 33, 47] desarrollado por Microsoft [9], basado en las redes inalámbricas IEEE 802.11.

RADAR mide la potencia y la relación señal-ruido de la señal wi-fi y la emplea para posicionar el dispositivo en un mapa previamente ya muestreado a base de huellas de señal. La importancia de RADAR radica en que es el primer método que utiliza realmente las huellas de señal para la localización, que empleamos en este proyecto. Otras empresas disponen de soluciones comerciales como WhereNet¹ que basan su funcionamiento en el método empleado por RADAR.

Tiene una precisión relativamente buena, de unos 2-3 m. Con tan solo cuatro 802.11 puntos de acceso. Es un sistema además de bajo coste ya que no requiere otra infraestructura más que los routers wi-fi que estén instalados, el único problema es que la infraestructura puede variar relativamente rápido pudiendo necesitar rehacer los mapas de potencias.

PinPoint 3D-iD: se colocan antenas en el techo que emiten secuencialmente señales, mientras el dispositivo a localizar emite su propia señal. La medida de tiempo de estas dos señales se emplea para calcular la posición (algoritmo de tiempo de vuelo) [58].

MotionStar magnetic tracker: es un sistema de gran precisión espacial (1mm de precisión y 0,1 ° de orientación) y una gran resolución temporal (1ms de refresco). Genera un sistema de seguimiento a través de pulsos continuos magnéticos que emite desde una antena pja. A continuación mide la posición y la orientación a través de una serie de antenas que reciben la respuesta en los tres ejes ortogonales espaciales del pulso transmitido teniendo en cuenta el efecto continuo del campo magnético de la Tierra. El mayor problema es que los sensores deben estar a una distancia de entre 1 y 3 metros del emisor [49].

UWB Ubisense: utiliza señales de espectro muy ensanchado, que presenta un buen comportamiento ante el multicamino. Tiene un alcance de unos 50 metros, con poca densidad de balizas y una precisión de unos 20 cm [8].

Easy Living: desarrollado por Microsoft, consiste en tres cámaras por equipo con reconocimiento facial, tonalidad de la piel y siluetas. Siendo este sistema más que suficiente para la localización de una persona dentro de una habitación. Su mayor problema es la alta necesidad de equipos potentes para realizar todo el cómputo necesario [28].

¹<http://www.widata.com>

Active Bagde: consiste en un móvil con un dispositivo de proximidad infrarrojo, este emite una señal de localización cada 10 segundos (es programable) a sensores infrarrojos instalados en el edificio que mandan la información a un servidor central. Puede aportar por lo tanto, tanto referencia absoluta como simbólica. El mayor problema que presenta es que no funciona bien en presencia de luces fluorescentes o con luz directa del sol, ya que en ambos casos aumentan mucho el ruido en la banda de la luz infrarroja [56].

Identificación por radiofrecuencia (*Radio-frequency identification* RFID): utiliza etiquetas activas. Las etiquetas activas son tipo de almacenamiento y recuperación de datos a través de transmisiones electromagnéticas a un circuito integrado compatible con radio frecuencia (RF). Estos circuitos con antenas pueden ser tanto pasivos, que no requieren una fuente de alimentación interna y únicamente emiten una señal cuando reciben una con un alcance de unos dos metros, como activas que tienen fuente de alimentación propia y pueden ir emitiendo periódicamente a mayor distancia o realizar tareas más complejas pero tienen un coste mayor. Por ejemplo, un tipo muy usado de etiquetas pasivas son las alarmas antirrobo de las grandes superficies.

Se basa en un algoritmo en tres dimensiones de agregación a través de la intensidad de la señal de radio. Reparten sensores de manera homogénea y calculando la correlación de la señal entre los distintos sensores logran obtener una posición precisa del dispositivo [25]. Estos sistemas se pueden utilizar en sistemas de localización de equipajes en los aeropuertos o seguimiento de paquetes en almacenes industriales.

Smart Floor: Consiste en una serie de sensores de presión en el suelo para capturar pisadas para localizar y seguir a peatones [41]. Un problema que presenta es su escasa escalabilidad dado que requiere un sensor por cada punto donde se localiza.

2.3. Localización por señales GSM

2.3.1. Introducción

Todos los sistemas arriba mencionados que requieren desplegar una infraestructura son, en mayor o menor medida, costosos y requieren equipos propios, su despliegue y mantenimiento. Además, en la mayoría de los casos, son de difícil escalabilidad. Una opción de menor coste es la localización a través de las infraestructuras ya existentes como en la anteriormente mencionada RADAR (véase 2.2.2) que utiliza 802.11. Otros emplean bluetooth, GSM o las señales de otras tecnologías como UMTS.

De entre estas tres tecnologías GSM presenta una serie de ventajas frente a las demás:

- GSM está ampliamente desplegada en septiembre del 2005 contaba con el 78 % del mercado global de telefonía móvil [52]. Hoy en día da cobertura al 80 % de la población mundial² con cerca de seis mil millones de conexiones móviles en noviembre del 2011[20].
- Lo anterior implica que podemos usar cualquier móvil (incluidos smartphones) para implementar nuestra aplicación.
- La infraestructura GSM está diseñada para tolerar fallos del suministro eléctrico, pudiendo posicionar nuestro dispositivo incluso ante una posible situación de emergencia [55].
- GSM trabaja sobre una banda licenciada, no habrá interferencias de otros dispositivos como sufren 802.11 o bluetooth de dispositivos como teléfonos móviles inalámbricos.
- La complejidad de las redes GSM y los estudios que hay que hacer antes de instalar cualquier antena implica que la red evoluciona lentamente y sufre cambios infrecuentemente [55].

2.3.2. Estudios previos realizados

En la literatura se pueden encontrar diferentes estudios que emplean la localización a través de señales GSM. En [52] consiguen obtener la actividad del individuo, ya sea estando parado, caminando (velocidad de entre tres y seis km/h) y conduciendo (entre 15-125 km/h) con una precisión del 85 %.

Place Lab logró el objetivo de obtener una localización en ciudad de poca granularidad a través de un dispositivo móvil que realiza un mapa con el mínimo esfuerzo la distribución de las señales, obteniendo aproximadamente una precisión de 100-150 metros [31]. Este estudio, al igual que el ejecutado por Laasonen [29], emplea únicamente para la localización las lecturas de la antena a la que está asociado, aunque en este último caso el objetivo era simplemente aportar una localización simbólica como *casa* o *droguería* y una estadística con esto sobre los lugares que el usuario suele visitar.

Laitinen et al. [30] empleó unos mapas de señales GSM más detallados, tomando la potencia de las 6 antenas que se reciben con mayor intensidad, obteniendo una precisión de 44 metros. Una precisión relativamente baja, principalmente por el número

²<http://www.gsm.org/technology/gsm/index.htm>, septiembre 2011

de señales tomadas para realizar los cálculos. En [55] logran obtener una precisión de entre 2 y 4 metros y situando en el piso correcto a una persona en un edificio con una precisión del 60% y de un 98% entre dos pisos, llegando para ello a usar hasta 29 canales distintos de GSM para tomar las huellas de señal, con una distancia entre cada uno de ellos de aproximadamente un metro.

2.3.3. Sobre GSM y datos de localización

GSM cuenta con una serie de identificadores que nos dan información sobre la antena, la potencia o la operadora a la que estamos conectados [3]:

- Identificador de celda (cellid): es un número que identifica únicamente cada estación base (BTS) dentro de un área definida por
- Código de país para móviles (MCC): es un código identificador que aporta el país. En el caso de España es el 214.
- Código de red de móviles (MNC): junto con el MCC se usa para identificar la operadora y el país. Por ejemplo en España 214 01 es Vodafone o 214 25 es LycaMobile.
- Código de localización de área (LAC): número de 7 cifras que identifica el área en el que se encuentra la estación base. Estas áreas pueden ser bastante grandes y son usadas para la localización de móviles y el establecimiento de conexiones entre móviles.
- Indicador de potencia recibida (RSSI): es la potencia recibida en dBm. Este valor está definido en [3] y varía entre 0 y 31 indicando 0 una potencia de -113 dBm o menor y 31 -51 dBm o mayor. Cada unidad significa un incremento de 2 dBm.

Estos identificadores están disponibles en varias bases de datos en internet y diversas APIs³ junto con las coordenadas de sus respectivas antenas GSM.

La infraestructura de red GSM está dividida en celdas a la cual se conecta cada dispositivo GSM. Las celdas pueden ser de distinto tamaño, pudiendo cubrir hasta 35 kilómetros. Cada celda y su red está compuesta de:

- Estación Base (*Base Station Subsystem BTS*): donde se disponen los emisores y receptores de radio. Son transmitidos por coaxial a la antena que radiará la señal.

³Zonetag de Yahoo (noviembre 2011): <http://tech.groups.yahoo.com/group/zonetag-dev/>, Open-cellid (noviembre 2011): <http://opencellid.org/>, API de Ericsson que incluye WiFi (noviembre 2011): <https://labs.ericsson.com/apis/mobile-location/>, <http://www.location-api.com/>

- Subsistema de red (*Network and Switching Subsystem*): muy similar a una red pja, se encarga de dar acceso a redes telefonicas y de datos.
- Centro de control y mantenimiento (*Operation Support System OOS*): el encargado de garantizar la explotacion tecnica y comercial de la red.

2.3.4. Algunas aplicaciones móviles en Android

A pesar de la disponibilidad de estas bases de datos en Internet⁴ la mayoría de estas aplicaciones utilizan la propia base de datos que ofrece la API de Google⁵.

Antennas: Toma la potencia de las antenas GSM mas proximas y triangula en funcion de la posicion y la potencia recibida de cada una de estas. Ademas remarcan que no siempre la base de datos de Google es pable y, en ocasiones, yerra por varias miles de kilometros. Ademas aporta informacion sobre la red y señal recibidas⁶.



Figura 2.2: Captura de pantalla de la aplicacion *Antennas*.

Maps/Navigaton: el navegador y los mapas detallados de Google que, cuando no disponen de señal GPS, se sirven de las señales GSM y 802.11⁷.

⁴<http://www.antennasearch.com/>, 27 de septiembre 2011

⁵<http://www.panix.com/~mpoly/android/antennas/r1.0>, septiembre 2011

⁶<http://www.panix.com/~mpoly/android/antennas/r1.0/>, septiembre 2011

⁷<http://www.google.com/mobile/navigation/>, septiembre 2011

2.3. LOCALIZACIÓN POR SEÑALES GSM

OpenSignal: da la posición aproximada, así como información acerca de las redes y señales, estadísticas de conexión a distintas estaciones y estadísticas de potencia de señal recibida⁸.



Figura 2.3: Capturas de pantalla de la aplicación OpenSignal.

RF Signal Tracker: otra aplicación muy parecida a las anteriores aportando posición e información de la red⁹.

En estas aplicaciones se hace patente algo que hemos comprobado a lo largo del proyecto: la mejor localización y más información sobre las antenas lo proporciona la red de GSM (2G) mientras que la información disponible sobre las redes 3G es pobre e imprecisa y en redes CDMA es prácticamente inexistente.

En el mejor de los casos, estas aplicaciones aportan una precisión de decenas de metros (se puede observar en función de los círculos alrededor de la posición en el mapa)

⁸<http://opensignalmaps.com/>, septiembre 2011

⁹<http://sites.google.com/site/androiddevelopmentproject/home/rf-signal-tracker>, septiembre 2011

y, experimentalmente, no solo acertar bajo el metro de Madrid presumiblemente porque estas antenas no están registradas todavía en la base de datos de Google.

2.4. Plataformas móviles

Hoy en día los móviles van aumentando la potencia hasta acercarse a la capacidad de un pequeño ordenador personal [14], habiendo en el mercado dispositivos con procesadores de doble núcleo [40] y a la espera de los cuatro núcleos como el chip de nVidia Tegra 3¹⁰. Incluso los sistemas operativos parece que empiezan a converger, como el futuro Windows 8 que se espera que funcione tanto en ordenadores convencionales como en tabletas y teléfonos inteligentes¹¹. Esto permite una libertad y una disponibilidad de recursos cada vez mayor para los desarrolladores, que también se ven beneficiados por los continuos avances de los sistemas operativos, como veremos a continuación. Además de un mercado cada vez mayor debido al crecimiento en el número de estos dispositivos, alcanzando el 40% del mercado en EE.UU. en septiembre del 2011¹².

2.4.1. Java 2 Micro Edition (J2ME)

Fue introducida en 1999, desarrollada por Sun Microsystems¹³ para desarrollo de software en dispositivos con recursos limitados.

Las aplicaciones deben ser ejecutadas a través de una máquina virtual (KVM: "Kilo Virtual Machine") que posee una gran capacidad de configuración, lo que permite una alta portabilidad tanto entre dispositivos como entre sistemas operativos y cualquier dispositivo actual es capaz de cargar aplicaciones de esta plataforma [42]. Podríamos decir que J2ME es una simplificación del estándar J2SE, con API reducida para adaptarse a las necesidades de las capacidades reducidas de los terminales móviles.

Esta plataforma no dispone de un mercado común y las aplicaciones deben descargarse directamente de un servidor conocido o a través de la operadora. Además de estar relegada a terminales antiguos o de gama baja, sin soporte para pantallas táctiles.

¹⁰<http://www.nvidia.com/object/tegra-superchip.html> noviembre 2011

¹¹<http://windows8beta.com/>, septiembre 2011

¹²http://blog.nielsen.com/nielsenwire/online_mobile/40-percent-of-u-s-mobile-users-own-smartphones-40-percent-are-android/, septiembre 2011

¹³<http://www.oracle.com/technetwork/java/javame/>, septiembre 2011

2.4.2. Symbian

Esta plataforma es el resultado de la alianza de Nokia, Sony Erikson y Samsung entre otras para competir con las ahora casi desaparecidos PalmOS y Windows Mobile. La plataforma paso a manos en exclusiva de Nokia¹⁴, pero su futuro parec a dudoso tras la alianza de esta con Microsoft para desarrollar smartphones con Windows Phone 7 y superiores [7]. Aunque Nokia anuncio que seguira lanzando nuevos moviles que ejecutaran sobre Symbian.

Utiliza el lenguaje de programacion orientados a objetos C++ y acepta tambien Java. Aunque la retrocompatibilidad es bastante pobre, necesitando en ocasiones volver a compilar para portar a versiones superiores de Symbian.

En cuanto a su tienda de aplicaciones OVI¹⁵, posee unas 50.000 aplicaciones con seis millones de descargas diarias. Este dato tiene una repercusion curiosa: que la aplicacion media tiene un 160% mas de descargas en OVI que en la App Store de Apple¹⁶, lo que denota la buena salud de la tienda y un importante atractivo para desarrolladores.

2.4.3. WebOS

Es el sistema operativo sustituto del antiguo PalmOS cuya primera version data de 1996 y diseñado para PDAs. En abril del 2010 fue adquirida por HP tras una larga epoca cosechando malos resultados debido a la competencia creciente con Apple, Android, Microsoft y BlackBerry y su futuro es un tanto incierto debido a la falta de claridad de HP sobre que hacer con este sistema operativo¹⁷.

El lenguaje principal de desarrollo es C/C++ aunque da soporte a Java y Visual Basic. La retrocompatibilidad es muy elevada, pudiendo ejecutar aplicaciones en versiones posteriores sin ningun problema. Dado que lleva muchos años en desarrollo, tiene un amplio numero de aplicaciones aunque su presencia en el mercado es cada vez mas escasa debido, en parte, a la falta de nuevos terminales. La instalacion de aplicaciones se realiza a traves del PC.

Desgraciadamente se ha interrumpido el desarrollo de este sistema operativo, aunque segun informo HP exploraran otros opciones para seguir desarrollando WebOS y su software [6].

¹⁴<http://symbian.nokia.com/>, septiembre 2011

¹⁵<http://store.ovi.com>, septiembre 2011

¹⁶http://www.developer.nokia.com/Distribute/Ovi_Store_statistics.xhtml, septiembre 2011

¹⁷<https://developer.palm.com/>, septiembre 2011

2.4.4. Blackberry

Plataforma desarrollada por RIM (Research In Motion)¹⁸ para sus terminales orientados a los negocios. En 1999 lanzaron su primer terminal, la Blackberry 850 un dispositivo con teclado físico "qwerty" y un servidor de correo electrónico Push, navegación por internet, además de las características típicas de un móvil de gama alta de la época y otras opciones de comunicación inalámbrica que le brindaron un gran éxito en poco tiempo [17].

Estas características se han mantenido en casi todos sus terminales. Si bien se ha extendido a otros mercados aparte del empresarial (disponen todavía del 21 % del mercado de los smartphones aunque haya caído del 25 % que mantenía hace tres meses¹⁹) debido a la competencia con iOS y Android, por ello quizá hayan diversificado su oferta con dispositivos puramente táctiles o incluso el desarrollo de una tableta propia²⁰.

Blackberry dispone de una tienda de aplicaciones, la App World²¹ con aproximadamente 3 millones de aplicaciones descargadas diariamente y un total de 45.000 aplicaciones. Su desarrollo puede ser en HTML5, CSS y JavaScript.

2.4.5. Android

Plataforma de desarrollo creada por Google y Open Handset Alliance. Tras no llegar a un acuerdo con la entonces independiente Sun Microsystems para que se uniera a esta plataforma, tuvo que desarrollar parte de la plataforma²² correspondiente a Java retrasándose el lanzamiento de la versión 1.0 más de un año hasta el 2008.

