

Escola Universitària Politécnica de Mataró

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA**

**INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN, ESPECIALIDAD EN
TELEMÁTICA**

SISTEMA DE GEOLOCALIZACIÓN DE PERSONAS CON DEMENCIA

Memoria

**SERGIO GARCIA ANDREU
PONENT: PERE BARBERÁN**

PRIMAVERA 2011



**TecnoCampus
Mataró-Maresme**

Dedicatoria

Dedicado a todas aquellas personas que me han apoyado en el transcurso de los años y han confiado en mí en todo momento.

Resumen

El objetivo principal de este proyecto es diseñar, implementar y evaluar un sistema que será capaz de dar un servicio de localización mediante un teléfono móvil utilizando la señal del GPS, para aquellas personas que tienen riesgo de perderse.

El proyecto está dividido en tres partes: la primera es la implementación de una aplicación para un teléfono que debe ser capaz de obtener la señal GPS, tratarla y enviar las coordenadas a un servidor mediante el protocolo XMPP. La segunda parte se basa en la recepción del mensaje y su almacenaje en una base de datos. La tercera es la implementación de una web para visualizar sobre un mapa la posición del móvil.

Resum

L'objectiu principal d'aquest projecte és dissenyar, implementar i avaluar un sistema que serà capaç donar un servei de localització mitjançant un telèfon mòbil utilitzant el senyal GPS, per a aquelles persones que tenen risc de perdre's.

El projecte està dividit en tres parts: la primera és la implementació d'una aplicació per a un telèfon que ha de ser capaç d'obtenir la senyal GPS, tractar-la i enviar les coordenades a un servidor mitjançant el protocol XMPP. La segona part es basa en la recepció del missatge i el seu emmagatzematge en una base de dades. La tercera és la implementació d'un web per visualitzar sobre un mapa la posició del mòbil.

Abstract

The main objective of this project is to design, implement and evaluate a system that will be able to give a location service using a mobile phone using GPS signal to those who are at risk of being lost.

The project is divided into three parts: the first is the implementation of an application for a phone to be able to get the GPS signal, treat it and send the coordinates to a server using the XMPP protocol. The second part is based on the reception of the message and its storage in a database. The third is the implementation of a website to display on a map the position of the mobile.

Índice

Índice de figuras.....	V
Índice de tablas.....	VII
Glosario de términos.....	IX
1. Objetivo.....	1
1.1. ¿Cuál es la problemática de las personas con demencia?.....	1
1.2. ¿A qué colectivo va dirigido?.....	2
1.3. ¿Cuál es la motivación para realizar este proyecto?.....	2
1.4. ¿Qué se quiere conseguir?.....	3
1.5. ¿Cómo funciona el sistema?.....	4
1.6. Estado del Arte.....	5
2. Funcionamiento básico del GPS.....	9
2.1. Historia y Cronología.....	9
2.2. Funcionamiento.....	12
2.3. Triangulación.....	12
2.4. Midiendo la distancia.....	12
2.5. Obtención de un Timing Perfecto.....	12
2.6. Posicionamiento de los Satélites.....	13
2.7. Corrección de Errores.....	13
3. Entorno del terminal móvil.....	17
3.1. Sistemas operativos.....	17
3.1.1. Android:.....	18
3.1.2. iOS:.....	19
3.1.3. Windows Phone:.....	20
3.1.4. Blackberry:.....	21
3.1.5. Symbian OS.....	22
3.2. ¿Por qué Android?.....	23
3.3. ¿Qué es Android?.....	23
3.4. Lenguaje programación.....	24
3.5. Entorno de programación.....	25
3.6. Instalación de Eclipse.....	25

3.6.1. Java	25
3.6.2. Eclipse.....	26
3.6.3. SDK de Android.....	26
3.6.4. ADT (Android Development Tools).....	30
3.7. Aplicación Android	32
3.7.1. Modulo GPS.....	34
4. XMPP	39
4.1. ¿Qué es XMPP?	39
4.2. Funcionamiento	39
4.2.1. Funcionamiento del Cliente	40
4.3. Puntos fuertes de XMPP	41
4.4. Puntos débiles de XMPP	42
4.5. Módulo emisor XMPP Android	42
4.6. Modulo Receptor Java	44
5. APACHE	51
5.1. ¿Qué es?.....	51
5.2. Características principales	51
5.3. Funcionalidades	52
5.4. Caso práctico	52
6. Api de Google Maps	55
6.1. ¿Qué es Api de Google Maps?	55
6.2. Utilidades.....	55
6.3. Obtención de API Key.....	56
6.4. Aplicación web	57
6.4.1. Administrador.....	58
6.4.2. Dar de alta	59
6.4.3. Posición	60
6.4.4. Recorrido	61
7. Conclusiones.	63
8. Mejoras futuras.....	65
9. Referencias	67
10. Anexos.....	71

10.1. Presupuesto	71
10.1.1. Coste de personal.....	72
10.1.2. Coste material informático	72
10.1.3. Coste total del proyecto.....	73

Índice de figuras.

Fig. 1.1. Esquema funcional del sistema de geolocalización.	4
Fig. 1.1. Ejemplo del programa latitude.	6
Fig. 1.3. Ejemplo del sistema keruve.	7
Fig. 2.1. Sistema de navegación LORAN.	10
Fig. 2.2. Posicionamiento de los satélites.	13
Fig. 2.3. Imagen de un sistema de posicionamiento diferencial.	14
Fig. 3.1. Comparativa entre sistemas operativos.	17
Fig. 3.2. Logotipo de android.	18
Fig. 3.3. Logotipo de iOS.	19
Fig. 3.4. Logotipo de Windows Phone.	20
Fig. 3.5. Logotipo de Blackberry.	21
Fig. 3.6. Logotipo de Symbian OS.	22
Fig. 3.7. Resultado de la verificación de java –versión.	26
Fig. 3.8. Ventana de propiedades del sistema.	27
Fig. 3.9. Variables de entorno.	28
Fig. 3.10. Ventana de modificación de las variables de entorno.	28
Fig. 3.11. Android SDK and AVD Manager.	29
Fig. 3.12. Ventana de instalación de los componentes del SDK.	29
Fig. 3.13. Instalación del nuevo software en el Eclipse.	30
Fig. 3.14. Añadir repositorio en el Eclipse.	30
Fig. 3.15. Ventana de instalación de nuevo Software.	31
Fig. 3.16. Esquema de funcionamiento del sistema de Geolocalización.	32
Fig. 3.17. Imagen de la aplicación instalada en un terminal HTC.	33
Fig. 3.18. Imagen de la aplicación ejecutada en un terminal HTC.	34
Fig. 3.19. Ejemplo de configuración del archivo AndroidManifest.xml.	35
Fig. 3.20. Ejemplo de configuración del archivo main.xml.	35
Fig. 3.21. Función OnlocationChanged.	36
Fig. 3.22. Función dumpLocation.	37
Fig. 4.1. Ejemplo de pasarela de XMPP.	41
Fig. 4.2. Función InitConnection.	43
Fig. 4.3. Función enviar.	44
Fig. 4.4. Función Desconectar.	44
Fig. 4.5. Esquema de funcionamiento de la aplicación Java.	45
Fig. 4.6. Imagen que muestra de las librerías Smack.	46
Fig. 4.7. Imagen del archivo hibernate.cfg.xml.	47
Fig. 4.8. Imagen del archivo contact.hbm.xml.	48
Fig. 4.9. Imagen del archivo Contact.java.	49
Fig. 4.10. Imagen del archivo pom.xml.	50
Fig. 5.1. Grafica de evolución de los diferentes servidores.	51
Fig. 5.2. Web del Sistema de geolocalización.	53

Fig. 5.3. Bases de datos de usuarios.....	53
Fig. 5.3. Bases de datos para el registro de la posición de los usuarios.....	54
Fig. 6.1. Ejemplo de Googlemaps.....	55
Fig. 6.2. Página para generar la Key de Google Maps.....	56
Fig. 6.3. Página de registro de Google Maps.....	56
Fig. 6.4. Parte del código donde está la clave de Google Maps.....	57
Fig. 6.5. Pantalla principal de la aplicación web, partes de menú.....	57
Fig. 6.6. Pantalla de la validación como administrador web.....	58
Fig. 6.7. Pantalla con las funciones del administrador.....	59
Fig. 6.8. Ejemplo de borrado de un registro.....	59
Fig. 6.9. Ejemplo de borrado de un registro.....	60
Fig. 6.10. Pantalla donde se aprecia la última posición del dispositivo.....	60
Fig. 6.11. Pantalla donde se puede visualizar el recorrido del dispositivo.....	61

Índice de tablas.

Tabla 10.1: Desglose de actividades.....	71
Tabla 10.2: Coste de actividades.	72
Tabla 10.3: Coste de material informático.	72
Tabla 10.4: Coste total.....	73

Glosario de términos.

XMPP	Protocolo extensible de mensajería y comunicación de presencia
PHP	Hypertext Pre-processor
GPS	Global Positioning System (sistema de posicionamiento global)
DGPS	Differential GPS, o GPS diferencial
GSM	groupe spécial mobile
SDK	Software Development Kit - Kit de desarrollo de software
RDS	Radio Data System
WAAS	Wide Area Augmentation System
EGNOS	European Geostationary Navigation Overlay Service
MSAS	Multi-functional Satellite Augmentation System
QWERTY	Es la distribución del teclado más común
FM	Frecuencia Modulada
IDE	Es un entorno de programación que consiste en un compilador, un gestor de código y un debugger.
SMS	Short Message Service servicio de mensajes cortos
ADT	Android Development Tools
JDK	Java Development Kit , es un software que provee herramientas de desarrollo para la creación de programas en java
PDA	Personal Digital Assistant, (Asistente Digital Personal), también denominado ordenador de bolsillo

TCP	Transmission Control Protocol , Protocolo de Control de Transmisión
HTTP	Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto
XML	eXtensible Markup Language, lenguaje de marcas extensible
SSL	Secure Sockets Layer, protocolo de la capa de conexión segura
TLS	Transport Layer Security seguridad de la capa de transporte
API	Application Programming Interface
MIDP	Mobile Information Device profile es una versión de Java integrada en el hardware de celulares relativamente modernos que permite el uso de programas tales como juegos, aplicaciones o todo tipo de software.

1. Objetivo.

El objetivo del proyecto es desarrollar un conjunto de aplicaciones que sean capaces de localizar en una pantalla del ordenador la posición del teléfono móvil, especialmente indicado para aquellas personas que puedan tener algún tipo de demencia, así como para la localización de menores.

El proyecto está dividido en tres partes:

La primera se compone de una aplicación para un Smartphone que permitirá la obtención de la señal del sistema GPS y la transmitirá haciendo uso de los servicios de datos que proporcionan las operadoras de telefonía.

La segunda está compuesta por una aplicación que realizara la función de recibir los datos enviados por el Smartphone e incorporarlos en una base de datos.