Actualmente conviven la versión 3.2 Honeycomb pensada para tabletas junto con la 2.3 Gingerbread a punto de actualizarse a la 2.4 Ice Cream Sandwich²³. El sistema operativo tiene un núcleo basado en Linux con código abierto. Su SDK son un conjunto de APIs que dan acceso a todos los recursos del terminal, la programación es en Java sobre un entorno como Eclipse y se dispone de un emulador bastante realista con el que realizar pruebas. Una ventaja con respecto a su competidor directo es que para empezar a programar para estos dispositivos no es necesaria ninguna licencia aunque, si

¹⁸<http://us.blackberry.com/developers/>, 30 septiembre 2011

¹⁹<http://mashable.com/2011/08/30/android-iphone-blackberry-smartphone-stats/>, 31 de agosto de 2011

²⁰<http://es.blackberry.com/>, 30 septiembre 2011

²¹<http://us.blackberry.com/apps-software/appworld/>, 29 septiembre 2011

²²http://www.computerworld.com/s/article/9219818/Google_offered_to_split_mobile_revenue_with_Sun_papers_show/, septiembre 2011

²³<http://developer.android.com/sdk/index.html>, septiembre 2011

quisiesemos publicar, necesitamos una licencia para publicar en el Android Market²⁴ (tienda de aplicaciones de Android) por 25 dólares que dura 30 años.

Hoy en día la tienda de aplicaciones de Android es la mayor de todas sus competidoras, con 500.000 aplicaciones con 6 mil millones de descargas hasta la fecha y el mayor porcentaje de aplicaciones gratuitas de todas²⁵. Esto es debido a que también dispone de la mayor cuota de mercado con un 41,8%²⁶ gracias a un amplio catálogo de terminales de marcas como HTC, Motorola, Sony Ericsson o Samsung que tienen dispositivos tanto gama alta como media o baja.

2.4.6. Windows Phone

Es la plataforma sustituta de Microsoft Mobile y no compatible con esta. Esta más orientada que su antecesora al gran mercado, en lugar del empresarial al que iba dirigida Microsoft Mobile. Lanzada a nivel mundial a finales del 2010²⁷, actualmente esta ya en la versión 7.5 Mango y su aparente objetivo es converger con los sistemas operativos de PC cuando llegue Windows 8.

Una característica importante es que, si bien cualquier fabricante puede desarrollar un teléfono con Windows Phone, Windows exige unos requisitos mínimos para poder emplear su sistema operativo. Esto implica que no exista gama media o baja en Windows Phone, pero también garantiza a los desarrolladores unos recursos bastante notables para un dispositivo móvil.

El desarrollo de aplicaciones para esta plataforma²⁸ se puede programar empleando Silverlight (la versión de Microsoft de Adobe Flash), XNA (pensado para el desarrollo de juegos) o una versión compacta de .NET. El SDK está disponible gratuitamente y se instala sobre Visual Studio. Una vez desarrollada, la aplicación debe ser aprobada por Microsoft para su publicación.

2.4.7. iOS

Es el sistema operativo móvil de Apple, que usan sus dispositivos iPhone, iPod Touch e iPad.

²⁴<https://market.android.com/>, septiembre 2011

²⁵<http://www.androlib.com/appstats.aspx>, septiembre 2011

²⁶<http://mashable.com/2011/08/30/android-iphone-blackberry-smartphone-stats/>, 31 de agosto de 2011

²⁷<http://msdn.microsoft.com/en-us>, septiembre 2011

²⁸<http://create.msdn.com/en-US/>, septiembre 2011

Su primera version data del 2007 con el lanzamiento del iPhone 2G, un dispositivo que no contaba con conexión de datos ni multitarea [38]. Se han ido actualizando a la par, hasta llegar a la segunda versión de su tableta y a punto de publicar la quinta del iPhone (junto a este se ha ido actualizando siempre el iPod Touch) el próximo octubre.

Los desarrolladores tienen a su disposición un SDK que, al igual que sus competidores, viene con emulador para probar la aplicación y, su desarrollo, es en Objective C. El SDK de iPhone obliga a usar un terminal de Apple, no pudiendo desarrollar ninguna aplicación desde otro sistema operativo que no sea MacOS²⁹, además de necesitar una licencia o bien orientada a la docencia (limitadas y complicadas de obtener), o bien de desarrollador con un coste de 100 dólares al año. Tiene una portabilidad bastante alta, si bien hay que tener en cuenta las mejoras que han sufrido en la evolución natural de los dispositivos como resolución, procesador y demás recursos. Además de la necesidad de pasar un filtro que cumpla los estándares de Apple antes de poder publicar.

La tienda de aplicaciones (App Store) es la segunda en tamaño tras el Android Market, con 425.000 aplicaciones³⁰ es la que más ingresos por aplicaciones de pago tiene, si bien tiene un porcentaje mucho menor de aplicaciones gratuitas en comparación a otras plataformas.

2.4.8. Resumen de características

Las tablas 2.1 de este apartado no contempla los costes de desarrollar aplicaciones que no se van a publicar. Por ejemplo, para iOS solamente se puede desarrollar legalmente desde un MacOS X y con una licencia (o bien puede ser de estudiante gratuita). Mientras que para Android y WebOS se puede programar en Linux, Windows y MacOS X de manera gratuita sin limitaciones, siempre y cuando no queramos publicar.

Aunque en el apartado de WebOS aparezca una versión relativamente reciente, todavía hay mucha incertidumbre sobre el rumbo de este sistema operativo (vease apartado 2.4.3).

²⁹<http://developer.apple.com/>, septiembre 2011

³⁰<http://www.apple.com/es/iphone/apps-for-iphone/>, septiembre 2011

Plataforma	Lenguaje	Publicación de aplicaciones
J2ME	Java	Libre, descentralizado
Symbian	C++/Java	Tienda OVI (50.000)
WebOS	C/C++ soporta Java y Visual Basic	App Catalog
Windows Phone	Silverlight, XNA, .NET	Marketplace (40.000)
Blackberry	Java	BlackBerry App World (50.000)
Android	Java	Android Market (500.000)
iOS	Objective C	App Store (425.000)

Plataforma	Estado de desarrollo	Compatibilidad con versiones anteriores
J2ME	Parado 2008 (limitado a JRE 1.3)	Alta
Symbian	En funcionamiento Symbian Belle (agosto 2011)	Pobre (puede requerir recompilación)
WebOS	En suspenso WebOS 3.0.4 (octubre 2011)	Alta
Windows Phone	Continua Mango 7.5 (noviembre 2010, última versión noviembre 2011)	Baja (SO nuevo)
Blackberry	Continua BlackBerry OS 7 (agosto 2011)	Alta
Android	Continua 4.0 Ice Cream Sandwich (noviembre 2011)	Alta
iOS	Continua iOS 5 (octubre 2011)	Alta

Tabla 2.1: Tabla con las características principales de cada una de las plataformas disponibles en el mercado.

2.5. Conclusiones

2.5.1. Plataforma elegida

La razón fundamental de elección de la plataforma Android para el desarrollo del proyecto es por ser continuación de WAIRoutes [34], el cual ha sido desarrollado sobre Android.

Plataforma	Terminales	Coste publicación
J2ME	Cualquier terminal con MicroJVM ³¹	Gratuito, sin filtro
Symbian	Terminales Nokia: telefonos inteligentes	1 euro, filtro poco estricto
WebOS	Terminales Palm y HP: telefonos inteligentes y tabletas	Gratuito
Windows Phone	Unos 10 fabricantes: telefonos inteligentes	75 euros anuales. Filtro estricto
Blackberry	Terminales BlackBerry: telefonos inteligentes	Gratis. Filtro poco estricto
Android	Mas de 25 fabricantes: telefonos inteligentes y tabletas	20 dolares. Filtro poco estricto
iOS	iPod, iPhone, iPad	100 dolares anuales. Filtro estricto

Tabla 2.2: Tabla con las características principales de cada una de las plataformas disponibles en el mercado.

No obstante, la plataforma Android es una elección muy acertada. De las nombradas en este estado del arte (2.4), las óptimas para desarrollar un proyecto de estas características son Android, iOS y Windows Mobile. Ya que WebOS está casi retirada del mercado y la plataforma de Nokia está en serias dudas de continuar. Java 2 Micro Edition no está diseñado para telefonos inteligentes y BlackBerry está más orientada al mercado profesional, además de su complejidad para desarrollar aplicaciones.

Entre las tres opciones restantes, Android es la que presenta una gama más amplia de terminales (gama baja incluida), permitiendo mayor accesibilidad a la aplicación por todo tipo de público. Es, además, fácil de desarrollar ya que emplea Java, uno de los lenguajes orientados a objetos más sencillos e intuitivos. Si a todo esto le añadimos la gratuidad del desarrollo sobre Android y que es la plataforma más extendida en el campo de los telefonos inteligentes, se convierte sin duda alguna en la plataforma de elección.

2.5.2. Algoritmo e infraestructura

De todas las opciones de localización disponibles, la única opción que tenemos es implementar nuestro sistema sobre las señales GSM que son las únicas disponibles bajo la red de suburbano.

2.5. CONCLUSIONES

De los casos estudiados en los que han empleado unicamente las señales de GSM para la localización, nos basamos en el expuesto en [43]. En este, Varshavsky ha conseguido una precisión de unos cuatro metros con una precisión bastante aceptable a partir de mapas de un metro de granularidad de todas las antenas recibidas.

El método estudiado por Varshavsky nos permite una localización simbólica buena y de bajo coste gracias a la infraestructura ya existente. Si la distribución de antenas es lo suficientemente buena, nos permitiría balancear la precisión y la densidad de las muestras en los mapas.

Infraestructura del Metro de Madrid

3.1. Introducción

Todos los sistemas de localización dependen de su infraestructura. En nuestro caso, en la localización en interiores a través de señales GSM, dependemos de las antenas móviles instaladas. Si bien la infraestructura desplegada en el exterior está bien documentada y hasta hay bases de datos con la localización de las antenas, la información referente a las líneas de metro no está ni tan documentada ni tan disponible entre otras razones por ser de más reciente instalación.

3.2. Historia del metro de Madrid

La red de metro fue inaugurada por Alfonso XIII en 1919 con ocho paradas entre Cuatro Caminos y Sol, con la ahora desaparecida estación de Chamber y casi 4 kilómetros de túnel.

Entre 1920 y 1935 el metro crece hasta los 14,6 kilómetros con la ampliación de la línea 1 hasta Puente de Vallecas y la creación de las líneas 2 y 3 que cubren el trayecto entre la estación de Ventas y Cuatro Caminos y entre Embajadores y Sol respectivamente, además de la construcción del ramal que une Príncipe Pío con Opera. También en esta época se construyó el inicio de lo que ahora es la línea 4, pero que entonces era una ramificación de la línea 2 que iba desde Goya hasta Diego de León.

Durante la guerra civil el metro de Madrid sirvió como refugio para los bombardeos y partes del metro estuvieron cerradas debido al sitio al que estuvo sometida la ciudad

de Madrid.

Una vez finalizada la guerra civil y hasta el año 1954 se amplía la línea 3 para abarcar desde Legazpi hasta Arguelles y se crea la línea 4 desde Diego de León (hasta entonces bifurcación de la línea 2) y Arguelles alcanzando los 60 kilómetros de líneas de metro.

Los siguientes años, hasta finales de los 70, fueron años de grandes planes y necesidades urgentes de ampliar las líneas debido al fuerte crecimiento que estaba experimentando Madrid (doble la población en estas dos décadas). Este crecimiento obliga a ampliar las estaciones para poder emplear trenes de hasta 6 coches pasando de 60 metros de andén a 90, una consecuencia de esto es el cierre de la estación de Chamber de la línea 1 al estar en una curva muy pronunciada y que fuese prácticamente imposible su ampliación.

En esta primera década, hasta 1966, se amplía la línea 2 hasta Ciudad Lineal en lo que en el futuro se pasará a como parte de la línea 5 de metro y se crea la línea 5 con un tramo entre Carabanchel y Plaza de España. Este último tramo posteriormente formará las líneas 10 y 5. Se puede observar el mapa al final de 1966 en la imagen 3.1.

En 1969 se introduce el hormigón armado en el revestimiento de las paredes del metro y la introducción del "método tradicional de Madrid", que es una modificación del "antiguo método Belga", en el que no se emplea tuneladora y se va revistiendo el túnel con hormigón armado apuntalado con perfiles metálicos (en lugar de ladrillo para el revestimiento y apuntalamiento con madera). Por lo tanto todas las líneas de metro hasta 1966 tienen pared de ladrillo.

Estos dos métodos, tanto el Belga como el tradicional de Madrid que se hacen con pico y pala, dan túneles más estrechos que los realizados con tuneladoras. Además son de anchura variable ya que lejos de tener un diámetro fijo, con estos métodos los técnicos se iban adaptando al terreno estrechando el túnel si fuese necesario.

En estos años además, se amplió el gabilo (altura del túnel) en varias líneas para adaptarse a la frecuencia de trenes y estar listos para nuevos modelos.

Las ampliaciones de esta época continuaron creándose la línea 5 con las estaciones ya existentes hasta Aluche, también se construyen las líneas 7 y 6 desde Las Musas hasta Avenida de América y Cuatro Caminos hasta Pacífico respectivamente y se amplía la línea 4 hasta Alfonso XII.

Durante los años entre 1978 y 1985 se ampliaron las líneas que contaban con gabilo ancho (líneas 6, 8 que es la actual 10 y 9), también en este periodo de tiempo se alcanzaron los 100 kilómetros de red de metro.

3.2. HISTORIA DEL METRO DE MADRID



Figura 3.1: Ampliaciones de metro entre 1955 y 1966. En todas estas líneas las paredes están recubiertas de ladrillo en lugar de hormigón.

Hasta finales de los 70 la empresa era de carácter privado pero, debido a los altos costes asociados al gran crecimiento que experimenta el metro en todos estos años que se traducen en pérdidas económicas, se decide nacionalizar la empresa.

En los años venideros, el metro es empleado por los partidos políticos como baza electoral. Prometiendo que ningún madrileño tendrá a más de 600 metros una estación de metro, esta propuesta provoca un crecimiento enorme de la red de suburbano.

Si en 1994 la red alcanzó los 115 kilómetros, en algo más de una década aumentó más de 150 kilómetros hasta convertirse en el sexto metro en número de kilómetros del mundo. Esta nueva red dejó de ser metropolitana para pasar a ser interurbana comunicando a Madrid con municipios como Leganes, Alcorcón y llegando al aeropuerto de Madrid. Durante esta etapa llega a haber hasta 10 tuneladoras de frente cerrado trabajando al mismo tiempo y la mayoría de estas líneas tienen un gabiño más amplio.

Los últimos años, en el periodo entre 1999 y 2007, se ha ido ampliando la red de

metro sobretodo en los municipios colindantes. Así se creó Metrosur uniendo Alcorcón, Fuenlabrada, Getafe, Leganes y Mostoles y Metronorte, Metroeste y Metro Ligero con municipios como Alcobendas, San Sebastián de los Reyes, Coslada y San Fernando de Henares ampliando la red hasta los 284 kilómetros actuales.

3.3. Infraestructura móvil

En el interior de metro solo se encuentra la estación base, las cuales están conectadas a través de fibra óptica proporcionada por Iberdrola que conecta con la OSS situada ya fuera de las instalaciones de metro. Todas las instalaciones del metro, salvo los equipos de las estaciones base, son administradas por la empresa Metrocall que es en parte propiedad de la pública Metro de Madrid y parte de la privada TecnoCom.

Cada estación base, donde las compañías instalan sus equipos, cubre un cierto número de estaciones y de túneles. La distribución de estas estaciones base se puede ver en 3.2, que nos da una idea de las zonas que abarca cada una. En este mapa las estaciones de metro numeradas son las que tienen una estación base. Desde aquí darán cobertura a varios túneles y otras estaciones.

En la línea 10 hemos medido las señales que se reciben (véase 5.3) de potencia y el identificador de antena (*cellid*) presente en cada estación[35]. De estos datos podemos deducir que túneles y estaciones cubren cada estación base de las cuales conocemos su posición (detalle en la imagen 3.3):

Plaza de Castilla: Cubre las estaciones de Cuzco, Plaza de Castilla y Santiago Bernabéu.

Nuevos Ministerios: Cubre Nuevos Ministerios y Gregorio Marañón.

Alonso Martínez: Da cobertura a Alonso Martínez y Tribunal.

Plaza de España: Solo contiene a Plaza de España.

Desde las estaciones base, la señal se transmite por cables coaxiales a las antenas y cables radiantes de las estaciones y túneles.

En los túneles nos podemos encontrar o bien cables radiantes o antenas repetidoras[45]. Esto depende en parte de la estructura del túnel, los más antiguos con pared de ladrillo no dan un anclaje lo suficientemente fuerte para aguantar con seguridad el peso de una

3.3. INFRAESTRUCTURA MÓVIL



Figura 3.2: Lneas de metro con cobertura móvil. Las estaciones base están en las estaciones numeradas. Las líneas verdes son con cobertura del exterior.



Figura 3.3: Detalle del mapa de cobertura de telefonía sobre la línea 10.

antena, además en muchos el gabilo es demasiado bajo como para, aun teniendo hormigón armado, suponer un peligro para el tren al pasar este muy cerca de la antenna. En estos casos una solución práctica es el cable radiante.

Podemos ver el comportamiento tanto del cable radiante en la figura 3.4 como de los repetidores de antena en la figura 3.5 donde la curva azul representa las pérdidas sin tren. En ambos casos la señal se mantiene por encima de los -90 dBm hasta pasados los 300 metros, si bien el comportamiento del cable radiante parece mejor también hay que tener en cuenta que estos datos han sido extraídos de un estudio realizado por un fabricante de cable radiante con datos de los metros de Chicago y Hong Kong[13].

En el caso de instalación de antenas repetidoras de señal a lo largo del túnel, en el metro están colocadas cada 150-200 metros para garantizar un mínimo en la recepción de la señal. Mientras que el cable radiante se dispone a lo largo de todo el túnel con amplificadores cada ciertos metros.