La tercera está compuesta de una aplicación web, que es la encargada de la administración de los usuarios y mostrará el recorrido que han realizado con su Smartphone.

1.1. ¿Cuál es la problemática de las personas con demencia?

Se entiende por demencia, la pérdida de la función cerebral que ocurre con ciertas enfermedades y afecta a la memoria, el pensamiento, el lenguaje, el juicio y el comportamiento.

Es por ello que las personas con demencia necesitan de ayudas externas para hacerles más fácil el día a día.

La demencia más frecuente es el Alzheimer, se trata de una enfermedad neurodegenerativa irreversible, que ataca al cerebro y produce un grave deterioro de las funciones cognitivas, conductual y funcional que condena a quien lo padece a depender de un cuidador las 24 horas del día.

Es una enfermedad ligada a la edad, si bien no es exclusiva de los ancianos. Actualmente en España hay unos 800.000 enfermos.

1.2. ¿A qué colectivo va dirigido?

Principalmente este proyecto está dirigido a personas con algún tipo de problema que les impida ubicarse en un entorno urbano o rural y que corran el peligro de perderse. La aplicación facilitara a los usuarios y a sus familiares la posibilidad de tener localizado a sus seres queridos.

Otras posibilidades seria:

A nivel de empresa podríamos seguir el recorrido de un vehículo de transporte (camión/furgoneta) en una ciudad o fuera de ella para saber si se sigue el itinerario correcto, organizar flotas para asegurar un área o periodo de cobertura y saber si realmente se está dando una calidad de servicio a un cliente (conocer la hora de salida /llegada de un autocar o llegar a conocer la frecuencia de paso de un autobús).

Saber si una persona mayor a andado y donde se encuentra ubicada en cualquier momento.

Para un deportista amateur podría por ejemplo, conocer la ruta que ha seguido en su entrenamiento.

1.3. ¿Cuál es la motivación para realizar este proyecto?

Si buscamos aplicaciones relacionadas con el GPS, en el mercado podemos encontrar multitud de ellas mencionadas en el punto 1.6 y con funcionalidades similares.

Con la intención de cubrir la necesidad de localizar la posición de una persona a través de su terminal móvil nació la idea de desarrollar la aplicación

El software que pretendemos desarrollar no quiere competir en funcionalidad con ninguno, pero sí proporcionar un servicio completo de posicionamiento. Las razones las detallamos a continuación:

- El desarrollo de la aplicación debe ser completamente gratuita, requiere un tráfico constante de datos, para ello se sugiere disponer de una conexión de datos en el terminal móvil.
- Para proporcionar una aplicación gratuita se decide que estará desarrollada para el sistema operativo Android.
- Para poder proporcionar un mejor servicio se dispondrá de una base de datos donde se almacenarán los puntos, la base de datos será Mysql.
- Para poder posicionar los puntos en un mapa utilizaremos las APIs de google maps.

1.4. ¿Qué se quiere conseguir?

Con este proyecto lo que se quiere conseguir es proporcionar a las personas necesitadas una herramienta que les facilite el día a día, proporcionándoles la tranquilidad de que siempre estarán localizables mediante su dispositivo móvil y aportando también tranquilidad al entorno familiar, ya que se acabarán aquellas largas esperas sin saber donde localizar a su familiar.

1.5. ¿Cómo funciona el sistema?

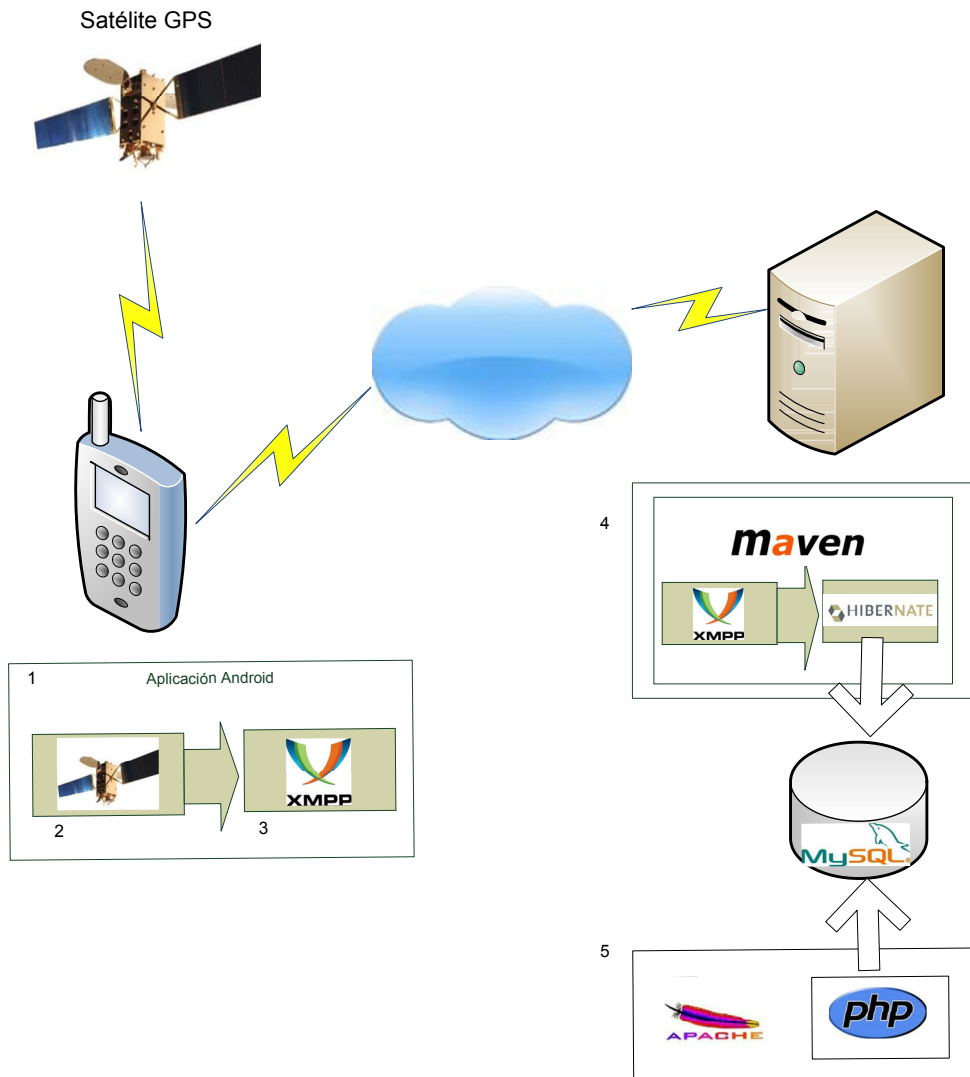


Fig. 1.1. Esquema funcional del sistema de geolocalización.

- 1- Aplicación Android
- 2- Módulo de GPS
- 3- Módulo emisor XMPP Android
- 4- Módulo receptor XMPP Java
- 5- Aplicación web

El usuario que disponga de un Smartphone podrá instalar una aplicación, la cual está compuesta de dos módulos importante:

Módulo GPS: es el encargado de la obtención de los datos enviados por el sistema de satélites GPS, una vez obtenida dicha información entrara en funcionamiento el siguiente modulo.

Módulo emisor XMPP: es el encargado de las comunicaciones entre el Smartphone y el servidor, utilizando el protocolo de comunicaciones XMPP

Los siguientes módulos estarán alojados en un servidor:

Módulo receptor XMPP: es el encargado de la recepción de los mensajes enviados desde los Smartphone hasta el servidor. La aplicación una vez se pone en marcha esta escuchando todas las comunicaciones que se le envían y almacena la información recibida (longitud, latitud, fecha, dirección, imei) en una base de datos Mysql.

La aplicación web: está alojada en un servidor Apache que proporciona los servicios de php y aloja también la base de datos Mysql. La web mostrara la posición del usuario sobre un mapa utilizando una API de Google Maps, gestiona la base de datos que almacena las diferentes localizaciones y gestiona a los usuarios.

1.6. Estado del Arte

Antes de empezar a desarrollar el proyecto se ha llevado a término un estudio de las webs que hay actualmente en la red y que ofrecen un servicio similar. Finalmente se han seleccionado las siguientes:

Google Latitude

Son aplicaciones de monitorización o tracking que funcionan mediante la red GSM; el usuario que quiere ser localizado manda voluntariamente su ubicación al servicio. Simultáneamente alguien previamente autorizado con una conexión a Internet puede situar en un mapa la posición de los múltiples usuarios que estén mandando la información.

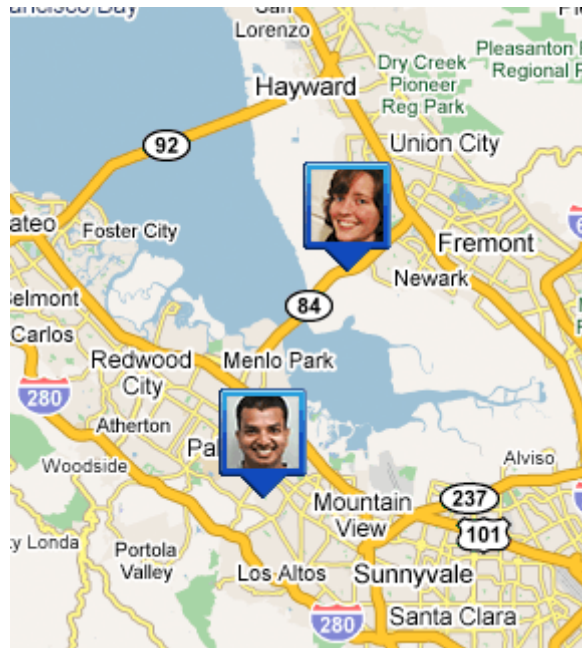


Fig. 1.1. Ejemplo del programa latitude.

Localizalotodo.com

También existe la aplicación para instalar en móviles en la página www.localizalotodo.com que entre otros servicios, proporciona la localización del móvil que hayan sido dado de alta en su servicio.

Localizador GPS de personas (Keruve)

Se trata de un reloj que lleva integrado un sistema GPS y un sistema GSM, para las comunicaciones.

Está destinado para personas con Alzheimer. Estas personas llevarán puesto el reloj, por otro lado el familiar dispone de un terminal desde donde puede visionar la posición sobre un mapa. Las comunicaciones se realizan mediante el envío de sms entre el terminal y el reloj.

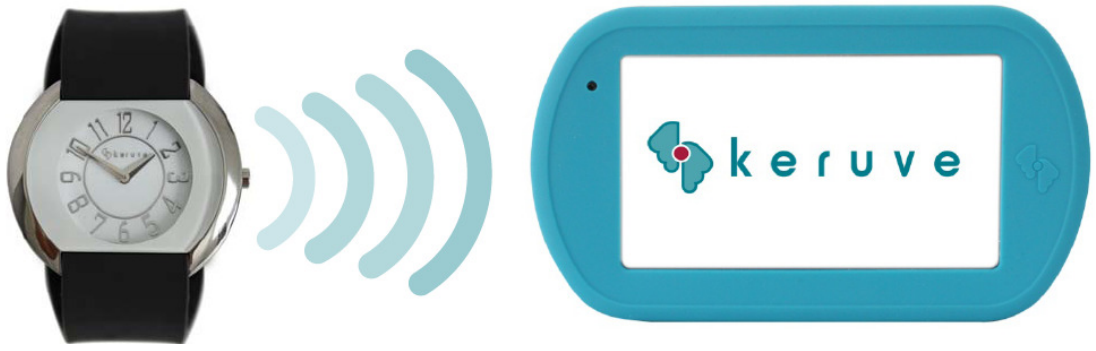


Fig. 1.3. Ejemplo del sistema keruve.