Para determinar la infraestructura de las estaciones basta con bajarse y buscar las antenas, ya que se pueden ver fácilmente buena parte de las que hay instaladas. Nos hemos encontrado tres modelos, todos expuestos en la figura 3.6:

3.3. INFRAESTRUCTURA MÓVIL

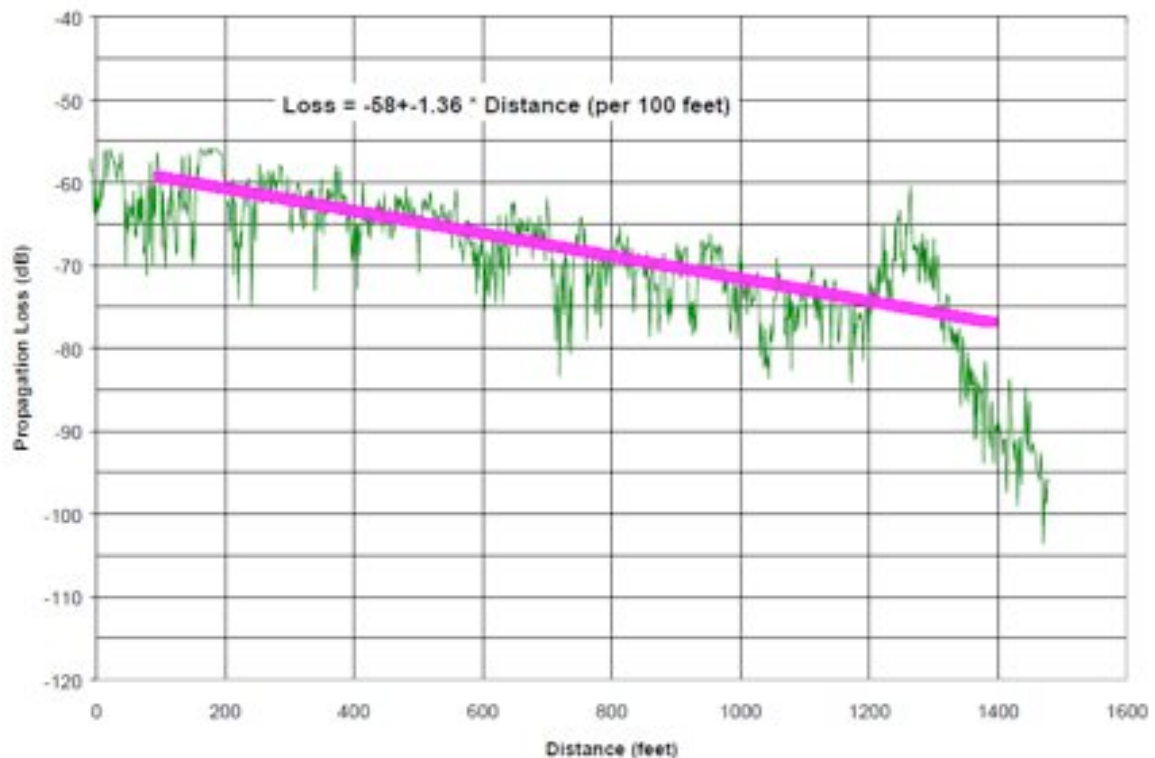


Figura 3.4: Pérdidas de propagación de cable radiante (RCT7-CPUS) en un túnel circular de hormigón armado en la banda PCS (1800-1900MHz).

- Antena bidireccional: se encuentra instalada en todas las estaciones para dar cobertura a los andenes y, por tanto, de las que principalmente recibiremos la señal a la hora de elaborar los mapas. Las podemos ver en el centro de uno de los dos andenes en una pared a una altura de unos dos metros o dos metros y medio.
- Antena unidireccional: utilizada para dar cobertura a algunas partes de los pasillos. Se encuentra en las paredes.
- Antena omnidireccional: se encuentra en los techos de los pasillos, también para dar cobertura a estos. Estas las podemos encontrar sobretodo en intersecciones de varios pasillos o esquinas.

Todas estas antenas, en el fondo son repetidores con el mismo *cellid*. Dan una cobertura muy buena a todas las estaciones y túneles pero también una variación de la potencia recibida de señal muy homogénea en cada estación. Con lo que, salvo en raras ocasiones, obtendremos una potencia muy similar. Siempre y cuando sea bajo las mismas condiciones (gente o presencia de trenes). En estas ocasiones la señal se diferenciará únicamente en los *cellid*.

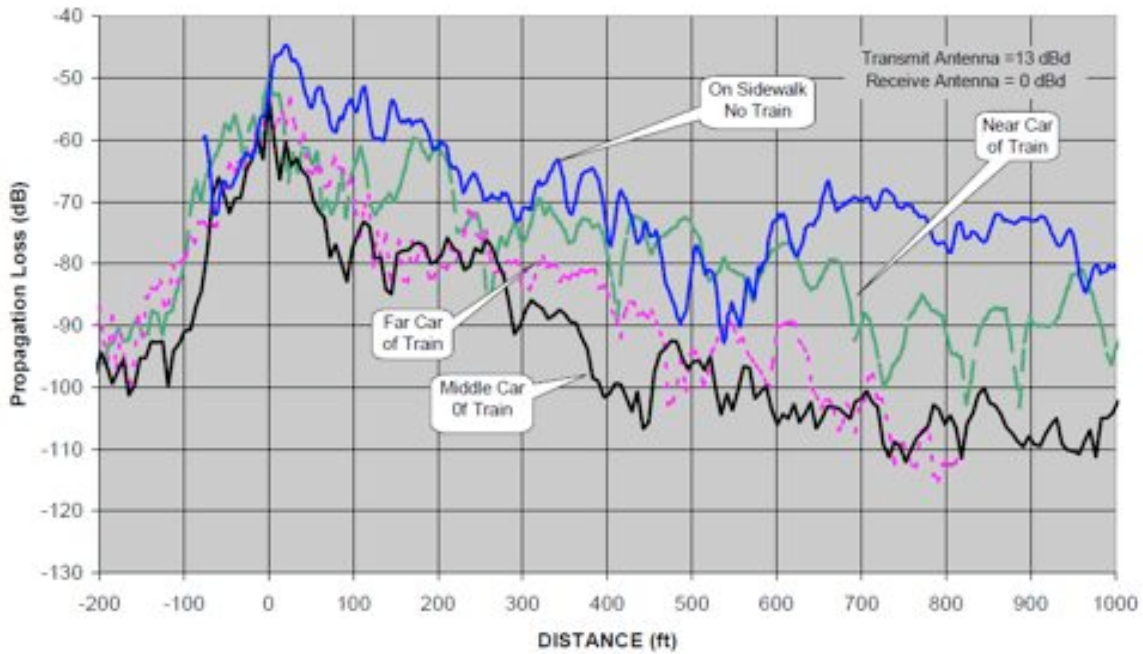


Figura 3.5: Perdidas de propagacion de una antena en un tunel circular de hormigon armado en la banda PCS (1800-1900MHz).

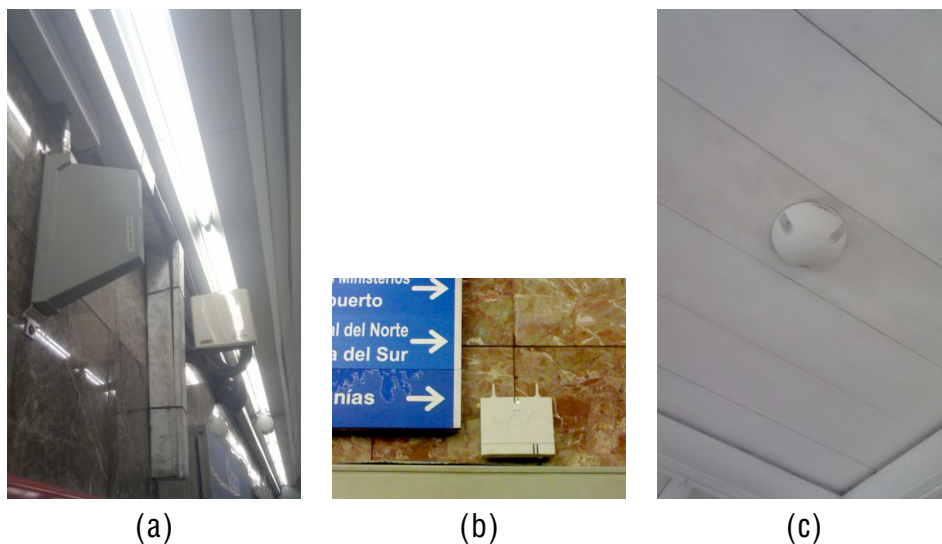


Figura 3.6: a) Antena bidireccional. b) Antena unidireccional. c) Antena omnidireccional.

Diseño de aplicaciones

El desarrollo del sistema se ha hecho sobre la plataforma Android. Para trabajar en ella hemos requerido de la plataforma de desarrollo Eclipse¹, sobre la que se instala el SDK y todas las herramientas necesarias para su programación.

Todo el proyecto, salvo el tratamiento de datos que ha sido sobre Matlab, está escrito en Java.

4.1. Introducción a Android

Hasta hace poco buena parte de la programación sobre móviles era a bajo nivel, diseñando las aplicaciones para un móvil o línea de móviles del mismo fabricante con hardware específico. Android y sus competidores (como Windows Phone o WebOS) son una nueva generación de sistemas operativos diseñados para introducir una capa entre el programador de aplicaciones y el hardware liberando trabajo a estos desarrolladores [15, 50, 37].

Android, dentro de este grupo, se presenta como un sistema operativo de código abierto (*open-source*) y optimizada para móviles por la *Open Handset Alliance*, así como una plataforma para el desarrollo y la publicación de aplicaciones para estos mismos dispositivos en Java. La estructura, funcionamiento y mecanismos de la creación de las aplicaciones se detalla a continuación.

¹<http://www.eclipse.org/>, diciembre 2011

4.1.1. Aplicaciones

Las aplicaciones en Android se ejecutan dentro de una maquina virtual de Java (vease 4.1.4). El sistema operativo es el encargado de iniciarlas, pausarlas y pnalizarlas, en funcion de si la misma aplicacion decide cerrar o bien por falta de memoria.

Por defecto, hay disponibles para el usuario un conjunto de aplicaciones basicas como navegador, cliente de email, administrador de contactos y otros. Todos estan programados en Java y accesibles al programador a traves de la reutilizacion de los componentes de las aplicaciones.

4.1.2. Plataforma de desarrollo de las aplicaciones

Los desarrolladores tienen pleno acceso a la misma API que se ha usado para las aplicaciones basicas de Android. Ademas la arquitectura ha sido diseñada para la reutilizacion de los componentes, cualquier aplicacion puede publicar y ofrecer sus capacidades (siempre sujeto a las restricciones de seguridad de la plataforma) a cualquiera que quiera sacar provecho de la misma. As mismo permite al usuario reemplazar componentes de una aplicacion.

Remarcando que una aplicacion es un conjunto de servicios y sistemas que incluyen:

- Un entorno gráfico muy variado con una gran cantidad de *widgets* disponibles como botones, listas e incluso un navegador web embebido.
- *Content Providers* que permiten a las aplicaciones intercambiar datos entre s .
- Un administrador de recursos (*Resource Manager*), que da acceso a recursos como gráficos o los archivos de las vistas.
- Un administrador de notipcaciones (*Notification Manager*) que permite a cualquier aplicacion mostrar alertas en la barra de estado de Android.
- Un administrador de actividades (*Activity Manager*) que controla la vida de las actividades y un navegador entre aplicaciones.

4.1.3. Bibliotecas

Android incluye una serie de librer as C/C + +, de las cuales algunas de las mas importantes son:

Bibliotecas estándar de C (System C library): una implementación derivada de BSD de la biblioteca estándar de C (*libc*), optimizada para sistemas embebidos basados en Linux.

Bibliotecas multimedia (Media Libraries): para la reproducción y grabación de algunos de los formatos de imagen y audio más populares como MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.

Administrador de superficie (*Surface Manager*): administra el acceso de gráficos tanto 2D como 3D.

LibWebCore: el motor del navegador web.

SGL: motor de gráficos en dos dimensiones.

3D libraries: una implementación de OpenGL ES 1.0, utiliza tanto aceleración hardware como software.

FreeType: renderizado de fuentes.

MiniSQL (*SQLite*): la base de datos relacional disponible para todas las aplicaciones.

4.1.4. Ejecución en Android

Cada aplicación en Android ejecuta su propio proceso, con su propia instancia de la máquina virtual Dalvik (DVM). Dalvik ha sido diseñada para que cada dispositivo pueda ejecutar múltiples VMs de manera eficiente. Para ello, la DVM ejecuta archivos en su propio formato (Dalvik Executable *.dex*), el cual está optimizado para utilizar la mínima memoria posible. Es el compilador de Java el que se encarga de transformar las clases en los *.dex* con la herramienta "dx".

4.1.5. Núcleo Linux

La administración de recursos como el *hardware* (y sus *drivers*), seguridad, administración de memoria o el control de procesos es llevada a cabo por el núcleo Linux 2.6 el cual posee una interfaz de bajo nivel para dar accesos al resto de *software*. Por ejemplo la máquina virtual Dalvik depende del núcleo de Linux para funcionalidades subyacentes como el manejo de hilos y la administración de memoria a bajo nivel.

4.2. Fundamentos de las aplicaciones de Android

Una vez instalada y ejecutada la aplicación en Android, hay que tener en cuenta que cada aplicación se ejecuta en su propio *sandbox*:

- Android es Linux y, como tal, multiusuario. Cada aplicación es un usuario diferente. Cada usuario (aplicación) con sus permisos autorizados en el manifiesto (vease 4.2.2), el objetivo de esto es que cada aplicación tenga únicamente acceso a los recursos que necesita.
- Cada proceso tiene su propia DVM para que cada aplicación este aislada de las demás por seguridad y estabilidad.
- Por defecto, cada aplicación ejecuta su propio proceso de Linux. Se apagará el proceso cuando no sea necesario o el sistema necesite memoria.

No obstante, en caso de necesidad, es posible que dos aplicaciones sean el mismo usuario y también que sean ejecutadas en el mismo proceso compartiendo la DVM para conservar recursos.

4.2.1. Componentes de una aplicación

Son los bloques esenciales de una aplicación. Cada componente es un punto distinto sobre el que el sistema podrá potencialmente acceder. Cada uno es su propia entidad y juega un papel único como bloque para construir y definir el comportamiento de toda la actividad.

Hay cuatro tipos distintos de componentes. Cada uno tiene un propósito distinto y distinto ciclo de vida que define cómo es creado y destruido:

Actividad (*activity*): una actividad representa una pantalla con una interfaz de usuario. Las actividades en el sistema son administradas como una pila de actividades. Cuando una nueva actividad es iniciada, se coloca en la primera posición de la pila y la anterior pasa a segundo plano hasta que la nueva se finalice o pause. Este ciclo de vida de una actividad en la imagen 4.1.

Por ejemplo, un cliente de correo tiene una actividad para la bandeja de mensajes entrante, otra para la lectura de un email en concreto, otra para su redacción...

Servicio (*service*): es un componente que se ejecuta de fondo, para realizar tareas de larga duración, procesos remotos o algún trabajo que requiere del usuario ya

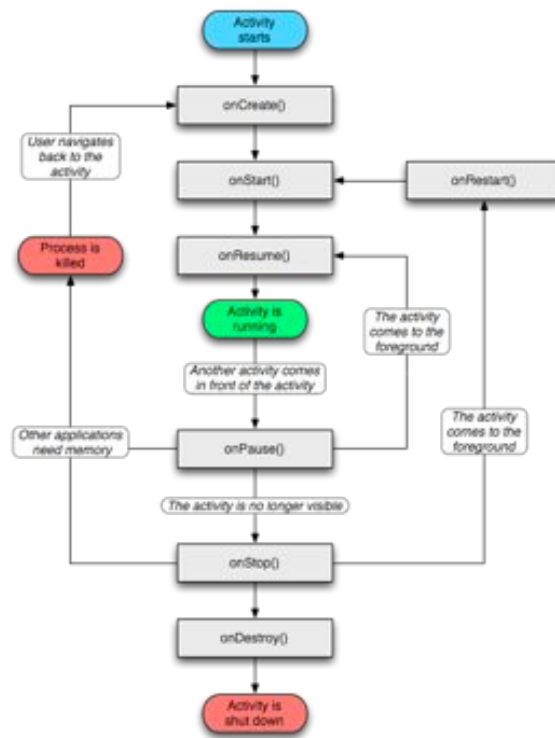


Figura 4.1: Ciclo de vida de una actividad.

que nunca muestran una interfaz de usuario. Un ejemplo ser a un reproductor de musica que reproduce canciones mientras el usuario navega por la lista de reproduccion o ejecuta otras aplicaciones.

Proveedores de contenido (*content providers*): un proveedor de contenido es el encargado de manejar un conjunto compartido de datos de la aplicacion. Se puede almacenar esta informacion en un archivo de sistema, en una base de datos *SQLite* o en cualquier otro almacenamiento persistente a la que la aplicacion pueda acceder. A traves de este componente cualquier otra aplicacion puede solicitar o modipcar los datos almacenados (si el proveedor de contenido lo permite). Como por ejemplo acceder a los contactos para leer o modipcar la informacion de cualquier persona.

Broadcast receivers: componente encargado de dar respuesta a cualquier aviso enviado v a *broadcast*. Estos pueden estar generados por el sistema operativo (que se ha apagado la pantalla, queda poca bater a...) o por una aplicacion (una aplicacion ha detectado que se ha descargado datos y estan disponibles para cualquier aplicacion)

Como se ha mencionado antes, cualquier aplicacion puede iniciar cualquier componente perteneciente a otra aplicacion. Como la captura de una foto con la camara o la

lectura de un código QR que, para no crear una actividad que se encargue de ello, se puede usar la aplicación que facilita Android o, en el caso del código QR el componente que se encarga de ello en la aplicación *Barcode Scanner*. Este componente parecerá que forma parte de tu aplicación.