2. Funcionamiento básico del GPS

El sistema “GPS” nace en 1973 y queda oficialmente declarado como funcional en 1995. Es un sistema que inicialmente se desarrolló con enfoque de estrategia bélica pero a través de los años el gobierno de Estados Unidos decidió permitir el uso al público en general con ciertas limitaciones de exactitud.

Es un sistema utilizado en la actualidad por muchos otros sistemas e inclusive ya es una herramienta de trabajo, por ejemplo es utilizado en aeronaves, para guiarse en el espacio, por los geólogos para la medición de movimientos telúricos, por ingenieros para el control de estructuras como puentes colgantes y evidentemente por la fuerza militar y secreta de los Estados Unidos de América.

2.1. Historia y Cronología

La historia y cronología del sistema GPS desde sus predecesores hasta su etapa de implementación total.

1920's Orígenes de la radionavegación.

Principios de la II Guerra Mundial – LORAN, el primer sistema de navegación basado en la llegada diferenciada de señales de radio desarrollado por el laboratorio de Radiación de MIT. LORAN fue también el primer sistema de posicionamiento capaz de funcionar bajo cualquier condición climatológica pero es solamente bidimensional (latitud y longitud).

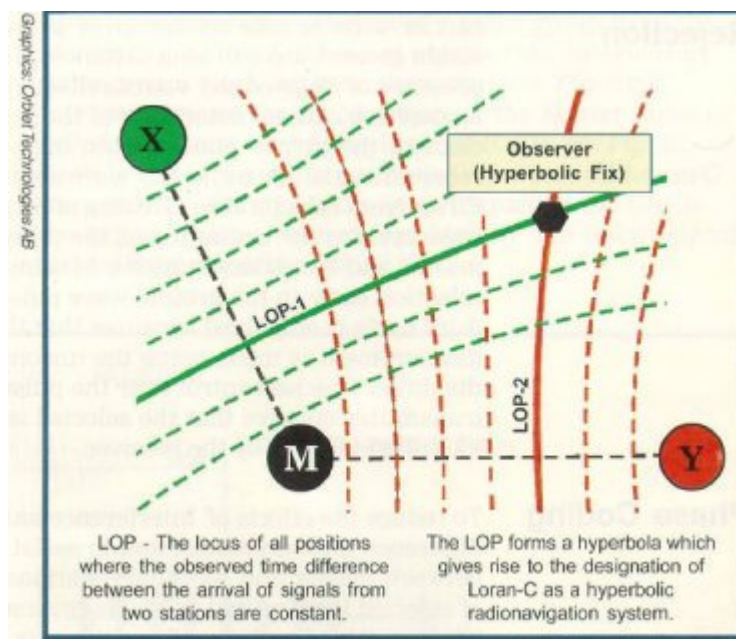


Fig. 2.1. Sistema de navegación LORAN.

1959 TRANSIT, el primer sistema operacional basado en satélites, fue desarrollado por Johns Hopkins (Laboratorio de Física Aplicada) bajo el Dr. Richard Kirschner. A pesar de que la intención de TRANSIT era dar soporte a la flotilla de la marina de Estados Unidos, las tecnologías empleadas para el sistema demostraron ser útiles para el sistema de posicionamiento global (GPS). El primer satélite fue lanzado en 1959.

1960 El primer sistema de posicionamiento de tres dimensiones es sugerido por Raytheon Corporation en necesidad de la fuerza aérea.

1963 La compañía aeroespacial lanzó un estudio en la utilización de un sistema espacial para el sistema de navegación para los vehículos en movimiento a gran velocidad y tres dimensiones; esto los llevó directamente al concepto de GPS. El concepto involucraba medir los tiempos de llegada de las señales de radio transmitidas por los satélites cuyas posiciones eran bien conocidas. Esto proporcionaba la distancia al satélite cuya posición era conocida que a la vez establecía la posición del usuario.

1963 La fuerza aérea da apoyo a este estudio bautizándolo Sistema 621B.

1964 Timation, un sistema de satélites, Naval es desarrollado por Roger Easton en los laboratorios de investigación Naval para el desarrollo de relojes de alta estabilidad, capacidad de transferencia de tiempo y navegación en dos dimensiones.

1968 El departamento de defensa de los Estados Unidos establece un comité llamado NAVSEG (Navigation Satellite Executive Comité) para coordinar los esfuerzos de diversos grupos de navegación satelital.

1971 El sistema 621B es probado por la fuerza aérea dando resultados de una precisión de centésimas de milla.

1973 El secretario de la defensa decide que los diferentes sistemas de navegación que se estaban creando, se unificaran y crearon un solo y robusto sistema de navegación.

1974 Junio. Rockwell international fue contratado como proveedor de los satélites GPS.

1974 Julio 14. El primer satélite de NAVSTAR fue lanzado

1978 El primer block de satélites fue lanzado. Un total de 11 satélites fueron lanzados entre 1978 y 1985. Un satélite fue perdido debido a una falla de lanzamiento.

1982 DoD decide reducir la constelación de satélites de 24 a 18.

1983 Después de la caída de una Unión Soviética, el gobierno de Estados Unidos informa que el sistema GPS podrá ser utilizado por las aeronaves civiles.

1988 El secretario de las Fuerzas Aéreas anuncia la expansión de la constelación de GPS de 18 a 21 satélites y tres repuestos.

1989 El primero del un block de 28 satélites es lanzado en Cabo Cañaveral, Florida

1990 Dod Activa SA – una degradación en la exactitud del Sistema de forma planeada. El sistema es probado en la guerra del Pérsico.

1991 El gobierno ofrece el sistema de GPS a la comunidad internacional sin costo durante los siguientes 10 años.

1993 El gobierno declara el sistema formalmente funcionando con sus 24 satélites en órbita.

1995 El gobierno de Estados Unidos, Bill Clinton se compromete mediante una carta a la ICAO a proveer las señales de GPS a la comunidad internacional

2.2. Funcionamiento

El sistema GPS funciona en cinco pasos lógicos: Triangulación, Medición de distancia, Tiempo, Posición y Corrección.

2.3. Triangulación

La posición se calcula en base a la medición de las distancias a los satélites. Matemáticamente se necesitan cuatro mediciones de distancia a los satélites para determinar la posición exacta. En la práctica se resuelve la posición con solo tres mediciones si podemos descartar respuestas ridículas o utilizamos ciertos trucos. Se requiere de todos modos una cuarta medición por razones técnicas que más adelante se verán.

2.4. Midiendo la distancia

La distancia al satélite se determina midiendo el tiempo que tarda una señal de radio, emitida por el mismo, en alcanzar nuestro receptor de GPS. Para efectuar dicha medición asumimos que ambos, nuestro receptor GPS y el satélite, están generando el mismo Código Pseudo Aleatorio, exactamente el mismo momento. Comparando cuanto retardo existe entre la llegada del Código Pseudo Aleatorio proveniente del satélite y la generación del código de nuestro receptor de GPS, podemos determinar cuánto tiempo le llevó a dicha señal llegar hasta nosotros. Multiplicamos dicho tiempo de viaje por la velocidad de la luz y obtenemos la distancia al satélite.

2.5. Obtención de un Timing Perfecto

Un timing muy preciso es clave para medir la distancia a los satélites. Los satélites son exactos porque llevan un reloj atómico a bordo. Los relojes de los receptores GPS no necesitan ser tan exactos porque la medición de un rango a un satélite adicional permite corregir los errores de medición.

2.6. Posicionamiento de los Satélites

Para utilizar los satélites como puntos de referencia debemos conocer exactamente donde están en cada momento. Los satélites de GPS se ubican a tal altura que sus órbitas son muy predecibles. El Departamento de Defensa controla y mide variaciones menores en sus órbitas. La información sobre errores es enviada a los satélites para que estos a su vez retransmitan su posición corregida junto con sus señales de timing.

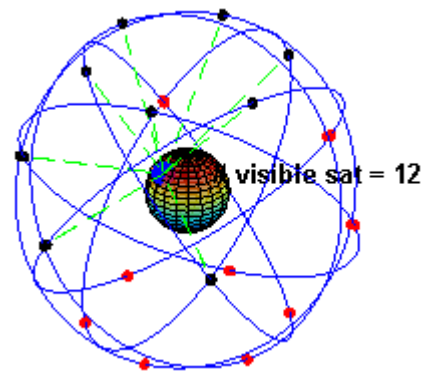


Fig. 2.2. Posicionamiento de los satélites.

2.7. Corrección de Errores

La ionosfera y la troposfera causan demoras en la señal de GPS que se traducen en errores de posicionamiento. Algunos errores se pueden corregir mediante modelación y correcciones matemáticas. La configuración de los satélites en el cielo puede magnificar otros errores. El GPS Diferencial puede eliminar casi todos los errores.

El DGPS (Differential GPS), o GPS diferencial, es un sistema que proporciona a los receptores de GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada. Se concibió fundamentalmente debido a la introducción de la disponibilidad selectiva (SA).



Fig. 2.3. Imagen de un sistema de posicionamiento diferencial.

El fundamento radica en el hecho de que los errores producidos por el sistema GPS afectan por igual (o de forma muy similar) a los receptores situados próximos entre sí. Los errores están fuertemente correlacionados en los receptores próximos.

Un receptor GPS fijo en tierra (referencia) que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y puede calcular los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia.

La estructura DGPS quedaría de la siguiente manera:

- Estación monitorizada (referencia), que conoce su posición con una precisión muy alta. Esta estación está compuesta por:
 - Un receptor GPS.
 - Un microprocesador, para calcular los errores del sistema GPS y para generar la estructura del mensaje que se envía a los receptores.
 - Transmisor, para establecer un enlace de datos unidireccional hacia los receptores de los usuarios finales.

- Equipo de usuario, compuesto por un receptor DGPS (GPS + receptor del enlace de datos desde la estación monitorizada).

Existen varias formas de obtener las correcciones DGPS. Las más usadas son:

- Recibidas por radio, a través de algún canal preparado para ello, como el RDS en una emisora de FM.
- Descargadas de Internet, o con una conexión inalámbrica.
- Proporcionadas por algún sistema de satélites diseñado para tal efecto. En Estados Unidos existe el WAAS, en Europa el EGNOS y en Japón el MSAS, todos compatibles entre sí.