4.2.1.1. Activando componentes

Tres de los cuatro componentes (salvo los proveedores de contenido) son activados por mensajes asncronos denominados (*intent*). Los *intent* atan los componentes entre sí pertenezcan o no a tu aplicación.

Para actividades y servicios, el *intent* define la acción a realizar y la interfaz de usuario sobre la que se interactúa.

Para *Broadcast receivers*, el *intent* simplemente define el tipo de noticia que va ser recibido.

En el caso del cuarto tipo, proveedor de contenido, es activado por una solicitud procedente de un *ContentResolver*. Este último se encarga de manejar todas las transacciones para que el componente que maneja al proveedor de servicio no necesite hacerlo y, en su lugar, simplemente llame a métodos del objeto *ContentResolver*. Esto permite crear una capa de abstracción entre el proveedor de contenidos y el componente que solicita la información (por seguridad).

En resumen, se pueden iniciar los distintos componentes:

- Una actividad, servicio o *broadcast* (en este caso no es ningún proceso sino solo información) a través de un objeto tipo *intent*.
- Un proveedor de contenido haciendo una solicitud a un *ContentResolver*.

4.2.2. *Manifest*

A la hora de lanzar un componente de cualquier aplicación, el sistema operativo debe conocer qué componentes contiene y va utilizar así como la versión mínima de la API sobre la que se ha desarrollado. Esto se indica a través del manifiesto (*AndroidManifest.xml*) de la aplicación. Este es un archivo en el que el desarrollador declara los recursos que va a utilizar. A la hora de instalar la aplicación, el usuario debe aceptar los permisos que el desarrollador ha incluido en el manifiesto. Estos permisos pueden ser, por ejemplo, acceso a internet, a los contactos o acceder a la tarjeta de memoria.

4.2.2.1. Declarar componentes

La tarea básica que debe hacer el manifiesto es informar sobre cada componente presente en la aplicación. Por ejemplo la figura 4.2.

```
<activity android:name=".PowerMeasurement" android:label="@string/app_name">
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter></activity>
```

Figura 4.2: Ejemplo de declaración de la actividad *PowerMeasurement* en el manifiesto

Actividades, servicios y proveedores de contenidos no declarados en el manifiesto no serán nunca ejecutados aunque estén incluidos en el código fuente de la aplicación. No obstante *broadcast receivers* pueden ser creados dinámicamente y registrados en el sistema a través de la llamada a la función *registerReceiver()*.

4.2.2.2. Declarar capacidades de los componentes

Como se ha expuesto en 4.2.1.1, se puede emplear un *intent* para iniciar actividades, servicios y *broadcast receivers*. No obstante, la verdadera utilidad de los *intents* radica en el concepto de *intent action*. A través de las cuales simplemente describes el tipo de acción a realizar y dejar que el sistema se encargue de encontrar el componente que sea capaz de realizar esta acción concreta y la inicia. Si hubiese varios componentes, el sistema simplemente da a escoger al usuario cuál prefiere utilizar. Estas asociaciones se realizan a través de las declaraciones en el manifiesto.

4.2.2.3. Declarar requisitos de la aplicación

Hay una gran diversidad de dispositivos que utilizan Android, además de distintas versiones según se vayan actualizando o no. Para evitar que la aplicación sea instalada en dispositivos que no cuentan con los componentes necesarios para el correcto funcionamiento o que simplemente tienen una versión de Android inferior a la que se utilizó para el diseño de la misma se debe especificar estos dos aspectos: versión y componentes, si bien en cuanto a los componentes se puede determinar que usa un elemento del hardware pero no es imprescindible.

Se debe tener en cuenta:

- Tamaño de la pantalla y densidad de píxeles. Así Android se puede encargar de escalar los interfaces de usuario en función de la pantalla del dispositivo o, si se prefiere, se pueden diseñar específicamente para algunas resoluciones determinadas.
- Configuración de entrada del dispositivo: como botones o *trackball*. Se puede especificar si se necesitan alguno de ellos en el dispositivo.
- Características del dispositivo: la potencia del procesador o sus capacidades gráficas. También otros componentes como la cámara, el *Bluetooth* o el tipo de pantalla táctil.
- Versión de Android. Esencial, ya que esta define el nivel de la API utilizada. Pueden no existir métodos, objetos o librerías enteras en versiones más antiguas.

4.2.2.4. Context

Context es un interfaz que da acceso a la información global del entorno de la aplicación. Es una clase abstracta de cuya implementación se encarga el sistema Android y permite acceder a recursos específicos y clases y realizar solicitudes de operaciones: iniciar actividades, mandar y recibir *intents*, etc..

En nuestro caso, esta clase permite también acceder al estado del teléfono y las comunicaciones. De tal manera que a través de esta clase solicitamos al sistema la creación del objeto *TelephonyManager*, el cual contendrá toda la información sobre los servicios telefónicos: intensidad de señal y la antena a la que está conectado el dispositivo.

4.3. Aplicaciones desarrolladas

Como el algoritmo elegido requiere de dos etapas (véase capítulo 5), hemos desarrollado dos aplicaciones que se basan en un parte de código común.

El primero de los programas permite obtener y almacenar sucesivas mediciones de la potencia GSM en un punto determinado. Combinando diferentes medidas, la aplicación permite realizar los mapas de potencias.

El segundo se encarga de posicionar al usuario empleando los mapas obtenidos por el primer programa.

4.4. ACTIVITY POWERMEASUREMENT

Como hemos visto todas las aplicaciones en *Android* se dividen en los distintos bloques que son las actividades. En nuestro caso estos dos programas comparten la actividad *PowerMeasurement*:

PowerMeasurement: Esta actividad consiste en un menu que permite lanzar cualquiera de las dos aplicaciones. As , se pueden tomar muestras o calcular la posicion. Esta actividad se ha incluido por comodidad a la hora de realizar las pruebas. No esta previsto que el usuario final pueda realizar los mapas de potencias, si no que estos son obtenidos por personal tecnico ya que posteriormente requieren un procesado.

Reading: Esta es la actividad del primer programa encargado de tomar las muestras de potencias para elaborar un mapa o simplemente para hacer un estudio de las señales.

Position: Este programa es el encargado de calcular la posicion en funcion de un mapa de potencias calculado previamente.

Las dos ultimas actividades, toman la informacion sobre las señales que recibe en ese instante el movil de un servicio que esta recibiendo continuamente el estado de la señal del movil.

En la pgura 4.3, se puede observar la relacion completa de las clases de la aplicacion.

4.4. Activity PowerMeasurement

Es la actividad que se inicia al abrir el programa, como todas las actividades tiene una vista asociada. En este caso es *menu.xml*. Escucha los eventos generados por los tres botones como se ve en la captura 4.4. Segun la opcion elegida, lanza la actividad correspondiente a traves de su *intent*.

4.5. Service SampleReader

Este es el servicio que sirve las muestras al resto de actividades. Se ejecuta (aunque la clase *Service* en Android propiamente dicha no es ni un hilo ni un proceso a parte de la actividad que lo lanza) por detras tomando la potencia recibida cada segundo

Class Diagram

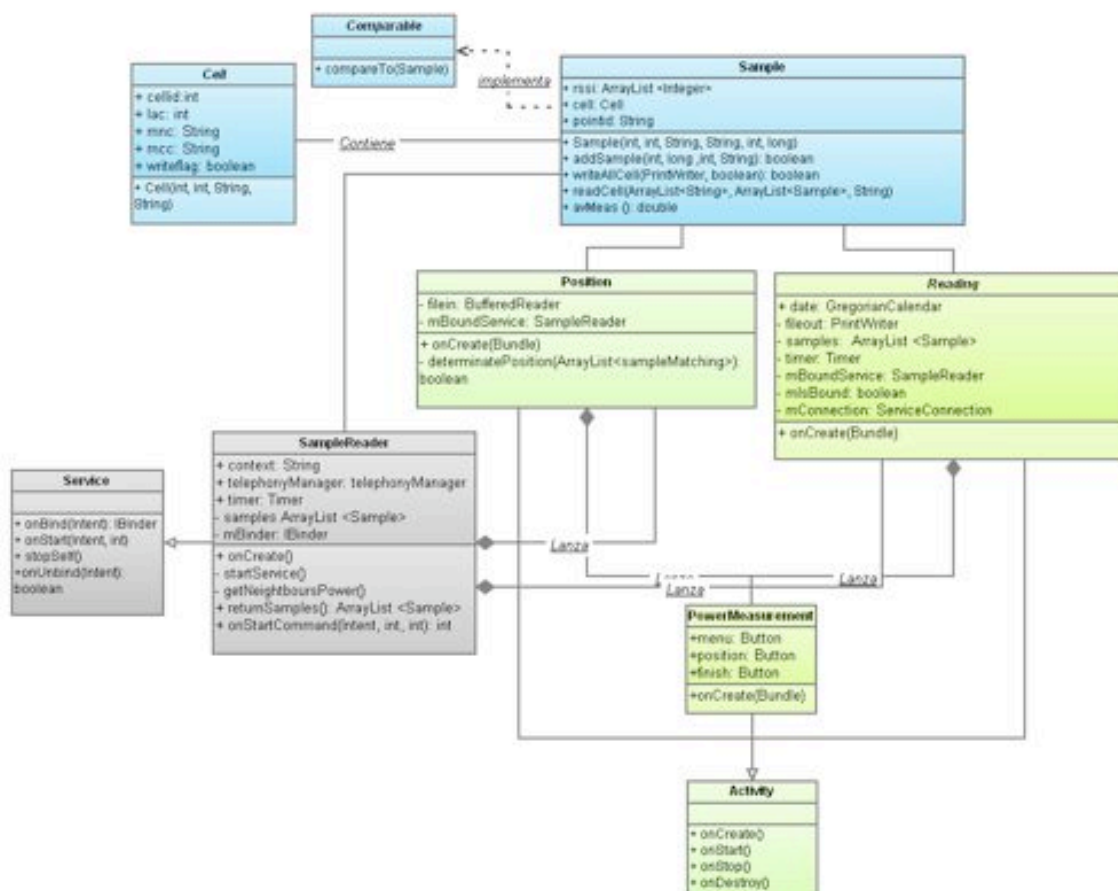


Figura 4.3: Diagrama UML de todas las clases de la aplicación. PowerMeasurement es la encargada de lanzar cada una de las actividades de uno u otro programa.

y guardandola en una *ArrayList* de tipo *Sample* que es lo que despues devuelve a las actividades.

El servicio lanza un *timer* con un un tiempo de ejecucion cada 1000 milisegundos. Cada vez que se ejecuta, la actividad emplea *context* (vease la seccion 4.2.2.4), en concreto utiliza *context.TELEPHONY_ SERVICE* que es la relacionada con toda las opciones de telefon a.

Ademas de comprobar cada segundo la potencia de la señal, este servicio implementa un *listener* que monitoriza cambios en la potencia de la señal, en cuyo caso actualiza el *ArrayList* con las nuevas potencias.

A la hora de obtener potencias, el programa utiliza solo las señales recibidas de GSM. Ademas cabe destacar que durante los primeros meses de trabajo utilizamos el

4.5. SERVICE SAMPLEREADER



Figura 4.4: Captura de la vista de la actividad PowerMeasurement. El diagrama de UML se puede ver en el completo (figura 4.3).

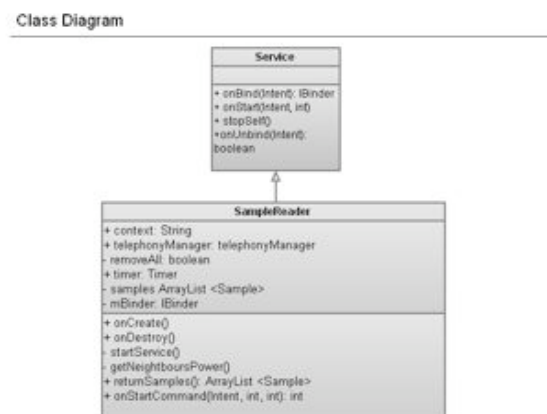


Figura 4.5: Diagrama UML en detalle de SampleReader.

modelo HTC magic, el cual durante esa época utilizaba todavía Android 1.6 y, en esa versión, solo daba información de la potencia en decibelios sobre las antenas vecinas y no sobre la que estaba conectado, de la que solo daba información de la calidad de señal buena, media o mala.

Cabe destacar, por último, que cada vez que se solicitan las muestras al servicio, este elimina todas las muestras antiguas para no enviarlas varias veces.

4.6. Activity Reading

Esta actividad permite al usuario tomar muestra en una localización en particular. Los métodos y servicios que incluye se resumen en la figura 4.7. Para ello lanza el servicio *SampleReader* con el que está conectado y recoge las potencias leídas por este.

Para estos dos objetivos esta actividad se ha diseñado con dos métodos de muestreo, el primero es el modo normal en el que durante el tiempo de muestra se recogen todos los valores para guardar únicamente la media de todas las medidas (el empleado para realizar los mapas) y el segundo o modo granular en el que se guarda una muestra por segundo. Este segundo método ha sido utilizado para hacer pruebas de potencia.

La interfaz de usuario de esta actividad contiene los siguientes elementos (véase la figura 4.6):

1. EditText: para introducir un texto que será o bien un evento para el modo granular o bien el nombre con el que se desea guardar el archivo de muestras.
2. Botones "Add event" y ".txt name": para registrar un evento o registrar el nombre del archivo respectivamente. En el primer caso el texto introducido se incluye directamente en el archivo a salvar.
3. Botón "Write Sample": guarda la muestra actual en el archivo e inicia una nueva muestra.
4. Botón "Stop Reading": guarda la muestra, detiene el muestreo y cierra el programa.
5. Spinner o menú desplegable: modifica el tiempo de la alarma automáticamente para la opción de 30 segundos, uno, dos y medio, cinco minutos o desactivada. El objetivo de esta alarma es avisar al usuario para obtener un tiempo por muestra constante y homogeneizar al máximo el mapa de muestras.
6. RadioButton "Granular mode": para activar el modo granular y no tomar la media de las muestras sino una cada segundo.



Figura 4.6: Captura de Reading.

Nada más arrancarse la actividad, se lanza *SampleReader*, activa el temporizador de la alarma y se enlaza el servicio. En el momento que se pulsa cualquiera de los botones o se modifica el Spinner se realiza la actividad correspondiente, todas explicadas anteriormente.

La escritura de muestras, tanto en el modo granular como el normal, se activa cuando se pulsa uno de los botones "Write Sample" o "Stop Reading". En ese momento borra las muestras antiguas de la variable *Sample* (ya escritas) y añade todas las muestras a esta variable. Comprueba si está activado el RadioButton y graba o bien la media de las muestras tomadas, o bien todas las muestras de manera secuencial. Si es "Stop Reading" para el servicio, incluye la línea "Fin de lecturas" en el archivo y detiene la actividad.

Class Diagram

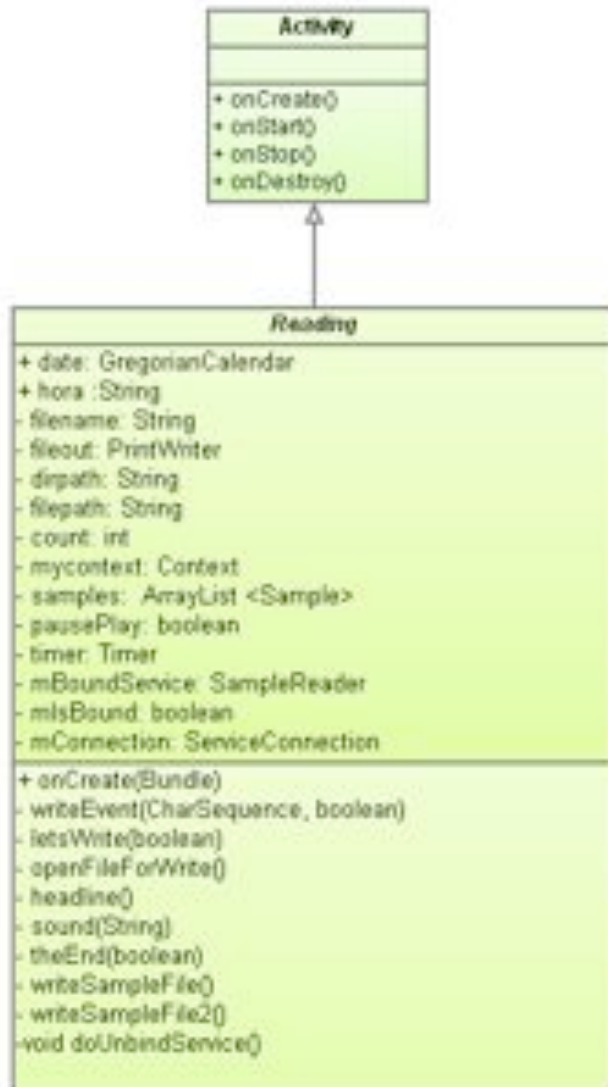


Figura 4.7: Diagrama UML en detalle de Reading.

4.7. Mapa de muestras

En el apartado 4.6, se explica como se toman las muestras para realizar el mapa. Las muestras quedan guardadas en un archivo de texto. Este archivo se estructura con una cabecera y las muestras seguido de un "ñ de muestra" y, si fuese el caso, un "ñ de lecturas" que indican el ñ de la muestra o del archivo respectivamente.

4.7. MAPA DE MUESTRAS

Los datos de la muestra se apuntan en una sola línea separando mediante espacios las distintas huellas de señal (*fingerprints*). El orden y los tipos de datos que aparecen en el mapa están explicados en la tabla 4.1.

first_ TimeSample	Tiempo en milisegundos en que se tomó la primera muestra del reloj interno del móvil.
last_ TimeSample	Tiempo en milisegundos en que se tomó la última muestra del reloj interno del móvil.
cellid	Identificador de antena. Es el identificador principal que usamos para distinguir antenas.
lac	Location Area Identity. Es un identificador de área para las señales GSM, utilizado para determinar la posición del usuario.
mcc	Mobile country code. Identificador del país.
mnc	Mobile network code. Es un identificador de compañía, el problema que tiene es que, en ocasiones, Android lo devuelve como nulo.
num_ Samples	Dado que el rssi es una media de las muestras tomadas a lo largo de un minuto, este valor indica el número de muestras entre las que se ha hecho la media y nos da una idea de la frecuencia de aparición de la antena.
rssi	Potencia leída por el móvil. En el mapa será la media obtenida a lo largo del minuto de muestreo.

Tabla 4.1: Tabla con los campos disponibles en el mapa de muestras.

Podemos ver en la figura 4.8 un ejemplo de una muestra con una única huella de señal.

La cabecera está formada por tres líneas. La primera indica la hora y la fecha en la que se tomó la muestra, la segunda es el identificador de la muestra, la tercera es el enlace que asocia al área y, por último, la cuarta es una cabecera sobre el orden de los datos de la muestra.

A continuación aparecen los datos propiamente dichos. Las dos cifras de tiempo en milisegundos, seguido de los datos de identificación de la antena. La potencia sale la última. Si hubiesen más huellas de señal estas seguirían a continuación en la misma línea.

Este mapa es procesado posteriormente para agrupar las distintas muestras y reunirlos en un único fichero. Dado que generalmente se van a realizar los mapas en varias fases, el muestreo quedará separado en varios archivos. Para unirlos se copian todas las muestras, con cabecera incluida en el mismo fichero y se añade "Fin de lecturas" al

```

12_at_22_3_201119h24m
castilla_hospital3
link castilla
first_TimeSample last_TimeSample cellid lac mcc mnc num_Samples rssi
83536193 83965198 263 2835 07 214 510 -79
Fin de muestra

```

Figura 4.8: Ejemplo de una muestra con dos *fingerprints*

pnal de este.

4.8. Activity Position

El objetivo de esta actividad es determinar la posición calculando la distancia euclídea con respecto al mapa de muestras.

Para ello primero lanza el servicio *SampleReader*, el cual al igual que en la sección anterior está conectado a nuestra actividad, para obtener la potencia actual que recibe el teléfono. Esta la almacena en un *ArrayList* de tipo *Sample*.

Una vez lanzado el servicio, se abre el *plancher* con el mapa de muestras y se almacena en otro *ArrayList* tipo *Sample*. Además, según se van leyendo las muestras, se añade en la clase privada *sampleMatching* el *pointid* y el *link* de la muestra. Este última clase se emplea para la clasificación de muestras en las distintas estaciones o áreas. Además, implementa *Comparable* para poder implementar posteriormente el método de posicionamiento por puntuación.

Una vez almacenadas las muestras del mapa, se inicia la función *determinatePosition*, que es la encargada de comparar las muestras del mapa con las actuales recibidas servicio. En ella, lo primero que se hace es llamar al servicio *SampleReader* para actualizar las potencias recibidas. Estas nuevas muestras del servicio se guardan como el mapa en un *ArrayList* de la clase *Sample*. Este proceso se puede ver en la figura 4.10

A continuación compara el *cellid* de cada muestra estos dos *ArrayList* y guarda la posición de las coincidencias de los dos *ArrayList* en el tipo de dato *sampleMatching*. Además, dado que el mapa de muestras (vease la sección 4.7) está dividido en distintos sectores, también se guarda el máximo número de coincidencias. Este tamaño será el tamaño del subespacio vectorial. Es decir, si hay coincidencias con antenas de tres sectores habiendo 4 coincidencias con en el primer sector y 3 en el segundo y tercero, la distancia euclídea se calculara sobre un subespacio de dimensión cuatro.

4.8. ACTIVITY POSITION

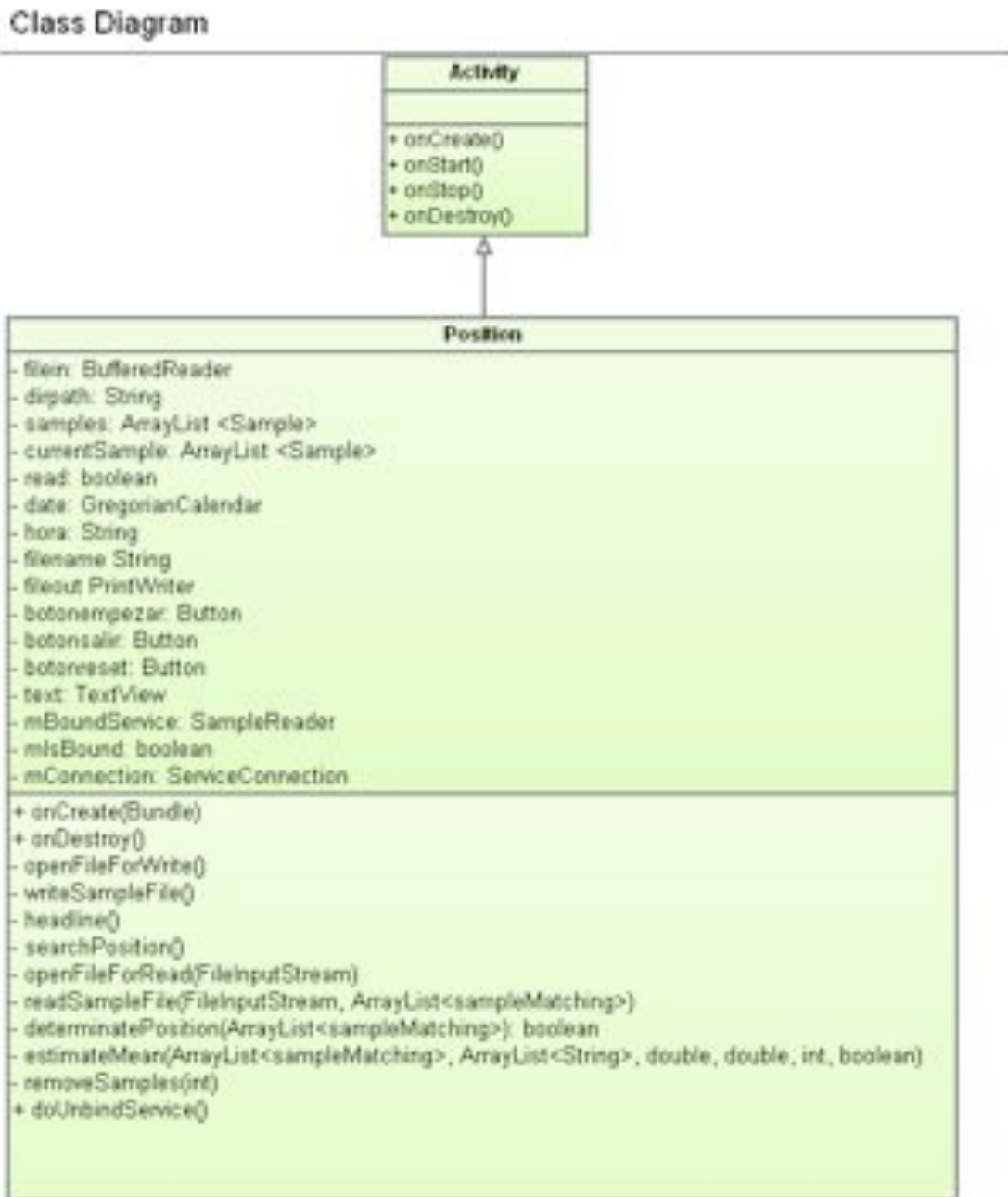


Figura 4.9: Diagrama UML en detalle de la actividad Position.

Todos los resultados se envían a la pantalla a través de *TextView*, que es una clase básica de los *widgets* de las interfaces Android que hereda de *View* y permite mostrar y editar textos. Podemos ver una doble captura de un resultado en la figura 4.11.

Una vez realizados los cálculos, actualiza la pantalla y espera a que se detenga el

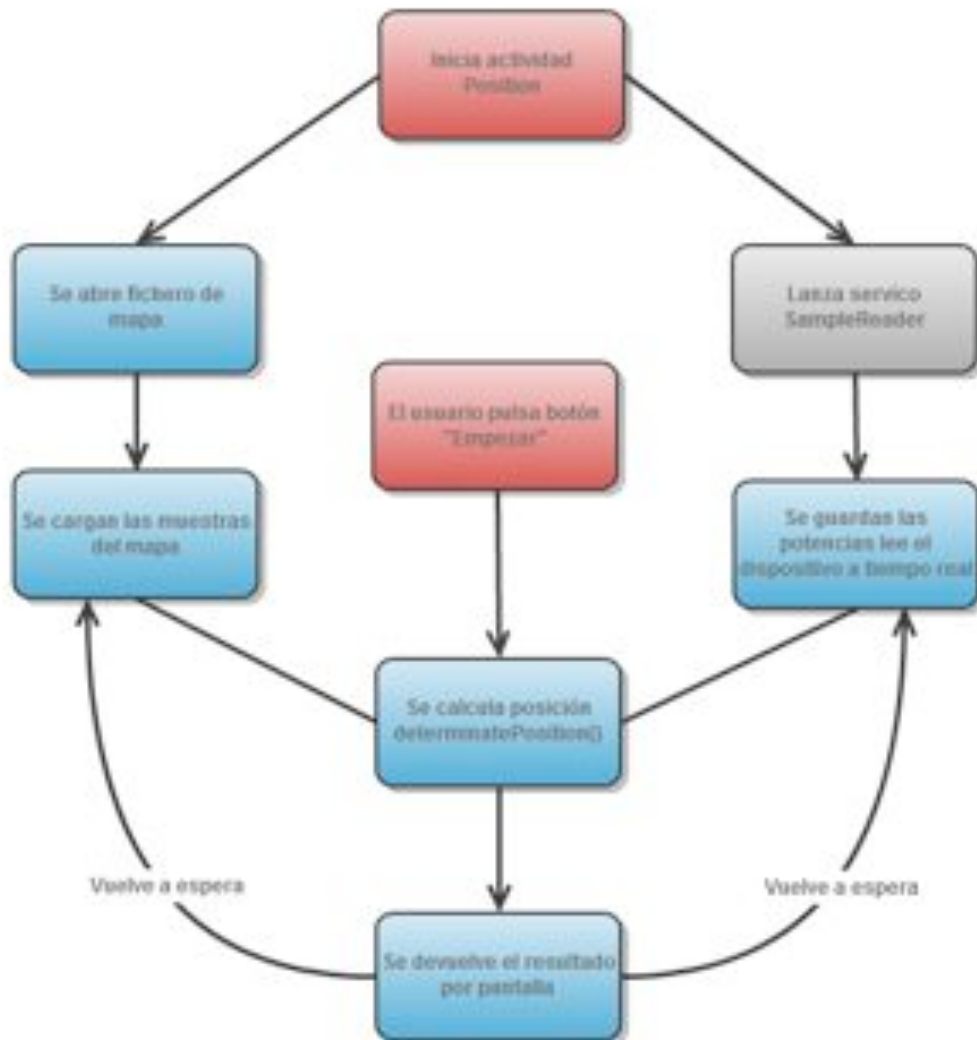


Figura 4.10: Diagrama de flujo simplificado para la actividad Position.

programa o se solicite calcular de nuevo la posición.

4.9. Clases auxiliares

En este apartado se explican las clases que se encargan del almacenamiento de datos. Estas clases son básicamente las que contienen las muestras de potencia y las que las



Figura 4.11: Dos muestras de una localización. Debajo de los botones aparece el texto con todas las distancias calculadas de cada uno de los puntos.

administran. Además son clases que no incluyen ningún elemento de la API de Android, solamente de las librerías de Java.

4.9.1. Cell

Hay dos clases que se han creado públicas y una privada. La primera que es `Cell`, hereda de `Object` y contiene los datos básicos que podemos obtener acerca de una celda o antena GSM:

- `int cellid`: Contiene el `cellid` de la antena. Puede ser un entero de hasta 7 dígitos.
- `int lac`: Otro entero de hasta 7 dígitos. Contiene el Location Area Code, código utilizado por las operadoras para actualizar la posición de los usuarios junto con el MNC y el MCC. Al ser un número de 16 bits se puede distinguir 65536 áreas distintas con un MCC y MNC comunes.
- `String mnc`: Mobile Network Code que, en combinación con el MCC, se utiliza

para identificar al operador que nos está dando servicio. Serán un máximo de tres dígitos

- String `mcc`: Mobile Country Code, otro entero de tres dígitos, nos indica el país.
- boolean `write_log`: bandera que se ha utilizado en algunas pruebas para indicar si estaba escrito o no. Actualmente se ha dejado de usar. Pero se deja ya que puede resultar útil en el futuro. Tampoco se han desechado los métodos que la utilizan, aunque al término de este proyecto no se utilicen.

4.9.2. Sample

A esta clase se accede siempre desde `Sample`. Esta clase es la encargada de todas las actividades relacionadas con los datos, incluida la escritura en archivos. Desciende de `Object` e implementa `Comparable`.

Los datos que almacena `Sample` son:

- `ArrayList Integer rssi`: una lista que contiene todas las muestras de potencia que vamos tomando de la celda en cuestión.
- `ArrayList Long timeStamp`: cada potencia tomada tiene asociado el instante de tiempo en el que se ha tomado.
- `long firstTimeStamp`: el primer instante de tiempo en el que se ha tomado la muestra. Este dato ha resultado útil en el análisis de potencias, ya que nos da ayuda a calcular la frecuencia de aparición de una determinada antena.
- `long lastTimeStamp`: último instante de tiempo en el que se ha tomado la muestra, complementa al dato anterior.
- `int num_Samples`: número total de muestras que se han tomado de la celda
- `Cell cell`: utiliza la clase anterior como núcleo de información de la muestra.
- `String pointid`: identificador de la antena, en el programa toma el nombre del área seguido de un número.
- `String link`: se utiliza como alias para agrupar distintas muestras en distintos áreas/sectores como puede ser una habitación o una estación.
- `int priority`: se coloca para un par de pruebas dando prioridad manualmente a unas antenas antes que a otras. No se empleó inicialmente ninguna variación en este sentido, debido a las pocas señales que se recogen en el metro.

Esta clase gestiona todo lo relacionado con la muestra y cuenta con los siguientes metodos:

- **Sample:** hay tres constructores, en funcion del numero de datos que se le introduzcan, uno de ellos vac o.
- **addSample:** añade nuevas potencias a la muestra y, de nuevo, hay tres metodos distintos.
- **existElement:** busca coincidencias con un *ArrayList* de *Samples* que se introduzca. Se puede buscar en todos los elementos o a partir del numero introducido.
- **matchedElements:** devuelve un array con todos los elementos del *ArrayList* que se le pasa que coinciden con el cellid del *Sample*. Esto se usa en la localizacion cuando tenemos el mapa que es un *ArrayList* con numerosas muestras con el mismo cellid y es necesario buscar las coincidencias.
- **writeCell:** recibe un *PrintWriter*, en el que escribe con el formato indicado en ?? la muestra pero solo un rssi.
- **writeAverageCell:** tambien recibe un *PrintWriter* en el que en este caso, calcula la media de todos los rssi que tenga la clase anotados.
- **readCell:** recibe un *String*, que corresponde a una unica muestra y se encarga de dividirlo e introducirlo. Necesita recibir tambien el pointid para poder o bien añadir la muestra a nuestro *Sample* o bien crear uno nuevo.
- **compareTo:** para ordenar en funcion de la intensidad de potencia en la muestra. En posibles algoritmos, las muestras mas utiles deber an ser las que reciban una potencia mayor y mas clara.
- **avMeas** y **avMeasArit:** calculo de la media de las potencias para su escritura.

Algoritmo de localización en interiores

En este capítulo se presentan el algoritmo utilizado y los mapas de potencia junto a las pruebas realizadas a lo largo del proyecto necesarios para el funcionamiento del algoritmo. El cual requiere unos costes mínimos y, para ello, emplear infraestructuras ya desplegadas que en este caso consiste en la red de GSM del Metro de Madrid.

5.1. Algoritmo de posicionamiento

Para realizar la localización se compara la potencia recibida de cada una de las antenas en el dispositivo móvil con un mapa de distribución de potencias. Los mapas incluyen múltiples muestras de cada estación, estas muestras se denominan huellas de señal, que después se comparan con la potencia recibida a la hora de la localización. La comparación consiste en calcular la distancia euclídea de la señal recibida con cada huella de señal del mapa.

Este es un método alternativo a la triangulación que tiene como ventaja un aumento en la precisión [55] y no requiere información previa sobre la infraestructura como posición exacta de las antenas. Por lo tanto, el método desarrollado necesitará una etapa previa para la realización del mapa y capacidad de procesamiento suficiente que dependerá del tamaño de los mapas.

Por lo tanto, de los algoritmos explicados en 2.2.1, hemos empleado el de huellas de señal (*fingerprints*). Este se basa en la potencia recibida de todas las señales GSM que podemos detectar. Tiene una relación muy directa entre la precisión y el número de antenas que es capaz de detectar el dispositivo, aun siendo la potencia de algunas de ellas muy baja [55].

5.1.1. Mapa de muestras

A la hora de realizar los mapas, en lugar de establecer un unico mapa y aproximarnos a una huella de señal o a varias que esten colindantes entre s , hemos dividido el mapa en zonas. Cada zona contiene un numero determinado de huellas de señal y devuelve como posicion una de las zonas:

Zonas: son un conjunto de huellas de señal que caracterizan un espacio concreto. Bien puede ser una habitacion, una estacion o un espacio no delimitado físicamente pero sera lo que devuelva como posicion el algoritmo de localizacion. La suma de todas las zonas constituira el mapa.

Huellas de señal: cada zona del mismo mapa debe tener el mismo numero de huellas de señal. Cada huella de señal es la potencia media entre todas las muestras tomadas en un punto a razon de una muestra por segundo. Una muestra se compone de la potencia instantanea de todas las señales recibidas (referido como RSSI), as como una serie de datos referentes a la antena de la señal (vease 2.3.3).