En los mensajes que se envían a los receptores próximos se pueden incluir dos tipos de correcciones:

- Una corrección directamente aplicada a la posición. Esto tiene el inconveniente de que tanto el usuario como la estación monitorea deberán emplear los mismos satélites, ya que las correcciones se basan en esos mismos satélites.
- Una corrección aplicada a las pseudodistancias de cada uno de los satélites visibles. En este caso el usuario podrá hacer la corrección con los 4 satélites de mejor relación señal-ruido (S/N). Esta corrección es más flexible.

El error producido por la disponibilidad selectiva (SA) varía incluso más rápido que la velocidad de transmisión de los datos. Por ello, junto con el mensaje que se envía de correcciones, también se envía el tiempo de validez de las correcciones y sus tendencias. Por tanto, el receptor deberá hacer algún tipo de interpolación para corregir los errores producidos.

Si se deseara incrementar el área de cobertura de correcciones DGPS y, al mismo tiempo, minimizar el número de receptores de referencia fijos, será necesario modelar las variaciones espaciales y temporales de los errores. En tal caso estaríamos hablando del GPS diferencial de área amplia.

Con el DGPS se pueden corregir en parte los errores debidos a:

- Disponibilidad selectiva (eliminada a partir del año 2000).
- Propagación por la ionosfera - troposfera.
- Errores en la posición del satélite (efemérides).
- Errores producidos por problemas en el reloj del satélite.

Para que las correcciones DGPS sean válidas, el receptor tiene que estar relativamente cerca de alguna estación DGPS, generalmente a menos de 1000 km. Las precisiones que manejan los receptores diferenciales son centimétricas, por lo que pueden ser utilizados en ingeniería.

Resumen acerca del funcionamiento del GPS.

- a. Triangulación. La base del GPS es la "triangulación" desde los satélites
- b. Distancias. Para "triangular", el receptor de GPS mide distancias utilizando el tiempo de viaje de señales de radio.
- c. Tiempo. Para medir el tiempo de viaje de estas señales, el GPS necesita un control muy estricto del tiempo y lo logra con ciertos trucos.
- d. Posición. Además de la distancia, el GPS necesita conocer exactamente donde se encuentran los satélites en el espacio. Orbitas de mucha altura y un cuidadoso control, le permiten hacerlo.
- e. Corrección. Finalmente el GPS debe corregir cualquier demora en el tiempo de viaje de la señal que esta pueda sufrir mientras atraviesa la atmósfera.

3. Entorno del terminal móvil

En los siguientes puntos se analizan los diversos sistemas operativos que incorporan los Smartphone actuales y un lenguaje de programación más adecuado para el desarrollo de una aplicación que se ubicara en el Smartphone.

3.1. Sistemas operativos

La primera tarea de análisis consiste en decidir para qué plataforma o sistema operativo se iba a programar la aplicación. Para ello identificamos los sistemas más populares.

Los sistemas más populares son los siguientes:

- Android
- BlackBerry OS
- Symbian OS
- Windows Phone
- iOS

En la siguiente tabla podemos apreciar el estado actual en el mercado como la evolución que está prevista que tengan los diferentes sistemas operativos.

Worldwide Mobile Communications Device Open OS Sales to End Users by OS (Thousands of Units)				
OS	2010	2011	2012	2015
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research In Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	296,647	467,701	630,476	1,104,898

Source: Gartner (April 2011)

Fig. 3.1. Comparativa entre sistemas operativos.

La figura anterior contiene los datos de ventas de teléfonos móviles, según el sistema operativo. Estos datos los proporciona la empresa Gartner, que es una empresa especializada en todo tipo de estudios de mercado relacionados con las tecnologías de la información. La tabla resulta una orientación suficientemente válida para resolver la disyuntiva de sobre qué sistema operativo programar la aplicación. En la tabla podemos apreciar como Android es el primer sistema operativo con diferencia seguido de iOS y Microsoft.

3.1.1. Android:

Es un sistema operativo inicialmente desarrollado por Android Inc., una firma adquirida por Google en el 2005. Este sistema operativo está basado en una versión modificada del Kernel de Linux.

Al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, Android se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde podemos ver problemas aún no resueltos y reportar problemas nuevos.



Fig. 3.2. Logotipo de android.

Lo bueno:

En definitiva al ser desarrollado de forma abierta, se trata de una ventaja tanto para los que desarrollan sus aplicaciones como para sus usuarios. Se puede personalizar el teléfono al máximo y modificar funciones del teléfono simplemente instalando una aplicación.

Otro punto a favor de Android es la increíble confianza que está recibiendo de los fabricantes. Gracias a ello, la oferta de teléfonos con Android es amplia y variada tanto en marcas como en precios.

Lo malo:

Hoy en día uno de los aspectos negativos de Android es su fragmentación, que aunque va mejorando, actualizar el sistema operativo a nuevas versiones no es tan fácil. También comparando con otros sistemas operativos, la cantidad de juegos disponible para Android es menor.

3.1.2. iOS:

Es el sistema operativo utilizado por el iPhone. Está basado en una variante del Mach kernel que se encuentra en Mac OS X. El iOS incluye el componente de software “Core Animation” de Mac OS X v10.5 que, junto con el PowerVR MBX el hardware de 3D, es responsable de las animaciones usadas en el interfaz de usuario.

iOS tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de Servicios Principales, la capa de Medios de comunicación y la capa de Cocoa Touch. El sistema operativo ocupa bastante menos de medio gigabyte del total del dispositivo, de 8 GB o de 16 GB. Esto se realizó para poder soportar futuras aplicaciones de Apple.



Fig. 3.3. Logotipo de iOS.

Lo bueno:

Buen diseño, funcionalidad, facilidad de uso, variedad de aplicaciones y juegos, lo convierten en un referente.

Su perfecta integración con los servicios de internet y equipos de sobremesa, especialmente Mac, es otro de sus puntos fuertes.

Lo malo:

El sistema de Apple es cerrado, por lo que hay menos posibilidades de cambiar la forma de funcionar del teléfono y un control más rígido de las aplicaciones publicadas. Además, si quieres disfrutar de un iPhone, sólo te queda desembolsar un precio bastante alto puesto que sólo hay un fabricante y un modelo.

3.1.3. Windows Phone:

Anteriormente llamado Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes (Smartphone) y otros dispositivos móviles.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente. Además, existe una gran oferta de software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de Windows Marketplace for Mobile.



Fig. 3.4. Logotipo de Windows Phone.

Lo bueno:

Un diseño moderno, práctico, atractivo y con características innovadoras han sorprendido ya a más de uno. Windows Phone cuenta con una gran inversión y se ha diseñado para competir con los más grandes, el resultado es un sistema moderno y capaz.

Lo malo:

La variedad de móviles con Windows Phone no es tan amplia como la que ofrecen Android o Symbian, aunque está en crecimiento.

La cantidad de aplicaciones disponibles en estos momentos es baja, aunque están facilitando el trabajo a los desarrolladores para llenar el hueco rápidamente.

3.1.4. Blackberry:

BlackBerry dispone de un sistema operativo multitarea (OS), lo que permite un uso intensivo de los dispositivos de entrada disponibles en los teléfonos, en particular la rueda de desplazamiento y el trackpad. El sistema operativo proporciona soporte para Java MIDP 1.0 y WAP 1.2.

Las versiones anteriores permitían la sincronización inalámbrica con Microsoft Exchange Server para el correo electrónico y calendario, al igual que Lotus Domino e-mail. El actual OS 5.0 proporciona un subconjunto de MIDP 2.0, y permite la activación inalámbrica completa y la sincronización con Exchange de correo electrónico, calendario, tareas, notas y contactos, y añade un soporte para Novell GroupWise y Lotus Notes.



Fig. 3.5. Logotipo de Blackberry.

Lo bueno:

Perfecto para el uso de correo electrónico, Blackberry destaca también por los aspectos de seguridad y por sus teclados QWERTY que, al estilo de un teclado de PC, permiten una escritura muy rápida.

Lo malo:

No se puede comparar su tienda de aplicaciones con las de Android o iTunes.

Tampoco existen tantas posibilidades de elección en cuanto a dispositivos y el potencial multimedia no es su fuerte principal.

3.1.5. Symbian OS

Symbian es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia, Sony Ericsson, Psion, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA's y Handhelds de PSION.

El objetivo de Symbian fue crear un sistema operativo para terminales móviles que pudiera competir con el de Palm o el Windows Mobile de Microsoft y ahora Android de Google Inc., iOS de Apple Inc. y Blackberry 6 RIM.



Fig. 3.6. Logotipo de Symbian OS.

Lo bueno:

Symbian ha sido siempre fiable e innovador. Con fuerte énfasis en las funciones básicas de telefonía y multimedia de sus dispositivos, también cuenta con un amplio mercado de aplicaciones externas y con una amplia variedad de dispositivos disponibles.

Se trata de una excelente opción para conseguir terminales de gama media y baja, debido a su fiabilidad, una cantidad razonable de buenas aplicaciones y posibilidades multimedia.

Lo malo:

Symbian ha perdido protagonismo con la llegada de iPhone y Android, sobre todo en los Smartphones punteros. Hay muchas y muy buenas aplicaciones para Symbian, pero no se puede comparar con la cantidad de oferta de nuevas aplicaciones de la competencia.

3.2. ¿Por qué Android?

Después de tener en cuenta los factores relativos al mercado, las perspectivas de futuro de cada uno de los sistemas operativos así como el estado actual del sistema operativo, a de tratarse de una tecnología que facilita el trabajo a los desarrolladores, proporcionando una amplia documentación oficial y no oficial.

Es por ello que Android es el sistema operativo más oportuno para llevar a cabo la programación de la aplicación.

3.3. ¿Qué es Android?

Android es un Sistema operativo, es decir el programa principal de un ordenador móvil o cualquier aparato que utilicemos.

Es un paquete de software basado en código abierto para teléfonos móviles, creado por Google y la Open Handset Alliance. Se encuentra dentro de millones de teléfonos y otros dispositivos móviles, algo que convierte a Android en una de las plataformas principales para desarrollar aplicaciones.

Es una plataforma de desarrollo completamente abierta y gratuita basada en Linux y en código abierto. Código abierto es aquel código que es conocido por todo el mundo y que cualquiera con unos mínimos conocimientos para ello puede cambiarlo y hacerlo a su medida.

Existen diferentes versiones de Android que se detallan a continuación:

Android 1.5 (Cupcake): incluyó numerosas mejoras, compatibilidad con teclados de software en la pantalla, grabación de videos y widgets.

Android 1.6 (Donut): Incorporó la compatibilidad con pantallas de alta y baja densidad.

Android 2.0 (Eclair): Añadió compatibilidad con la funcionalidad multitáctil, teclas virtuales, gestión de cuentas centralizadas, APIs de sincronización y HTML5. Fue sustituido rápidamente por Android 2.0.1 (se llamo igual, Eclair), que mejoro algunos problemas de la versión 2.0.

Android 2.1 (Eclair Maintenance Release 1, Versión de mantenimiento Eclair), incorporó la compatibilidad de fondos interactivos y mejoró la compatibilidad con HTML5.

Android 2.2 (Froyo): Permite la instalación de aplicaciones en dispositivos de almacenamiento externo (tarjetas SD), una maquina virtual mucho más rápida y APIs OpenGL ES 2.0.

3.4. Lenguaje programación

Para trabajar con Android el lenguaje de programación más óptimo es Java, ya que la mayoría de programas para Android están escritos en este lenguaje y utilizan las API de la biblioteca Java 5 Standard Edition (SE).

Android también utiliza un compilador estándar Java para compilar el código fuente, lo que lo hace que Java sea un lenguaje completamente integrado en Android.

3.5. Entorno de programación

Después de decidir el sistema operativo y el lenguaje de programación de la aplicación, necesitamos una herramienta para poder programar en este lenguaje.

Las IDE¹ más populares en Android son Eclipse y NetBeans. Ambas son gratuitas y de código abierto. Para escoger uno nos basamos en la amplia información recopilada que determina la compatibilidad entre el entorno de desarrollo y el emulador o el terminal sin tener que realizar pasos intermedios.

Finalmente se decide por Eclipse por la amplia aceptación que tiene entre los programadores y la amplia documentación que existe para trabajar con este entorno.

3.6. Instalación de Eclipse

3.6.1. Java

En primer lugar se recomienda tener instalada una copia de Java, esta debe ser la versión JDK² 5 o superior.

También es necesaria la instalación del JRE³ que se recomienda la descarga de la última versión del JDK SE 6.0 de Sun. La descarga se puede efectuar desde la siguiente dirección (<http://java.sun.com/javase/downloads>).

Para verificar la versión que está instalada en nuestro equipo basta con ejecutar el siguiente comando en una ventana de sistema.

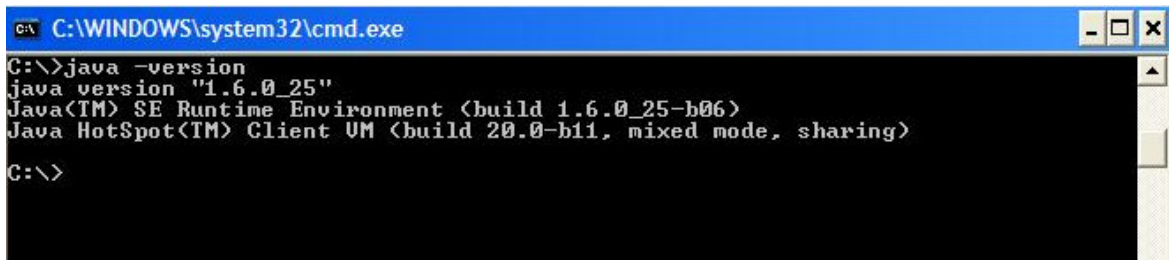
```
C:\> java -versión
```

Para el buen funcionamiento del sistema se desea una versión 1.6 o superior, como podemos ver en la siguiente imagen.

¹ Una IDE es un entorno de programación que consiste en un compilador, un gestor de código y un debugger.

² Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java

³ JRE, es un conjunto de utilidades que permite la ejecución de programas Java.

A screenshot of a Windows command prompt window. The title bar reads 'C:\WINDOWS\system32\cmd.exe'. The command prompt shows the following text:

```
C:\>java -version
java version "1.6.0_25"
Java(TM) SE Runtime Environment (build 1.6.0_25-b06)
Java HotSpot(TM) Client VM (build 20.0-b11, mixed mode, sharing)
C:\>
```

Fig. 3.7. Resultado de la verificación de java –versión.

3.6.2. Eclipse

Para instalar eclipse se recomienda instalar la última versión que en estos momentos es la 3.6.2, dicha versión se puede descargar gratuitamente desde la pagina de descargas de eclipse, <http://www.eclipse.org/downloads/>, donde se encontrará el paquete con el siguiente nombre: Eclipse IDE for Java Developers.

Una vez localizado la página y el paquete indicado se debe descargar en el escritorio temporalmente. Una vez todo el paquete este descargado, se debe descomprimir en una carpeta, en algún lugar conocido como puede ser c:\Eclipse.

3.6.3. SDK de Android

SDK⁴ es otra de las partes necesarias para el desarrollo de una aplicación en el sistema operativo Android.

Visite la página <http://d.android.com/sdk>, desde donde se podrá descargar la última versión.

Una vez se ha descargado el paquete, se debe descomprimir en una ubicación permanente. También se debe añadir el directorio /tools del SDK a la variable Path de Windows. Para ello, se pulsa con el botón derecho sobre Mi PC y selecciona Propiedades.

⁴ Software Development Kit - Kit de desarrollo de software

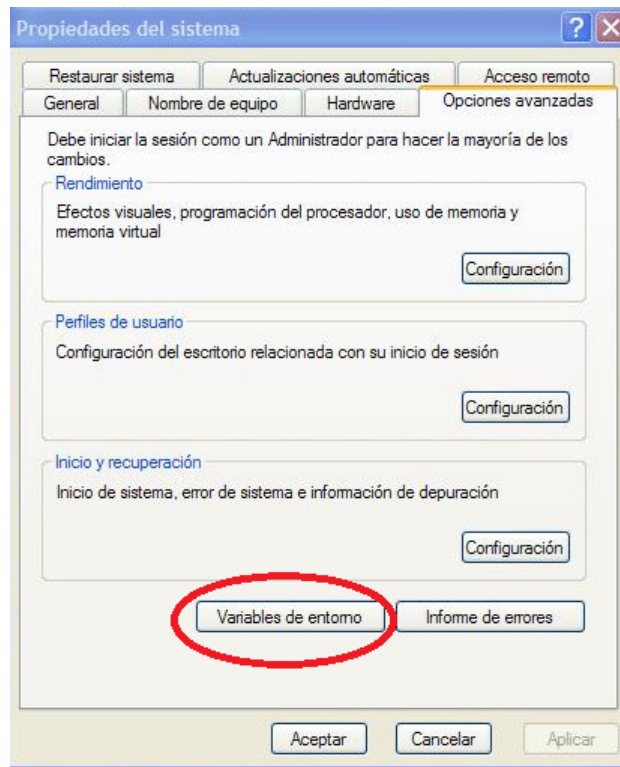


Fig. 3.8. Ventana de propiedades del sistema.

En la pestaña de Opciones avanzadas, se debe seleccionar Variable de entorno y Path en las variables del sistema. Se hace click en Modificar y añade al final (poniendo un punto y coma de separación) la ruta completa del directorio tools del SDK será: `C:\Archivos de programa \android-sdk-windows-1.5_r2\tools\;`”.

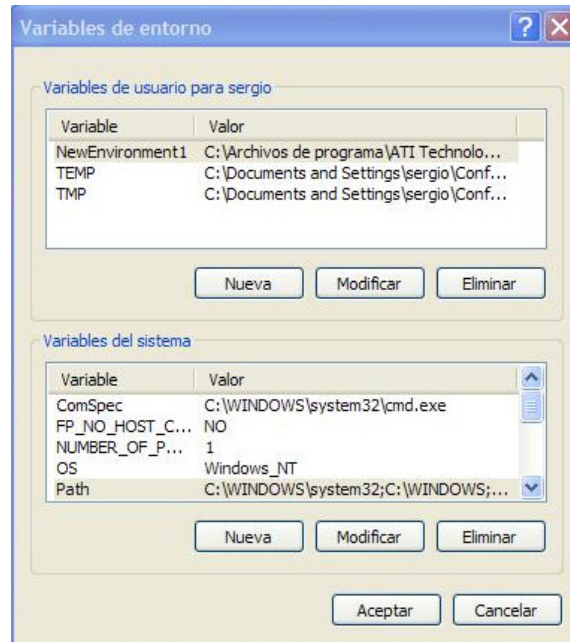


Fig. 3.9. Variables de entorno.

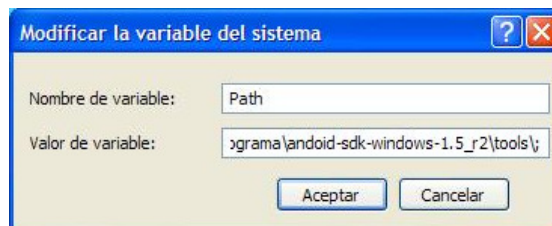


Fig. 3.10. Ventana de modificación de las variables de entorno.

A continuación, de debe seleccionar el archivo SDK Manager.exe que nos mostrará una serie de componentes disponibles, incluyendo documentación, librerías y controladores de USB. Todos los componentes que aparecen en el listado de las pantallas siguientes deben ser instalados en el directorio de instalación del SDK.

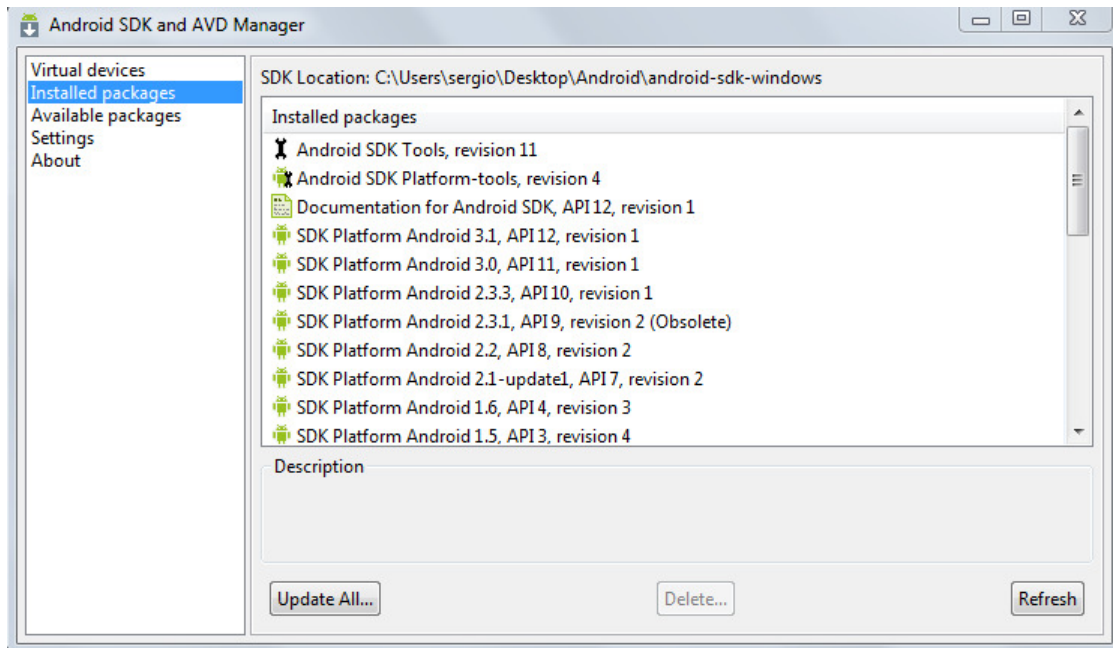


Fig. 3.11. Android SDK and AVD Manager.