Hemos intentado mantener la homogeneidad en el metodo de toma de huellas de señal. Esto es, emplear el mismo tiempo y la misma disposicion espacial para cada huella con el objetivo de que las interferencias con la potencia recibida sean las mismas para todas las huellas de señal tomadas. El momento del día no importa tanto, siempre y cuando las condiciones de alrededor no var en (por ejemplo en el metro la cantidad de gente o que haya o no trenes en la estacion), ya que la variacion de potencia recibida en el exterior en funcion del tiempo a lo largo del día es poco signipcativa [55] como se puede ver en la grafca 5.1.

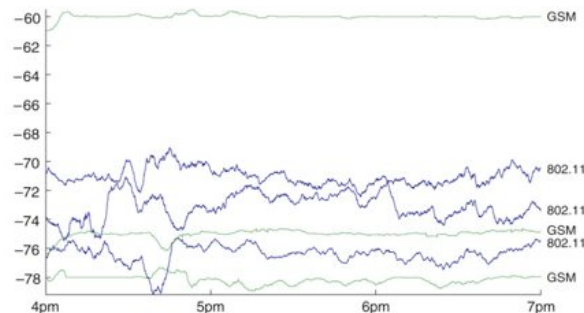


Figura 5.1: Estabilidad de la señal en dBm en funcion del tiempo para señales GSM y Wip [55].

5.1.2. Localización a partir del mapa

Como se ha dicho en el apartado anterior el algoritmo no devuelve coordenadas, sino una localización simbólica. Para ello, calcula la distancia euclídea a todas las huellas de señal de una zona y la media de todas las distancias obtenidas en esta zona. Esta es la razón por la cual es importante que cada zona tenga el mismo número de huellas de señal.

Tenemos por lo tanto un espacio subvectorial de varias dimensiones. El número de dimensiones vendrá determinado por el número de señales (o antenas) detectadas por el dispositivo móvil en el momento del cálculo de la posición y cada señal una de las coordenadas del subespacio. Los valores que puede tomar cada una de estas componentes viene dado por la TS 27.007 825 [3] con la potencia mínima de -113 dBm y un máximo de -51 dBm.

La distancia se calcula siguiendo la fórmula

$$D_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}$$

donde

$$x_i, y_i, z_i$$

son los componentes del vector de potencia de las i huellas de señal que coinciden en *cellid* con $x; y; z$ que son los componentes de potencias del vector de potencia que recibe el móvil en ese momento. Además si una huella de señal no contiene uno de los componentes del subespacio que forma la señal recibida (por ejemplo que una huella no tiene la señal que corresponde con la componente x) se le asigna a ese componente un valor de 113.

Finalmente se establece la posición a partir de todas las distancias euclídeas. Para determinar la posición a partir de estas se han implementado tres métodos:

- Puntuación: Se clasifican cada una de las distancias de cada sección por tamaño, cada distancia entonces se convierte en su correspondiente número natural (la distancia más pequeña el 1, la segunda el 2 y así sucesivamente) y el que obtenga una puntuación menor tras sumar todas las posiciones es la zona seleccionada.
- Media geométrica: En este caso se realiza la media geométrica de todas las distancias de cada zona.
- Media aritmética: Al igual que el anterior, se calcula la media aritmética de todas las distancias, siendo la elegida la menor.

La posición que se consideraba como buena es la que devolvían dos de los tres métodos o, si todos divergen, se toma como correcto el de la media aritmética.

Pongamos como ejemplo que queremos implementar el algoritmo en una casa distinguiendo entre tres habitaciones. Se realiza un mapa con huellas en las tres habitaciones como la mostrada en la figura 5.2: la habitación roja, la verde y la azul. En las tres habitaciones se reciben tres señales de las mismas tres antenas (para una representación más sencilla de los vectores en subespacios de tres dimensiones), con lo que obtenemos un subespacio de tres dimensiones igual para todas las zonas.

Aunque en este ejemplo hemos tomado solo tres habitaciones con tres antenas recibidas, no hay limitación entre el máximo número de habitaciones y zonas y número mínimo o máximo de antenas recibidas. Tan solo la capacidad del dispositivo móvil de procesar los datos.



Figura 5.2: Ejemplo de un mapa en una casa ficticia. Cada zona tiene las huellas de señal de un color diferente

A continuación nos situamos en la habitación azul y calculamos las distancias. En la figura 5.3 están representados una huella de señal por cada habitación junto al vector representante de la señal recibida en magenta y en negro el vector distancia entre ambos, del cual se calcula el módulo y se compara. Una vez hecho esto para las 9 muestras de cada habitación se procede con las tres clasificaciones.

Pongamos que los valores de las distancias han sido para la habitación azul

(0;1;4;0;1;4;4;4;5)

para la habitación roja

(4;4;4;2;6;6;8;8;8)

y para la verde

(3;3;3;5;6;6;7;10;11)

. En el caso de la media aritmetica el valor obtenido entonces para la media aritmetica de la habitacion azul

$$\bar{X}_a = \frac{\sum_{i=1}^9 D_i}{9} = 2,6$$

y de las habitaciones roja $\bar{X}_r = 5,6$ y verde $\bar{X}_v = 6$ que nos dar a como posicion la habitacion azul.

En el caso de la media geometrica obtendr amos para la habitacion azul

$$\bar{X}_a = \sqrt[9]{\prod_{i=1}^9 D_i} = 1,1$$

y para las habitaciones roja $\bar{X}_r = 5,1$ y verde $\bar{X}_v = 5,4$, que nos dar a tambien la habitacion azul. Aunque es poco probable que de distancia cero en las pruebas reales, cuando esto pasa se sustituye por un numero pequeño. En este caso 0.1.

Por ultimo se dan puntuaciones a los valores de tal manera que la habitacion azul quedar a con (1;2;5;1;2;5;5;5;6) con un total de 32, la habitacion roja (5;5;5;3;7;7;9;9;9) con una puntuacion total de 59 y la habitacion verde (4;4;4;6;7;7;8;10;11) con una puntuacion total de 61. Que nos dar a la habitacion azul.

En este caso los tres metodos coincidir an acertadamente en localizarnos en la habitacion azul.

5.2. Estudio preliminar

Antes de probar directamente la aplicacion en el metro de Madrid, se han hecho distintas pruebas con el pn de comprobar que el sistema funciona con un error dentro de unos margenes aceptables. Siempre teniendo en cuenta que esta aplicacion es para distinguir entre estaciones y, si fuese posible, entre andenes.

5.2.1. Pruebas de posición en el laboratorio

Todas las pruebas llevadas a cabo para el desarrollo se han realizado en el edificio B de la Escuela Politecnica Superior, concretamente en el laboratorio B-403 y las habitaciones y zonas colindantes.

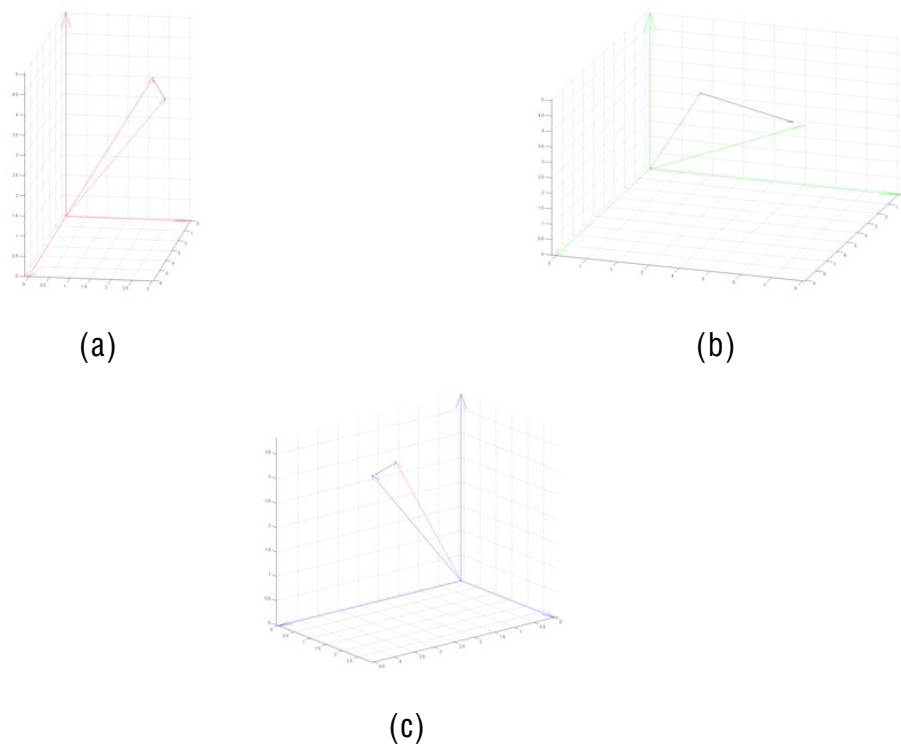


Figura 5.3: Distancia de la potencia medida a una huella de cada habitacion. Azul, rojo y verde las habitaciones, en magenta el vector medido en la prueba y en negro el vector distancia.

5.2.1.1. Mapas de muestras realizados

El mapa realizado en la Escuela Politecnica Superior cuenta con tres zonas: los servicios mas cercanos al laboratorio, el laboratorio B-402 y la zona de ascensores. Cada huella de señal esta compuesta por 12 muestras que se tomaban un intervalo entre muestra y muestra de cinco minutos con el movil en posicion horizontal en una superficie plana, como pod a ser una mesa, balda de estanteria o en el suelo como ha tenido que ser en la zona de los ascensores.

Las habitaciones presentes en el mapa tienen una distribución distinta de muebles y materiales de construcción. Distan unos diez metros entre sí siendo el más alejado la zona de los servicios y, como se puede ver más adelante, es en el que menos errores de localización hemos obtenido.

5.2.1.2. Resultados de localización

Todas las pruebas se han realizado con el movil en la mano y en varias posiciones dentro de cada zona.

Los resultados obtenidos se ven en la tabla 5.1. El numero de antenas detectadas durante las pruebas ha oscilado entre cinco y nueve, no dependiendo ni de la zona ni del momento (en el mismo sitio con medidas consecutivas el numero de antenas var a) mientras que el numero de antenas detectadas en el mapa de muestras llegaba hasta las 15. Es un resultado dentro de los margenes para nuestro proyecto, teniendo

	Aciertos	Fallos	Porcentaje acierto
Servicios	119	11	91.54 %
Ascensores	95	29	76.61 %
Laboratorio	91	19	82.73 %
Total	305	59	83.63 %

Tabla 5.1: Tabla con los resultados de los experimentos en el laboratorio. El punto mas lejano, los ascensores es el que menor error presenta.

en cuenta que lo que buscamos en un principio es una diferenciacion entre estaciones. La distancia entre las dos zonas mas alejadas es de entre 40 y 50 metros, por lo que tenemos una granularidad de entre 20 y 25 metros con una tasa de acierto por encima del 80 %. Siempre teniendo en cuenta que el objetivo de este proyecto es la localizacion las l neas de metro y este experimento sigue el esquema de una l nea. Si tenemos en cuenta cuando nos posicionamos en la habitacion mas alejada (una granularidad de 40 metros), la tasa de acierto sube por encima del 90 % y, si observamos solo los errores entre los servicios y la zona de ascensores, la tasa de acierto sube hasta el 99 % como podemos observar en la tabla 5.2.

	Aciertos	Fallos	Porcentaje acierto
Servicios	115	2	98.29 %
Ascensores	66	0	100 %
Total	181	2	99.15 %

Tabla 5.2: Indice de acierto si tenemos en cuenta unicamente los servicios y los ascensores.

5.2.2. Pruebas de posición del móvil

Para comprobar si la posición del móvil afecta a la intensidad de señal se han hecho tres pruebas en el modo granularidad del programa. Este modo (vease el apartado 4.6) en lugar de guardarse la media de las señales, se guarda cada muestra individual tomada cada segundo para poder realizar un estudio de la potencia.

Las tres posiciones han sido perpendiculares entre sí, tumbado y de canto como se muestra en la figura 5.4.

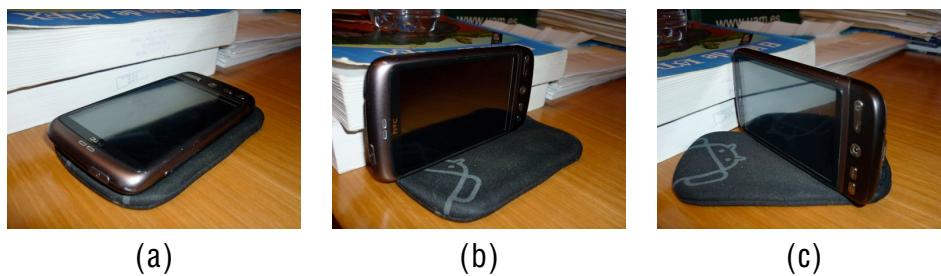


Figura 5.4: Tres posiciones perpendiculares entre sí en las que se ha medido la variación de potencia.

Las medidas han variado significativamente a la hora de cambiar de posición tanto aumentando como disminuyendo su intensidad. La variación máxima ha sido de 34 dBm y la media 24 dBm. La variación de las señales se puede ver en la gráfica 5.5 en la cual vemos que mientras el móvil está estático la señal se mantiene completamente estable.

5.2.3. Pérdida de intensidad sujetar el dispositivo

En esta prueba hemos medido la diferencia de potencia entre mantenerlo en la mano durante cinco minutos y cinco minutos encima de la mesa.

Las potencias no han oscilado de manera significativa ni cuando se tenía en la mano ni cuando se sujetaba en la mano, pero todas las señales que se han detectado en ambos casos han variado al soltar o coger el móvil con una variación entre 2 dBm y 20 dBm como se puede observar en la figura 5.6. La media de variación de potencia ha sido de 14,3 dBm.

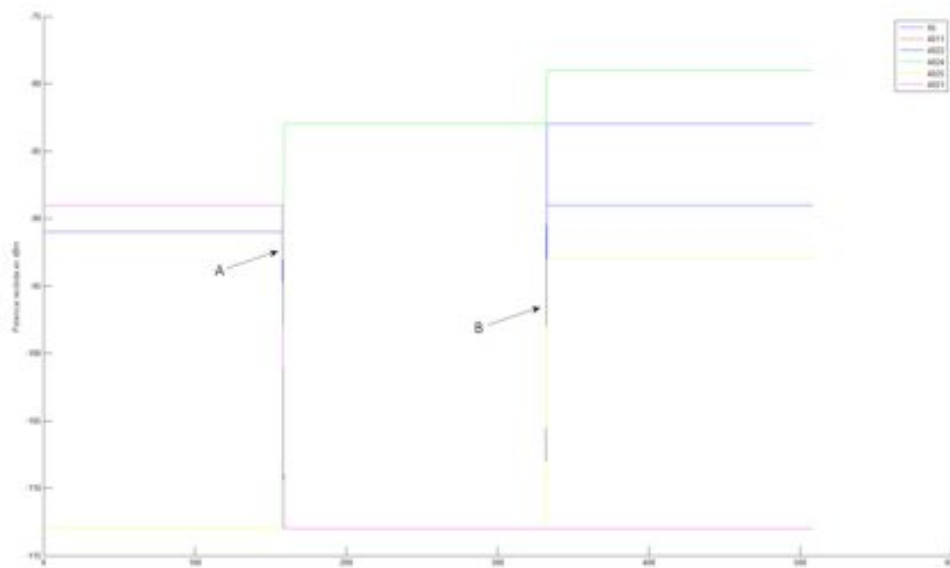


Figura 5.5: Variación de potencia de señal al cambiar de posición el dispositivo móvil. Evento A: cambio de tumbado a canto. Evento B: segundo cambio de posición.

5.3. Medidas y pruebas realizadas en el metro

En esta sección se analizan los mapas de muestras y pruebas realizados en el metro. Es importante tener en cuenta la infraestructura móvil expuesta en el capítulo 3, ya que los resultados obtenidos están directamente relacionados con esta.

5.3.1. Mapas realizados

Para realizar los mapas se han tomado medidas en todas las estaciones de la línea 10 desde Plaza de Castilla hasta Lago, ambas inclusive, y en el andén del ramal que une Príncipe Pío con Opera.

Se ha tomado la estación como zona en lugar de los andenes por dos motivos: los andenes son alargados y la distancia que hay entre dos extremos del andén es mucho mayor que la que hay entre los andenes, además las pruebas hechas en el laboratorio prácticamente descartan la posibilidad de diferenciar con un error aceptable entre dos puntos a menos de diez metros.