Para la correcta instalación de todos los componentes se debe seleccionar la opción remarcada en la siguiente figura.

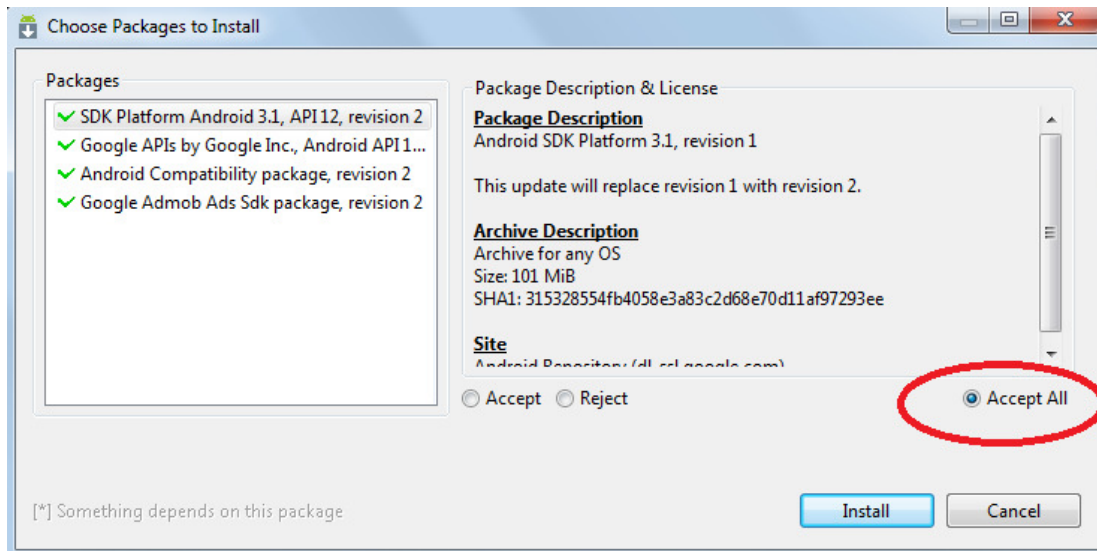


Fig. 3.12. Ventana de instalación de los componentes del SDK.

3.6.4. ADT (Android Development Tools)

ADT es un complemento que google ha escrito para eclipse. Para instalarlo se deben seguir los siguientes pasos:

- Abra el eclipse ejecutando el archivo eclipse.exe
- Seleccione el menú Help y haga click en Install New Software.

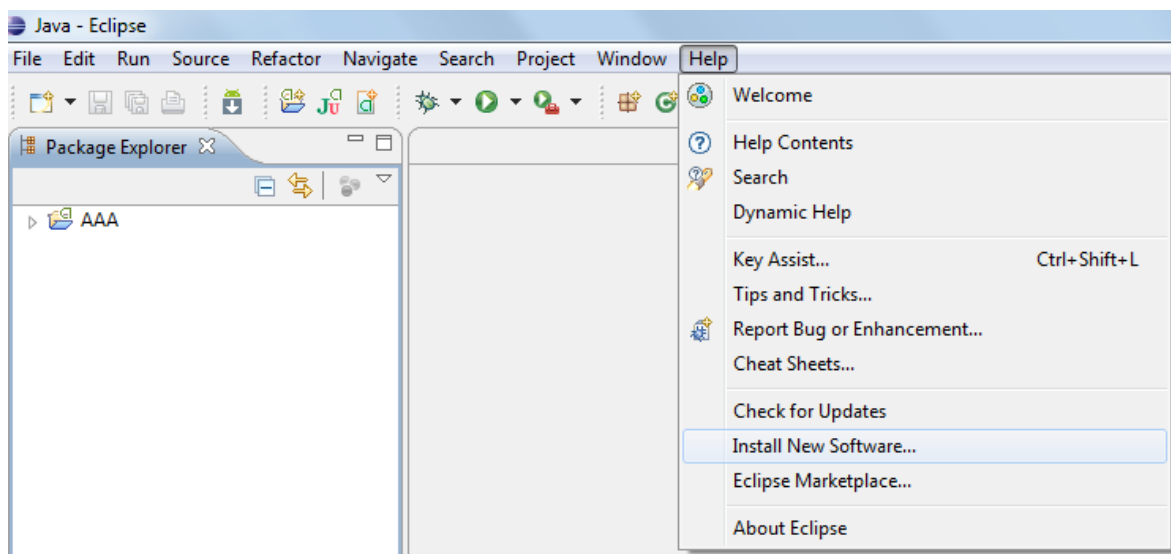


Fig. 3.13. Instalación del nuevo software en el Eclipse.

- Introduzca la ubicación del sitio Web de actuación de Android Development Tools: <http://dl-ssl.google.com/Android/eclipse/>.

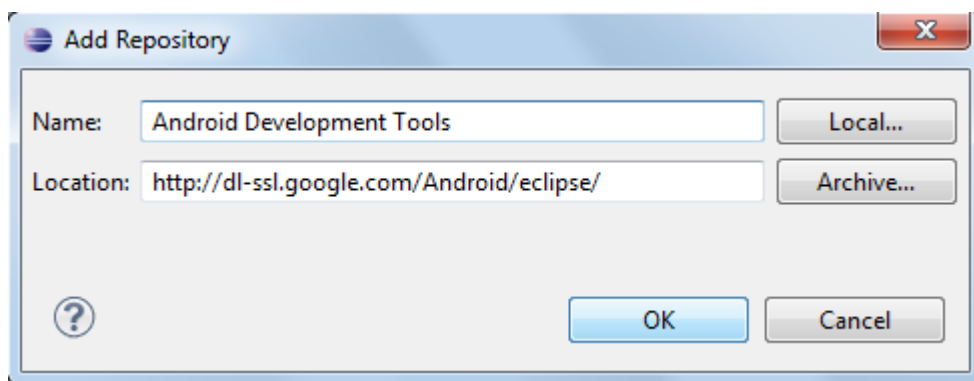


Fig. 3.14. Añadir repositorio en el Eclipse.

- Seleccione en el apartado Work With, Android Development Tools.

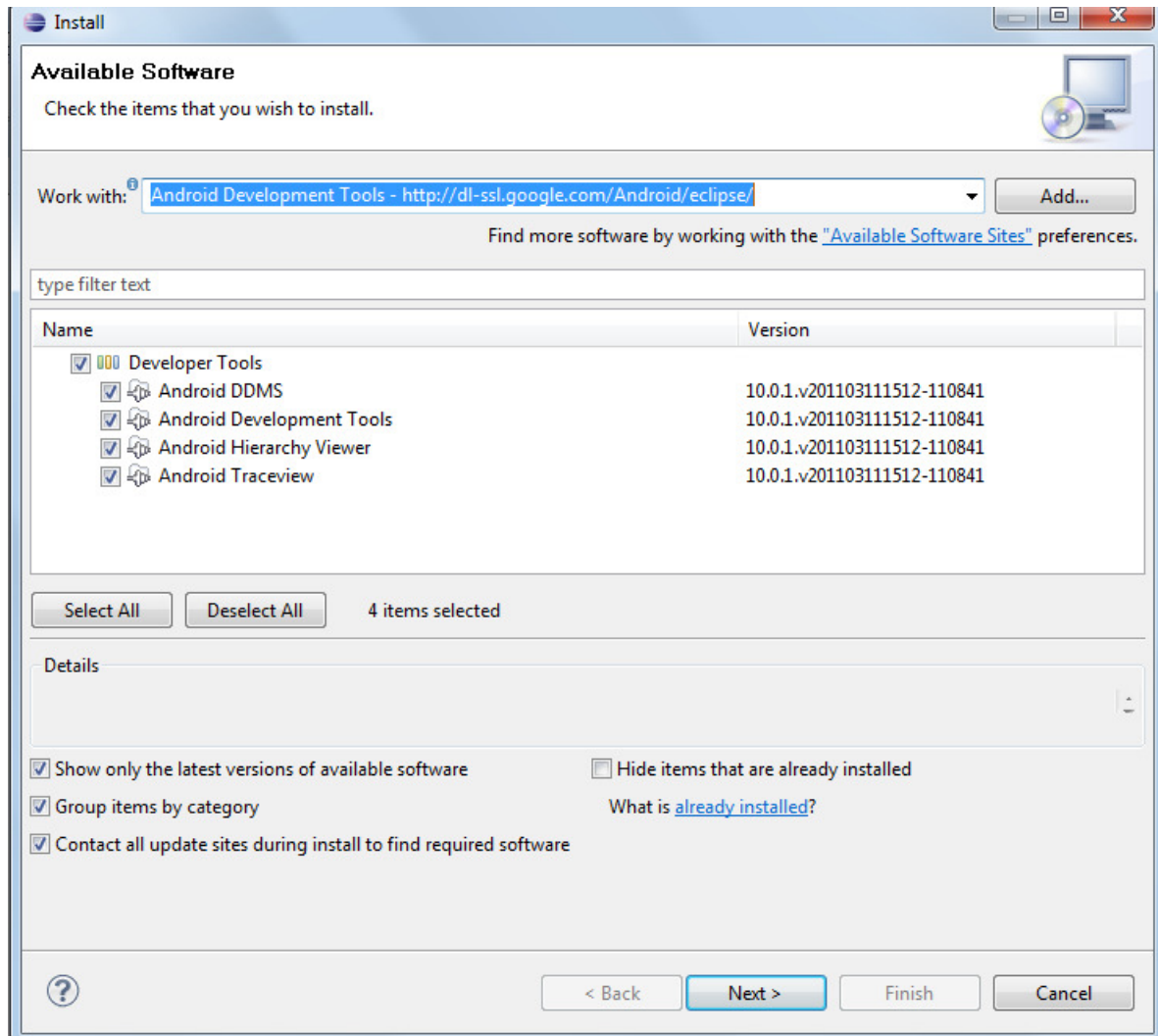


Fig. 3.15. Ventana de instalación de nuevo Software.

- Selecciona la casilla Developer Tools y haga click en Next, repase la lista de elementos que serán instalados y haga click en Next otra vez, acepte el contrato de la licencia y haga clic en Finish, para empezar la descarga y el proceso de instalación.
- Una vez finalizada la instalación se debe reiniciar el Eclipse.
- En el caso de aparecer algún tipo de mensaje de error, es necesario indicarle al Eclipse donde se encuentra ubicada el SDK de Android. Seleccione Windows>Preferences>Android e introduzca el directorio de instalación del SDK que se muestra en la figura 3.10.

3.7. Aplicación Android

Para la creación de la aplicación Android se han tenido dos puntos en cuenta:

- Uno la captación de la señal del satélites GPS, a través del dispositivo GPS que incorpora el dispositivo móvil
- Dos el envío de la información mediante el protocolo XMPP, haciendo uso de las tecnología de envío de datos a través de los dispositivos móviles.