El método seguido para realizar el mapa de muestras ha sido distinto al seguido en

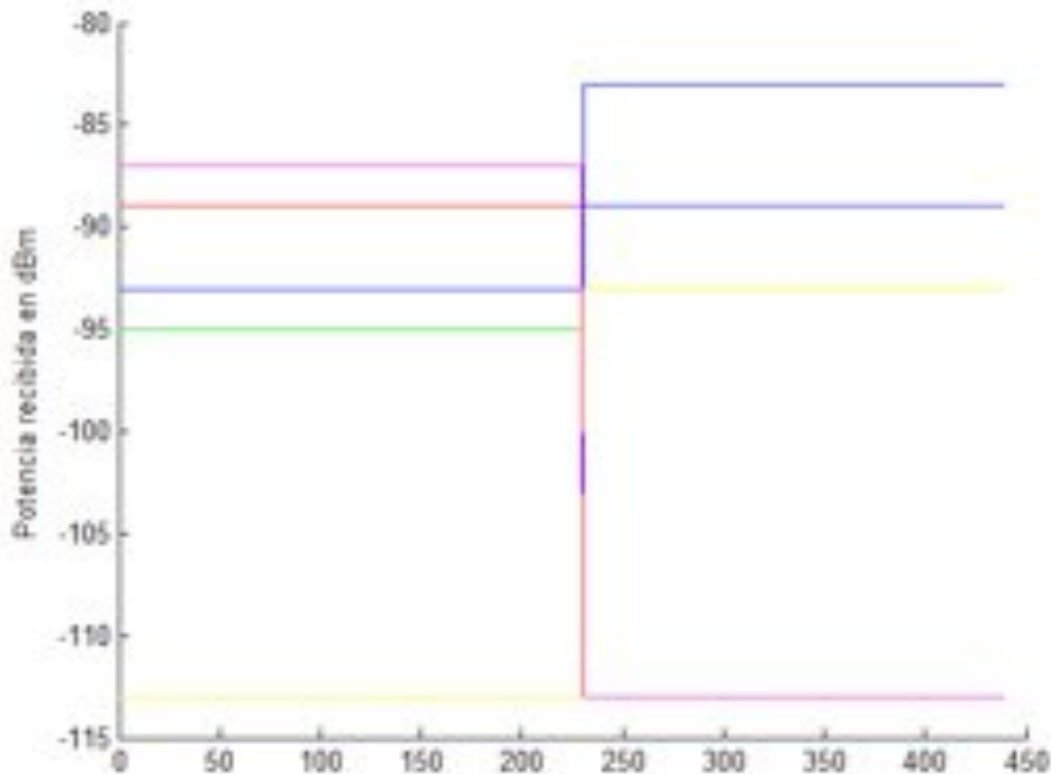


Figura 5.6: Variación de potencia al depositar el móvil en la mesa. Cada color representa una señal distinta.

el laboratorio, dado que hay que tener en cuenta mas factores:

- Con el telefono en la mano y quieto en un punto se toman las huellas de señal a lo largo de un minuto, con una muestra por segundo y, por tanto, un maximo de unas 60 muestras por huella y antena. El hecho de tener el telefono en la mano podr a parecer un inconveniente pero debemos pensar que es como estara casi siempre cuando lo emplee el usuario.
- Se han tomado diez huellas de señal por estacion. Cinco a cada lado, intentando maximizar la distancia entre cada una para cubrir casi toda la estacion, como se puede observar en la pgura 5.7 donde cada punto rojo es una huella de señal.
- El trapco de viajeros mientras se tomaban las huellas siempre ha sido bajo. Das laborables o festivos en horas no puntas (generalmente entre las 16h y las 20h). No todas las zonas fueron muestreadas el mismo d a, pero s en la misma franja horaria.

5.3. MEDIDAS Y PRUEBAS REALIZADAS EN EL METRO

- Se ha procurado que nunca hubiese un tren en la estación. Todas las huellas han tomadas sido sin trenes.

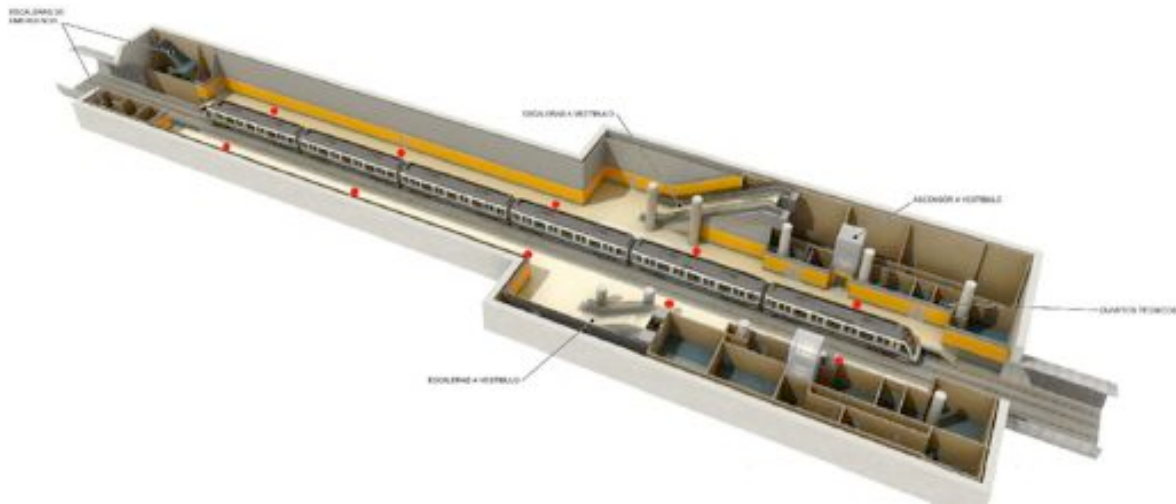


Figura 5.7: Esquema de donde se han tomado las huellas de señal en las estaciones de metro.

Hemos tomado las huellas de señal de solo un minuto por sentido practico, para reducir el tiempo de muestreo. La frecuencia de trenes durante casi todo el día es de un tren cada dos o tres minutos, con cinco minutos por huella ser a imposible tomarlas sin trenes. Además si queremos incluir en los mapas un número grande de estaciones, debemos reducir el tiempo por huella. De este modo hemos tardado en realizar el mapa (once zonas en diez estaciones) algo más de tres horas, debido también a los cambios de estación y de andén y a las interrupciones y esperas para evitar que hubiese trenes en la estación.

Gracias a este mapa y confirmado por los resultados del punto siguiente podemos observar que los *cellid* están compartidos por grupos de estaciones consecutivas. Si bien el primer grupo está muy claro, los siguientes se han establecido en parte gracias a los resultados de las pruebas:

- La primera zona será la que agrupa Plaza de Castilla con Cuzco y Santiago Bernabeu como se puede observar en la tabla 5.3. Se puede observar claramente que todos comparten el *cellid* 263. El 261 de Cuzco es meramente anecdótico, ya que aparece una sola vez. Mientras que el 1081 de Santiago Bernabeu que aparece en prácticamente todas las huellas de señal, este se recibe tan claramente debido seguramente a que el siguiente grupo cubre el túnel entre Santiago Bernabeu y Nuevos Ministerios resultando en una clara recepción en una estación que no pertenece al grupo.

- El segundo grupo no está tan claramente definido. Tanto porque en Nuevos Ministerios hay presentes señales propias del primer grupo, como porque Gregorio Marañón las tiene del grupo siguiente. Finalmente se ha decidido esta clasificación en parte por los resultados de las pruebas de localización. Podemos ver el grupo en la tabla 5.4.
- El tercer grupo está formado por las estaciones de Alonso Martínez y Tribunal si bien para Tribunal hay esta la señal de la antena 916 que la caracteriza, en las pruebas han habido numerosos errores entre estas dos estaciones.
- La estación de Plaza de España representa un grupo por sí misma. Contiene subcélulas *cellid* que no se reciben de manera relevante en ninguna otra estación. Se puede observar en la tabla 5.6 que la única antena que no es exclusiva de esta estación, a pesar de ser recibida en toda la estación tiene pocas muestras y con la señal media más débil de todas.
- Por último y aparte son las estaciones de Lago y Príncipe Pío. Estas estaciones reciben señales del exterior y se llegan a recibir en Príncipe Pío hasta 16 señales y en Lago hasta 20, luego son estaciones con muy buena recepción y en principio no deberían dar ningún problema de localización.

	263		1081		261	
	Número	Rssi	Número	Rssi	Número	Rssi
Plaza de Castilla	936/10	-62.2	0	-	0	-
Cuzco	865/10	-96	0	-	1/1	-113
Santiago Bernabeu	974/10	-59.8	735/9	-94.6	0	-

Tabla 5.3: Primer grupo de estaciones en las que se indica por cada célula el número de veces registrado con el número de huellas de señal en las que aparece, además de la potencia media (rssi) en dBm.

	1082		1081		263	
	Número	Rssi	Número	Rssi	Número	Rssi
Nuevos Ministerios	853/9	-57	1037/10	-60	865/10	-96
Gregorio Marañón	789/9	-70.1	1/1	-105	0	-
	762		395		701	
	Número	Rssi	Número	Rssi	Número	Rssi
Nuevos Ministerios	59/1	-99	81/4	-99	0	-
Gregorio Marañón	0	-	0	-	569/8	-71.05

Tabla 5.4: Grupo de nuevos ministerios de nuevo, se muestra el número de veces que aparece la célula por huellas y la potencia media.

5.3. MEDIDAS Y PRUEBAS REALIZADAS EN EL METRO

	701		333		702	
	Número	Rssi	Número	Rssi	Número	Rssi
Alonso Martinez	1316/9	-100.77	1277/10	-96.8	11/2	-74
Tribunal	862/9	-66.56	1011/10	-66	0	-
	1321		331			
	Número	Rssi	Número	Rssi		
Alonso Martinez	0	-	1/1	-113		
Tribunal	916/10	-82.8	0	-		

Tabla 5.5: Tercer grupo formado por las estaciones de Alonso Martinez y Tribunal con numero de muestras por huellas y su potencia media.

	333		1321		1322		1323	
	Número	Rssi	Número	Rssi	Número	Rssi	Número	Rssi
España	372/10	-102	999/10	-58.2	286/8	-101.25	803/9	-76.33

Tabla 5.6: Cellid con su frecuencia por numero total de huellas y potencia para la estacion de Plaza de España.

5.3.2. Pruebas de localización

Las pruebas se han realizado indistintamente dentro y fuera del tren con el movil en la mano, con aproximadamente la misma tasa de error tanto en anden como en el vagon.

Los resultados han sido bastante negativos. En las tablas se puede observar los aciertos y con cuales estaciones se equivoco el programa. En las tablas se presentan en tres grupos mas el cuarto grupo de Principe Pio y Lago:

- En el primer grupo compuesto por Plaza de Castilla, Cuzco y Santiago Bernabeu se han detectado los *cellid* 263 con una potencia muy parecida en todos los casos: entre -51 dBm y -70 dBm. Además ha inducido a error que en Plaza de Castilla se detectara en un caso la señal con *cellid* 261 con una potencia de -105 dBm, que es una señal que en los mapas solo se detecto en Cuzco sin practicamente potencia. Esta homogeneidad entre las tres estaciones junto con que solo hay una unica señal hace practicamente imposible discernir entre las tres estaciones unicamente con este algoritmo, se puede observar la aleatoriedad de los resultados en la tabla 5.7 con 1/3 de los aciertos totales.

Pongamos un ejemplo tanto con un fallo como con un acierto:

- Acierto: en Plaza de Castilla se recibe solamente la antena 263 con una potencia de -77 dBm. En el mapa, Santiago Bernabeu (vease tabla 5.3) tiene

en 9 de 10 huellas de señal 2 antenas con lo que su subespacio vectorial es de dos dimensiones y, como una antena no se detecta, en una de las componentes se suma la distancia máxima dando una distancia superior a 133. La media de distancias calculadas de Plaza de Castilla y Cuzco son de 20.8 y 33.5, ligeramente superior la de Cuzco debido a que una huella de señal tiene un subespacio de dos dimensiones. Obtenemos por tanto, que la posición es Plaza de Castilla.

- Error: en Plaza de Castilla se reciben dos señales de las antenas 263 y 261 con unas potencias de -55 y -105 dBm respectivamente. Plaza de Castilla tiene un subespacio de una dimensión, con lo que su distancia media queda 134.6, Cuzco en una de sus huellas sus componentes coinciden bajando la media hasta 127.6 y Santiago Bernabeu tiene una distancia media ligeramente superior con 136.7. La posición elegida será Cuzco cuando estamos en Plaza de Castilla.
- Como se ha mencionado en el apartado anterior, en el segundo grupo formado por Nuevos Ministerios y Gregorio Marañón, la estación de Nuevos Ministerios recibe las señales de la antena con *cellid* 263. Esto se puede ver claramente en los errores, ya que parte de los producidos en Nuevos Ministerios han sido por la estación de Santiago Bernabeu. En este caso en la estación de Nuevos Ministerios se han recibido señales de las antenas 1081, 1082 y 263 mientras que en Gregorio Marañón tan solo de la 1082. Los errores están basados básicamente de nuevo en el número e identidad de los *cellid*. Nuevos Ministerios tiene un gran número de antenas en el mapa que se detectan a la hora de encontrar la posición pero rara vez todas a la vez, lo que penaliza bastante la distancia. Esto induce enseguida a error con las otras dos colindantes que tienen un número menor de antenas y siempre darán una distancia menor. Los resultados están en la tabla 5.8.
- En la tabla 5.9 están presentes dos grupos, tanto el de la estación de Plaza de España, como el que acoge a Tribunal y Alonso Martínez. Este segundo grupo presenta un comportamiento bastante mejor que las dos primeras, en parte gracias a la antena 1321 que está presente en Tribunal y no Alonso Martínez. Plaza de España, como grupo aparte no presenta ningún inconveniente.
- Tema aparte son los casos de Lago y Príncipe Pío, que con antenas distintas y gran número de ellas presentan una tasa de acierto del 100%.

Como se puede observar, básicamente la localización depende únicamente de las antenas detectadas. La potencia recibida es muy homogénea entre estaciones y no hay las subcientas señales como para hacer un mapa que permita distinguir con precisión a través del método propuesto. Esto da una tasa de error bastante alta, si bien no hay errores al distinguir entre grupos de estaciones a la hora de determinar la estación concreta la tasa de acierto supera ligeramente el 60% y esto teniendo en cuenta las estaciones de Plaza de España, Príncipe Pío y Lago.

5.3. MEDIDAS Y PRUEBAS REALIZADAS EN EL METRO

	Plaza de Castilla	Cuzco	Santiago Bernabeu	Total aciertos
Plaza de Castilla	7	3	0	7/10
Cuzco	2	2	1	2/5
Santiago Bernabeu	4	1	0	0/5

Tabla 5.7: Resultados del primer grupo. Presenta un acierto del 36%, es decir casi completamente aleatorio entre las tres estaciones.

	Santiago Bernabeu	Nuevos Ministerios	Gregorio Marañón	Total aciertos
Nuevos Ministerios	2	0	4	0/6
Gregorio Marañón	0	0	6	6/6

Tabla 5.8: Resultados del segundo grupo. Presenta un acierto del 50%, de nuevo aleatorio entre ambas estaciones. Aunque en este caso hay errores con una estación del primer grupo.

	Alonso Martínez	Tribunal	Plaza de España	Total aciertos
Alonso Martínez	7	0	0	7/7
Tribunal	7	6	0	6/13
Plaza de España	0	0	11	11/11

Tabla 5.9: Resultados del tercer y cuarto grupo (formado solo por plaza de España). Un acierto del 65% y el 100% respectivamente.

No obstante si tenemos en cuenta que las zonas en las que se divide la cobertura móvil del metro (vease 3.3), la tasa de acierto es del 100% como se puede ver en la tabla 5.10. Las estaciones se agrupan como ya se ha mencionado anteriormente con 4 zonas más las exteriores:

- Plaza de Castilla: Cuzco, Plaza de Castilla y Santiago Bernabeu.
- Nuevos Ministerios: Nuevos Ministerios y Gregorio Marañón.

- Alonso Martinez: Alonso Martinez y Tribunal.
- Plaza de España: Plaza de España.
- Exteriores: Principe P o y Lago.

	Plaza Castilla	Nuevos Ministerios	Alonso	Plaza España	Exterior
Plaza Castilla	20	0	0	0	0
Nuevos Ministerios	0	12	0	0	0
Alonso	0	0	20	0	0
Plaza España	0	0	0	11	0
Exterior	0	0	0	0	16

Tabla 5.10: Resultados clasificados por zonas. Hay un 100% de aciertos.

En el apartado 5.1 se explican tres métodos para calcular la media de las distancias de todas las huellas de señal dentro de una zona. Cuando se aplican los tres y se eligen dos de tres la tasa de acierto cae hasta el 53%, incluso dando errores en casos como estando en Tribunal detectando las antenas 1321, 701 y 333 se da por posición correcta Alonso Martinez (que no contiene la antena 1321). Por lo tanto todos los resultados aquí mostrados son considerando únicamente como buena la media aritmética.

Estos resultados tan negativos se justifican en parte por la homogeneidad de las señales comunes en las zonas. En la tabla 5.11 podemos observar las potencias medias y las desviaciones de cada muestra dentro de las estaciones para cada una de las antenas comunes que presenta cada estación de cada zona (por ejemplo la zona de Alonso Martinez tiene en común con Tribunal las antenas 701 y 333). Estos datos nos dan que la diferencia entre las potencias recibidas en dos estaciones distintas con la misma estación base no será la suficiente información como para distinguir la posición entre estaciones.

5.3.3. Variación de potencia en estación por entrada de trenes

El fin de esta prueba es determinar la interferencia de los trenes a la recepción de señal. Las antenas en el andén están bien localizadas, así que para realizar la prueba nos hemos posicionado en varias estaciones en el andén contrario a la antena y hemos

5.3. MEDIDAS Y PRUEBAS REALIZADAS EN EL METRO

	701	333
	Media	Media
Alonso Martinez	$-100,8 \pm 24,4$	$-96,8 \pm 3,6$
Tribunal	$-66,6 \pm 5,3$	$-66 \pm 19,5$

	263
	Media
Plaza de Castilla	$-65,2 \pm 5,5$
Cuzco	$-58,8 \pm 4,52$
Santiago Bernabeu	$-59,8 \pm 3,2$

	1082
	Media
Nuevos Ministerios	-87 ± 4
Gregorio Marañón	$-105 \pm 24,6$

Tabla 5.11: Las potencias medias de las antenas comunes de cada zona con sus respectivas desviaciones estandar de cada una.

medido con el movil en la mano mientras entraban y sal an trenes (con sus respectivos pasajeros).

Se han podido dar todas las situaciones posibles: llegada y salida de un unico tren (en ambos sentidos) y ambos trenes en la estacion. La grafca que lo representa es la 5.8.