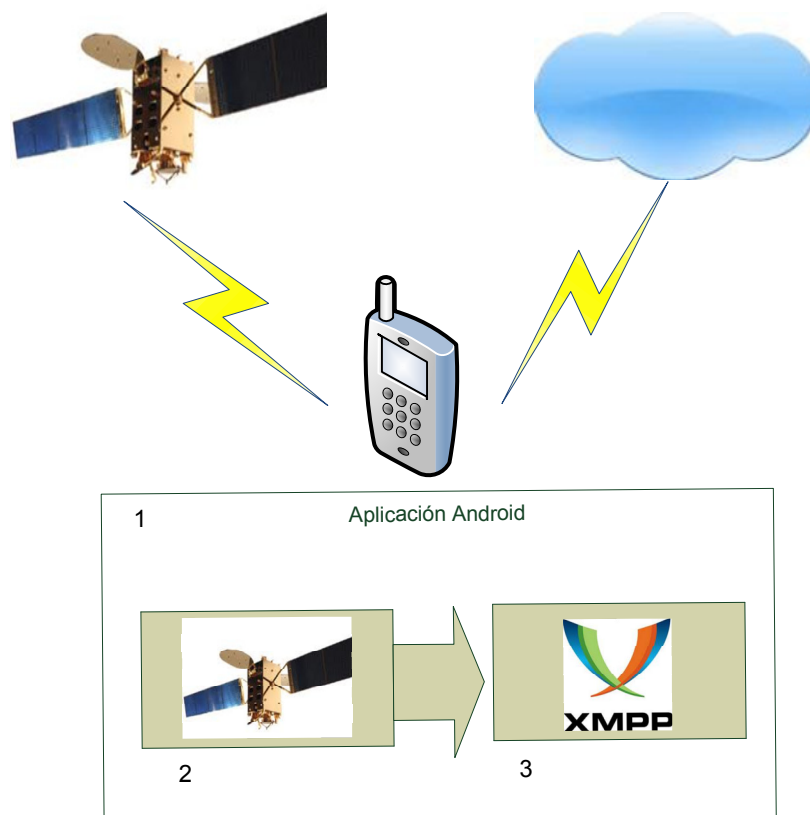


Fig. 3.16. Esquema de funcionamiento del sistema de Geolocalización.

- 1- Aplicación Android
- 2- Módulo de GPS
- 3- Módulo emisor XMPP Android

La aplicación no dispone de ningún tipo de configuración ya que está pensada para ser instalada y con un solo clic empezar a funcionar.

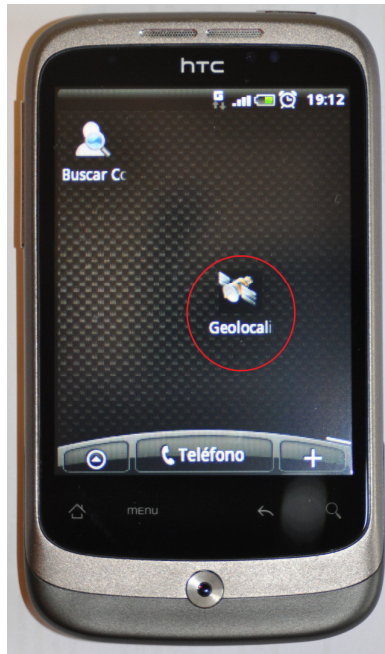


Fig. 3.17. Imagen de la aplicación instalada en un terminal HTC.

La aplicación una vez está en marcha envía un paquete de datos que incorpora la siguiente información (longitud, latitud, imei y fecha) cada vez que detecta un cambio de posición. El envío de los datos así como información adicional necesaria para que el usuario sea registrado en la web se muestra junto con el primer mensaje enviado, en los siguientes mensajes se muestra en la pantalla el número de envíos que se ha realizado y la información que es enviada al cliente que está alojado en el servidor.

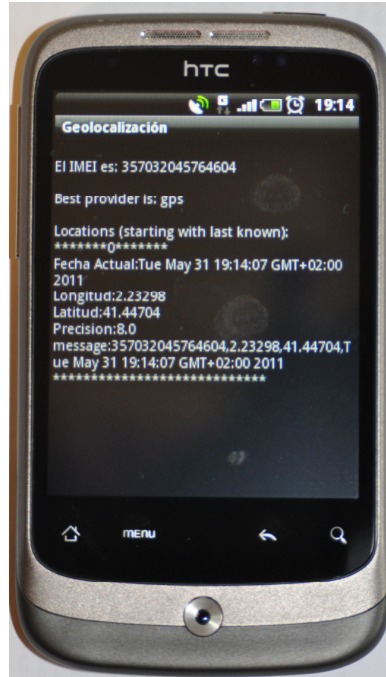


Fig. 3.18. Imagen de la aplicación ejecutada en un terminal HTC.

3.7.1. Modulo GPS

El módulo GPS es el encargado de la recepción de la señal que envían los 31 satélites que están moviéndose alrededor de la tierra.

Para ello, es necesaria la configuración de una serie de archivos que componen la aplicación de Android:

AndroidManifest.xml es uno de ellos, donde determinaremos que permisos tendrá nuestra aplicación, existen diversos permisos que dependiendo del uso que se le dé a la aplicación se podrán activar:

INTERNET: Para acceder a internet

READ_CONTACT: Para leer los datos de contactos de usuario.

WRITE_CONTACT: Para Escribir los datos de contacto del usuario.

RECEIVE_SMS: Para monitorizar los mensajes SMS entrantes.

ACCESS_COARSE_LOCATION: Para utilizar la posición aproximada que ofrecen las torres de telefonía móvil.

ACCESS_FINE_LOCATION: Permite utilizar los sistemas de posicionamiento más exactos como son el GPS.

READ_PHONE_STATE: Permite leer a información propia del teléfono (imei).

```
</application>
<uses-sdk android:minSdkVersion="7" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.READ_PHONE_STATE" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
</manifest>
```

Fig. 3.19. Ejemplo de configuración del archivo AndroidManifest.xml.

Para poder mostrar la información en la pantalla del teléfono se ha configurado el archivo main.xml y se le ha añadido la propiedad Scrollview para poder ir insertando todos los mensajes en la misma pantalla y poder desplazarla hacia arriba y hacia abajo.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  >
  <TextView
    android:id="@+id/output"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    />
</ScrollView>
```

Fig. 3.20. Ejemplo de configuración del archivo main.xml.

La parte más importante del programa se encuentra en Geolocalización.java que es propiamente el núcleo del programa, donde están definidas las librerías necesarias para la obtención de la posición mediante el uso del sistema GPS. También están definidas las funciones importantes para el desarrollo del programa.

```
import android.R.layout;
import android.R.integer;
import android.app.Activity;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.location.LocationProvider;
import android.os.Bundle;
import android.os.Handler;
import android.widget.TextView;
```

El método `onLocationChanged` es activado cada vez que se detecta que la posición ha cambiado.

```
// Se llama cuando la ubicacion ha cambiado
public void onLocationChanged(Location location) {
    dumpLocation(location);
}
```

Fig. 3.21.Funcion OnlocationChanged.

Para poder hacer las actualizaciones de las posiciones es necesario el método `requestLocationUpdates`, para ello se invoca este método desde el objeto `LocationManager(mgr)`.

```
mgr.requestLocationUpdates(best, 15000, 1, this);
```

Los datos que tiene en cuenta el método `requestLocationUpdates` son un proveedor, un retraso, una distancia mínima y un objeto.

Una vez se dispone de la información desde la función `dumpLocation`, se procesa la información recibida, se le aplica un redondeo a la longitud y a la latitud, se obtiene la fecha actual, se recupera el imei obtenido anteriormente para poder crear el paquete de información, que es necesario enviar para poder ser localizado en la web de geolocalización.

```
private void dumpLocation(Location location) {
    if (location == null)
        log("\nLocation[unknown]");
    else{
        //log("\n" + location.toString());
        log("*****"+ conter +"*****");
        Double lat = redondear(location.getLatitude(),5);
        latitud = lat.toString();
        Double longi = redondear(location.getLongitude(),5);
        longitud = longi.toString();

        java.util.Date fechaActual= new java.util.Date();
        Fecha=fechaActual.toString();