En la esta prueba se ha podido hacer patente que los trenes inducen una variacion en la potencia recibida, pero tambien deja patente que esta no es mas signipcativa que otras interferencias como puede ser el paso de gente o, como se ha medido en las pruebas del laboratorio el hecho de sujetar el movil con la mano o la posicion de este. De hecho la variacion maxima a la que esta expuesta el movil no dipere mucho de las variaciones inducidas por sujetar el movil o su posicion, siendo esta variacion maxima ligeramente superior a los 20 dBm.

Se puede observar, ademas que hay otras variaciones igualmente signipcativas que no parecen tener correlacion con la entrada y salida de trenes, si bien siempre que entra o sale uno la potencia var a.

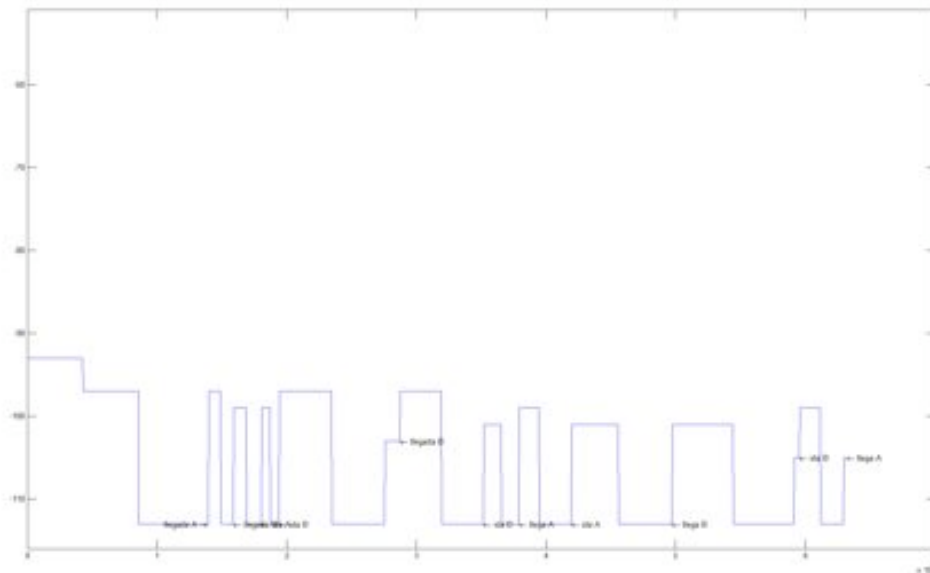


Figura 5.8: Potencia en recibida en dBm en funcion del tiempo donde se puede observar el efecto de los trenes en la señal. Tren A es el tren en el mismo anden en el que estamos, tren B el tren que circula en el sentido contrario

Conclusiones y trabajo futuro

6.1. Conclusiones

La localización en interiores es un problema no resuelto. Aunque hay soluciones aproximadas muy accesibles con los teléfonos inteligentes, no son todo lo precisas que muchas veces se requiere. Para aumentar esta precisión hace falta o ampliar la infraestructura o, utilizando la ya existente, realizar estudios y mapas de señal que luego se requieran para la localización.

Con este proyecto hemos abordado el problema con esta segunda solución, un mapa de potencias. Para ello hemos desarrollado dos programas: uno de toma de muestras de potencia y el procesamiento de estas para realizar los mapas de potencia y un segundo programa que a partir del mapa realizado da una localización simbólica.

En primer lugar se han desarrollado los programas en el edificio de la Escuela Politécnica de la UAM, donde se han realizado los primeros experimentos y estudios. Durante estos experimentos realizados en tres habitaciones próximas entre sí hemos obtenido una precisión superior al 80%, demostrando la viabilidad del proyecto bajo las siguientes condiciones:

- Las condiciones de las muestras de los mapas no tienen que ser muy estrictas: no todas las muestras están a la misma distancia ni han sido tomadas de manera totalmente uniforme en las habitaciones. El móvil siempre en la misma posición pero a distintas alturas y sobre distintos materiales (mesas, azulejos, baldas de metal o suelo).
- La densidad de las muestras del mapa es variable, ya que todas las habitaciones tienen el mismo número de muestras pero no el mismo tamaño.
- La posterior localización se realiza con el móvil en la mano, esto implica que

cambia las condiciones de recepción como se ha observado en los experimentos expuestos en la sección 5.2.3.

Una vez obtenidos estos resultados procedimos a estudiar la viabilidad dentro del metro de Madrid. Se realizaron los mapas de las estaciones de la línea 10 comprendidas entre Plaza de Castilla y Lago. Las pruebas de localización posteriormente realizadas daban una indeterminación muy grande en grupos de estaciones. Esto es debido a las características de la infraestructura de la telefonía móvil dentro del metro parecen optimizar el empleo de esta de tal manera que las compañías comparten infraestructura. Eso repercute en que el número de antenas y señales recibidas en el suburbano son mínimas, recibiendo en muchos casos una única señal, mientras que en el laboratorio rara vez recibimos señal de menos de diez antenas.

La precisión del algoritmo depende del número de señales recibidas. Se han visto diversos algoritmos (véase el capítulo 2), como teniendo en cuenta solo seis señales se obtiene una precisión mucho menor que si se tienen en cuenta todas las recibidas por débiles que fuesen. Con una recepción de señal de 4 antenas en el mejor de los casos y como norma una o dos como mucho, este algoritmo no es capaz de precisar la localización mucho más allá de la simple localización por *cellid*.

Así, aunque hemos comprobado que es posible la localización con una granularidad de entre 10 y 20 metros en el interior de edificios, esto dentro del metro de Madrid no es posible solo con este algoritmo ya que ni siquiera logramos localizar y distinguir entre todas las estaciones. Si bien es fácil la localización en las agrupaciones por estaciones base en las que está dividido el metro.

6.2. Trabajo futuro

Aunque no se haya logrado el objetivo inicial de distinguir entre estaciones en el interior del metro a nivel de potencia de señal, a partir de esta información se puede desarrollar un sistema que permita a más alto nivel afinar el resultado. Con información extra como conocer la estación inicial o las estaciones previas se puede implementar un algoritmo que descarte estaciones erróneas. También una buena mejora sería agrupar las estaciones por zonas en función de sus estaciones base y sus relaciones y solapamientos. Por último y a medida que se amplíe la cobertura Wi-Fi en la red de metro, se podrá utilizar esta para mejorar la precisión.

En el nivel del programa en Android, no se han llegado a implementar varias ideas debido a la baja precisión. Como presentar la localización en un mapa de metro o convertir la actividad de localización en un servicio y que este trabaje por detrás, mandando un mensaje vía *Toast* (un *Widget* de Android que manda un mensaje corto

6.2. *TRABAJO FUTURO*

al usuario) cada vez que haya un cambio de estacion o, en su lugar, a otra aplicacion que lo necesitase como es WAlroutes.

Bibliografía

- [1] *Ad hoc positioning system (APS) using AOA*, volume 3, April 2003.
- [2] *An UWB relative location system*, 2003.
- [3] 3gpp ts 27.007 v6.3.0 3rd generation partnership project; technical specification group terminals; at command set for user equipment (ue) (release 6). Technical report, Global System for Mobile communications, 2003-06.
- [4] *The limits of localization using signal strength: a comparative study*, 2004.
- [5] *WLAN location determination in e-home via support vector classification*, volume 2, 2004.
- [6] Hp confirms discussions with autonomy corporation plc regarding possible business combination; makes other announcements. <http://www.businesswire.com/news/home/20110818006301/en/HP-Confirms-Discussions-Autonomy-Corporation-plc-Business>, 2011.
- [7] Nokia and microsoft form partnership. <http://www.bbc.co.uk/news/business-12427680>, 2011.
- [8] C. P. y. J. R. nez Fernando Seco C. P. y. J. R. nez Fernando Seco Carlos Prieto y Javier Roa Antonio R. Jimenez, Fernando Seco. Tecnologías sensoriales de localización para entornos inteligentes. *Simposium UCAM I 05 Computación Ubicua e Inteligencia Ambiental*, 1:75{86, 2005.
- [9] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2:775{784, 2000.
- [10] Richard Hartnett Patrick M. Thompson Benjamin B. Peterson, Chris Kmiecik and Jose Mendom. Spread spectrum indoor geolocation. *J. Inst. Navigat*, 45:97{102, 1998.

- [11] Stefan Carmien, Melissa Dawe, Gerhard Fischer, Andrew Gorman, Anja Kintsch, and James F. Sullivan. Socio-technical environments supporting people with cognitive disabilities using public transportation. *Transactions on Human-Computer Interaction (ToCHI)*, 12:233{262, 2005.
- [12] Inc. Cisco Systems. Location tracking approaches. *cisco.com*, 2:1{20, 2008.
- [13] Andrew Corporation. Comparison of propagation losses in tunnels with antennas and radiax. *Distributed Communications Systems, 2601 Telecom*, June 29, 2001.
- [14] David Matellano Criado. Interfaz de control de entornos de inteligencia ambiental a traves de iphone. Master's thesis, Universidad Autonoma de Madrid, 2010.
- [15] Jerome (J.F.) DiMarzio. *Android. A programmer's Guide*. Mc Graw Hill, 2008.
- [16] Christopher Drane, Malcolm Macnaughtan, and Craig Scott. Positioning gsm telephones. *IEEE Communications Magazine*, 36:46{54, 1998.
- [17] P. Saco et al. La telefon a celular en la era blackberry. *Negotium*, 5(13):71{79, 2009.
- [18] B. T. Fang. Simple solutions for hyperbolic and related position boxes. *Aerospace and Electronic Systems, IEEE Transactions on*, 26(5):748{753, 1990.
- [19] Andre Gunther and Christian Hoene. Measuring round trip times to determine the distance between wlan nodes. In *in Proc. of Networking 2005*, pages 768{779. Springer-Verlag, 2005.
- [20] GSMA. Asia pacific mobile observatory 2011 driving economic and social development through mobile broadband. 2011.
- [21] Beverly Harrison and Anind Dey. What have you done with location-based services lately? *IEEE Pervasive Computing*, 8(4):66{70, 2009.
- [22] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. *Wirel. Netw.*, 8:187{197, March 2002.
- [23] JePrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57{66, August 2001. This article is also excerpted in "IT Roadmap to a Geospatial Future," a 2003 report from the Computer Science and Telecommunications Board of the National Research Council.
- [24] Lichtenegger H. Collins J. Hofmann-WellenhoP, B. *Global Positions System: Theory and Practice*. Springer Verlag, 1997.
- [25] JePrey. Spoton: An indoor 3d location sensing technology based on rf signal strength.

- [26] M. Kanaan and K. Pahlavan. A comparison of wireless geolocation algorithms in the indoor environment. In *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, volume 1, pages 177{182, 2004.
- [27] Eiman Kanjo, Jean Bacon, David Roberts, and Peter LandshoP. Mobsens: Making smart phones smarter. *IEEE Pervasive Computing*, 8(4):50{57, 2009.
- [28] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-camera multi-person tracking for easyliving. pages 3{10, 2000.
- [29] Kari Laasonen, Mika Raento, and Hannu Toivonen. Adaptive On-Device Location Recognition. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, chapter 21, pages 287{304. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2004.
- [30] H. Laitinen, J. Lahteenmaki, and T. Nordstrom. Database correlation method for GSM location. In *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)*, volume 4, pages 2504{2508. IEEE, 2001.
- [31] Anthony Lamarca, Yatin Chawathe, Sunny Consolvo, JePrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, JeP Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. In *Proceedings of the Third International Conference on Pervasive Computing*, May 2005.
- [32] Xinrong Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila. Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems. In *Vehicular Technology Conference, 2000. IEEE VTS Fall VTC 2000. 52nd*, volume 6, pages 3015{3020 vol.6, 2000.
- [33] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067{1080, November 2007.
- [34] Pablo A. Haya Lucia Angles-Alcazar and Rosa Carro. Wai-routes: A route-estimator system for aiming public transportation users with cognitive impairments. *Proceedings of the Sixth IASTED International Conference Advances in Computer Science and Engineering*, pages 167{174, 2010.
- [35] Pilar Alvarez. Conversaciones bajo tierra. *El País*, 05/06/2011.
- [36] Metro Madrid. Estad stica de transporte de viajeros. <http://www.ine.es/daco/daco42/daco4210/tv1209.pdf>, 2010.
- [37] Reto Meier. *Proffesional Android 2 Application Development*. Wrox, 2010.
- [38] Sean Morrissey. *History of Apple Mobile Devices*. Apress, 2010.

- [39] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. Landmarc: indoor location sensing using active rpd. *Wirel. Netw.*, 10:701{710, November 2004.
- [40] nVidia. The benefits of multiple cpu cores in mobile devices. *Whitepaper*, 2010.
- [41] Robert J. Orr and Gregory D. Abowd. The smart door: A mechanism for natural user identification and tracking, 2000.
- [42] R.R. Oskui. Especificacion formal en coq del modulo de control de acceso de midp 2.0 para dispositivos moviles interactivos.
- [43] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate gsm indoor localization. *UbiComp 2005: Ubiquitous Computing*, 1:141{158, 2005.
- [44] K. Pahlavan, Xinrong Li, and J. P. Makela. Indoor geolocation science and technology. *IEEE Communications Magazine*, 40(2):112{118, February 2002.
- [45] Beatriz Alonso Pedro Sanchez. La red de metro dispondra de cobertura total para los telefonos moviles en junio. *ABC*, 07/10/2007.
- [46] Sichitiu M. L. Peng, R. Angle of arrival localization for wireless sensor networks. in sensor and ad hoc communications and networks. *SECON*, 1:374{382, 2006.
- [47] Eva María García Polo. Tecnicas de localizacion en redes inalambricas de sensores. *Redes inalambricas de sensores: arquitecturas y aplicaciones*, 1:77{98, 2008.
- [48] Nissanka Bodhi Priyantha, Hari Balakrishnan, and Bodhi Priyantha. The cricket indoor location system. Technical report, MIT, 2005.
- [49] Blood E. B. Steiner T. O. Jones H. R. Raab, F. H. Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronic Systems*, 15(5):709{717, 1979.
- [50] Zigurd Mednieks & Blake Meike Rick Rogers, John Lombardo. *Android Application Development*. O'Reilly, 2009.
- [51] Sajid Siddiqi, Gaurav S. Sukhatme, and Andrew Howard. Experiments in monte-carlo localization using wip signal strength. In *Proceedings of the International Conference on Advanced Robotics*, Coimbra, Portugal, July 2003.
- [52] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y. Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, JePrey Hightower, William G. Griswold, and Eyal de Lara. Mobility detection using everyday gsm traces. In *Proceedings of the Eighth International Conference on Ubiquitous Computing (UbiComp 2006)*, Lecture Notes in Computer Science, pages 212{224. Springer-Verlag, September 2006.
- [53] D. J. Torrieri. Statistical theory of passive location systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-20(2):183{198, February 2007.

- [54] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [55] Alex Varshavsky, Eyal de Lara, JePrey Hightower, Anthony LaMarca, and Veljo Otsason. Gsm indoor localization. *Pervasive and Mobile Computing*, 3(6):698{720, 2007.
- [56] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91{102, January 1992.
- [57] Mark Weiser. The Computer for the 21st Century. *Scientific American*, (Communications, Computers, and Network), September 1991.
- [58] J. Werb and C. Lanzl. Designing a positioning system for pnding things and people indoors. *IEEE Spectrum*, 35(9):71{78, September 1998.

Vocabulario técnico, siglas y lenguaje técnico

1) Estado del arte

- Global Positioning System (GPS): Sistema de Posicionamiento Global.

2) Estado del arte

- Closest-neighbor (CN): vecino más cercano
- Residual weighting (RWGH): peso residual[26]
- Angle of arrival estimation: ángulo de llegada.
- Received signal strenght (RSS): potencia de señal recibida.
- Round times of flight (RTOF): tiempos de vuelo.
- Fingerprints: huellas de señal.
- AHLos: Ad-Hoc Localization System.
- Software development kit (SDK): paquetes de desarrollo software.
- Java Micro Edition virtual machine (MicroJvm): maquina virtual Java edición micro.
- Cellid: identificador de celda.
- Base transceiver station (BTS): estación base.
- Location area code (LAC): código de localización de área.
- Mobile Country Code (MCC): código de país para móviles.
- Mobile Network Code (MNC): código de red de móviles.
- Received signal strength indication (RSSI): indicador de potencia recibida.
- Radio-frequency identification (RFID): identificador por radiofrecuencia.

3) Infraestructura del Metro de Madrid

- Base Station Subsystem (BTS): Estación base.
- Network and Switching Subsystem: Subsistema de red.

APÉNDICE A. VOCABULARIO TÉCNICO, SIGLAS Y LENGUAJE TÉCNICO

Madrid, Abril 2012

El Ingeniero Jefe de Proyecto

Fdo.: Miguel Ruiz Cuesta

Ingeniero Superior de Telecomunicación

Apéndice B

Presupuesto

1) Ejecución Material	
▪ Compra de ordenador personal (Software incluido)	800 €
▪ Compra de dispositivo móvil	600 €
▪ Material de oficina	150 €
▪ Total de ejecución material	1550 €
2) Gastos generales	
▪ sobre Ejecución Material	352 €
3) Beneficio Industrial	
▪ 6 % sobre Ejecución Material	93 €
4) Honorarios Proyecto	
▪ 640 horas a 15 €/ hora	9600 €
5) Material fungible	
▪ Gastos de impresión	60 €
▪ Encuadernación	200 €
6) Subtotal del presupuesto	
▪ Subtotal Presupuesto	11855 €
7) I.V.A. aplicable	
▪ 16 % Subtotal Presupuesto	1896.8 €
8) Total presupuesto	
▪ Total Presupuesto	13751.8 €

Madrid, Abril 2012

El Ingeniero Jefe de Proyecto

Fdo.: Miguel Ruiz Cuesta

Ingeniero Superior de Telecomunicación

Pliego de condiciones

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un *posicionamiento en interiores basado en dispositivos móviles*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales.

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios

facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrataz anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares.

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

-
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.