        log("Fecha Actual:"+ fechaActual.toString());
        log("Longitud:"+ longitud);
        log("Latitud:"+ latitud);
        log("Precision:"+ location.getAccuracy());
        message=(imei+", "+longitud+", "+latitud+", "+Fecha);
        log("message:"+message);
        log("*****");
        enviar("pfc201011@gmail.com",message);
        conter= conter+1;
    }
}
```

Fig. 3.22. Función dumpLocation.

4. XMPP

4.1. ¿Qué es XMPP?

XMPP (*eXtensible Messaging and Presence Protocol*, en castellano, Protocolo extensible de mensajería y comunicación de presencia) es un protocolo abierto y extensible basado en XML, originalmente ideado para mensajería instantánea. Es el protocolo principal en el que está basada la tecnología *Jabber*. Con el protocolo XMPP queda establecida una plataforma para el intercambio de datos XML que puede ser usada en aplicaciones de mensajería instantánea. Por lo tanto, los beneficios que aporta XML son heredados por el protocolo XMPP.

A diferencia de los protocolos propietarios de intercambio de mensajes como *ICQ*, *Y!* y *MSN Messenger*, XMPP se encuentra documentado y se insta a utilizarlo en cualquier proyecto. Existen servidores y clientes libres que pueden ser usados sin coste alguno. Por poner un ejemplo, este es el protocolo que seleccionó Facebook, Tuenti y *Google* para su servicio de mensajería *Google Talk*.

XMPP tiene sus bases en el proyecto Jabber de Jeremie Miller, en 1998. Después de la liberación de software de Jabber en el 2000, XMPP fue publicado como RFC 3920.

4.2. Funcionamiento

XMPP es un servicio descentralizado, no existe ningún servidor general, cualquiera puede poner en marcha un servidor propio, el puerto estándar es el 5222.

Cada usuario en la red XMPP tiene un único identificador (Jabber ID, normalmente abreviado como JID). El JID está estructurado como una dirección de correo, con un nombre mas una dirección DNS, separado por el signo @ (nombredeusuario@dominio.com).

Cuando un usuario quiere estar identificado en distintos lugares, el servidor permite al cliente especificar una cadena de referencia conocida como recurso, que identificara al cliente que está utilizando el usuario (por ejemplo: casa, trabajo).Esto será incluido en el JID añadiendo el carácter “/” seguido del nombre del recurso. Por ejemplo

nombredeusuario@dominio.com/trabajo, los mensajes con esta dirección serán dirigidos al cliente habilitado en el trabajo.

4.2.1. Funcionamiento del Cliente

El cliente Jabber/XMPP comienza un flujo “stream” mandando la etiqueta XML <stream:stream> al servidor, esto representa el inicio de una sesión del cliente con el servidor. Cuando un servidor Jabber/XMPP acepta una conexión de un cliente responderá con <stream:stream> para confirmar el inicio de la sesión. Si hubiera un error en la sesión se enviaría la etiqueta <stream:error> explicando el problema y cerrando la conexión por parte del servidor.

Una vez que se ha establecido el flujo entre el cliente y el servidor pueden enviarse paquetes de acuerdo con los múltiples protocolos de Jabber/XMPP.

En la mayoría de los casos el servidor sólo le dejará al cliente enviar un número determinado de mensajes hasta que se autentifique, entre ellos como es normal los paquetes de autenticación. La autenticación permite al servidor verificar que el cliente es quien dice ser para poder actuar bajo el JID de quien se autentifica y enviar y recibir mensajes.

Tanto el servidor como el cliente pueden cerrar el flujo de datos en cualquier momento enviando una etiqueta de cierre </stream:stream>. En ese momento, cada uno podrá cerrar la conexión y terminar la sesión Jabber/XMPP. Es conveniente dejar a la otra entidad terminar de enviar un paquete que esté en curso antes de cerrar la conexión.

Conexión típica de Jabber/XMPP:

- Conectarse con el puerto 5222 del servidor.
- Enviar una etiqueta de sesión abierta <session:session>.
- Esperar la respuesta del servidor <session:session>.
- Usar el protocolo de autenticación para iniciar con su Jabber ID.
- Enviar y recibir paquetes de acuerdo a los protocolos Jabber/XMPP.

- Enviar la etiqueta de cierre `</session:session>` para terminar la sesión.
- Cerrar la conexión.

Una característica muy útil del protocolo *XMPP* son las pasarelas, que permiten a los usuarios el acceso a redes con otros protocolos de mensajería instantánea u otros tipos de mensajería como *SMS* o *E-mail* (ver Fig. 1). Este servicio no es proporcionado desde el cliente, sino desde el servidor mediante servicios de pasarela que proporcionan conectividad con alguna otra red. Esto significa que cualquier cliente *XMPP* puede ser usado para acceder a cualquier red en la que haya una pasarela, sin necesidad de adaptar el cliente o de que tenga acceso directo a *Internet*.



Fig. 4.1. Ejemplo de pasarela de XMPP.

No obstante, existía un pequeño problema de conectividad debido a que gran parte de los cortafuegos están configurados para permitir el paso del tráfico *TCP* dirigido al puerto usado por el protocolo *HTTP*, mientras que por lo general se bloquea el puerto utilizado por *XMPP*. Para solucionar este problema, *XMPP* utiliza *HTTP* para permitir el acceso a los usuarios que se encuentran tras cortafuegos.

4.3. Puntos fuertes de XMPP

Descentralización: La arquitectura de las redes *XMPP* es similar a la del correo electrónico; cualquiera puede poner en marcha su propio servidor *XMPP*, sin que haya ningún servidor central.

Estándares abiertos: La *IETF* ha formalizado el protocolo *XMPP* como una tecnología de mensajería instantánea estándar y sus especificaciones han sido publicadas como los RFC 3920 y RFC 3921. El desarrollo de esta tecnología no está ligado a ninguna empresa en concreto y no requiere el pago de *royalties*.

Seguridad: Los servidores *XMPP* pueden estar aislados de la red pública *Jabber* y poseen robustos sistemas de seguridad (como *SASL* y *TLS*). Para apoyar la utilización de los sistemas de cifrado, la *XMPP Standards Foundation* pone a disposición de los administradores de servidores *XMPP* certificados digitales gratis.

Flexibilidad: Se pueden hacer funcionalidades a medida sobre *XMPP*; para mantener la interoperabilidad, las extensiones más comunes son gestionadas por la *XMPP Software Foundation*.

4.4. Puntos débiles de XMPP

Sobrecarga de datos de presencia: Típicamente cerca de un 70% del tráfico entre servidores son datos de presencia y cerca de un 60% de estos son transmisiones redundantes. Actualmente se están estudiando nuevos protocolos para aliviar este problema.

Escalabilidad: *XMPP* también sufre el mismo problema de redundancia en los servicios de salas de chat y de suscripción. Actualmente se está trabajando en su solución.

Sin datos binarios: *XMPP* es codificado como un único y largo documento *XML*, lo que hace imposible entregar datos binarios sin modificar. De todas formas, las transferencias de archivos se han solucionado usando otros protocolos como *HTTP*. Si es inevitable, *XMPP* también puede realizar transferencias codificando todos los datos mediante base64.

4.5. Módulo emisor XMPP Android

El módulo emisor *XMPP Android* hace referencia a la Fig. 3.16. “Esquema de funcionamiento del sistema de Geolocalización” visto anterior anteriormente en el capítulo 3 “Terminal móvil”.

El módulo actual es el encargado de proporcionar a la aplicación la posibilidad de enviar la información usando el protocolo de comunicaciones *XMPP*, es por eso que se han

incorporado una serie de librerías y funciones para dotar al programa del control de las comunicaciones y así poder enviar la información que la clase `dumpLocation` tiene en el Modulo de GPS (su funcionamiento detallado en el capítulo anterior), donde la información que se debe enviar ha sido tratada y acondicionada para su envío.

Las siguientes líneas pertenecen a las librerías necesarias para el envío de la información:

```
import org.android.Xmpp.R;
import org.jivesoftware.smack.ConnectionConfiguration;
import org.jivesoftware.smack.XMPPConnection;
import org.jivesoftware.smack.XMPPEXception;
import org.jivesoftware.smack.*;
import org.jivesoftware.smack.packet.*;
```

La función `initConnection` es la encargada del establecer las comunicaciones con el servidor. Para ello se definirán una serie de parámetro necesarios que son:

Server_Host: El nombre del servidor de XMPP "talk.google.com".

Server_Port: Se trata del puerto que utiliza el servidor para establecer las comunicaciones.

Service_Name: El nombre del servicio "gmail.com".

Login: Nombre con el que se iniciara la sesión `garandse@gmail.com`.

Password: Contraseña necesaria para el inicio de la sesión "HI38Vk85".

```
public void initConnection() throws XMPPEXception {
    //*****Iniciar la Conexion*****
    ConnectionConfiguration config =new ConnectionConfiguration(SERVER_HOST, SERVER_PORT, SERVICE_NAME);
    m_connection = new XMPPConnection(config);
    m_connection.connect();
    m_connection.login(LOGIN, PASSWORD);
}
```

Fig. 4.2.Función `InitConnection`.

La función `enviar` es la encargada una vez se le ha proporcionado un destinatario y un mensaje de enviar la información, haciendo uso de la conexión que está establecida.

```
public void enviar(String to,String text){  
    Message msg = new Message(to, Message.Type.chat);  
    msg.setBody(text);  
    m_connection.sendPacket(msg);  
}
```

Fig. 4.3. Función enviar.

La función desconectar como su nombre indica es la encargada de la desconexión del servidor.

```
public void desconectar(){  
    m_connection.disconnect();  
}
```

Fig. 4.4. Función Desconectar.

4.6. Modulo Receptor Java

XMPP forma parte de las comunicaciones entre el terminal móvil y el servidor, es por ello que mediante Java se ha creado un módulo que realiza la función de mantener conectado el terminal móvil con el servidor, esta aplicación no requiere ningún tipo de interacción con el usuario. Es el encargado del servicio de geolocalización el que tendrá que lanzar esta aplicación una vez tenga en marcha el servidor web, para que la aplicación pueda insertar la información que le aportan los usuarios desde el teléfono móvil.

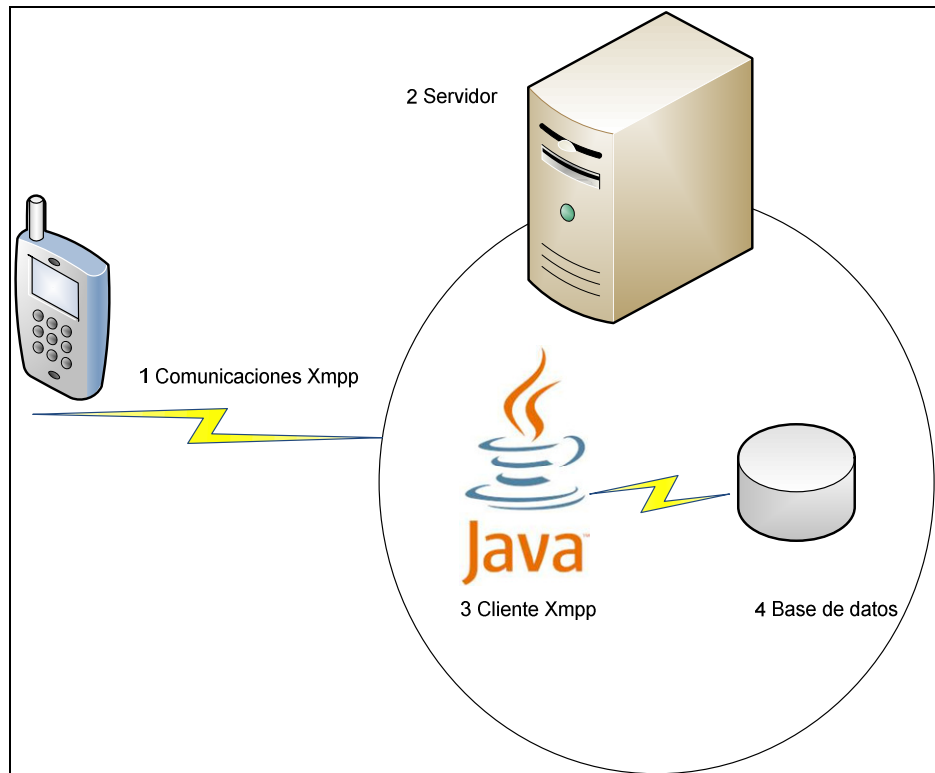


Fig. 4.5. Esquema de funcionamiento de la aplicación Java.

1. **Comunicaciones XMPP:** es el medio de comunicación entre el terminal móvil y la aplicación Java.
2. **Servidor:** Es el equipo que alberga las aplicación java y la base de datos.
3. **Cliente XMPP:** es el encargado de la recepción de las tramas de información enviadas desde el dispositivo móvil.
4. **Base de datos:** es la base de datos que aloja la información de los dispositivos móviles.

El módulo que se ha desarrollado tiene dos funciones principales:

La primera es la recepción de las comunicaciones desde el dispositivo móvil al servidor, para ello haciendo uso de librerías como Smack, se ha implementado una aplicación que recibe los paquetes usando el protocolo XMPP.

Una vez el proceso de recepción recibe los datos, los adapta para que el siguiente proceso pueda realizar su función.

El segundo módulo es el encargado del almacenamiento de la información que le proporciona el módulo anterior o módulo de comunicaciones, en una base de datos donde están definidos los campos de longitud, latitud, imei, address y date. La base de datos estará alojada en el mismo equipo y funcionando con un servidor Apache.

La implementación de la aplicación se puede dividir en tres fases:

La primera la recepción de la información creando un cliente XMPP mediante el uso de las librerías de Smack.

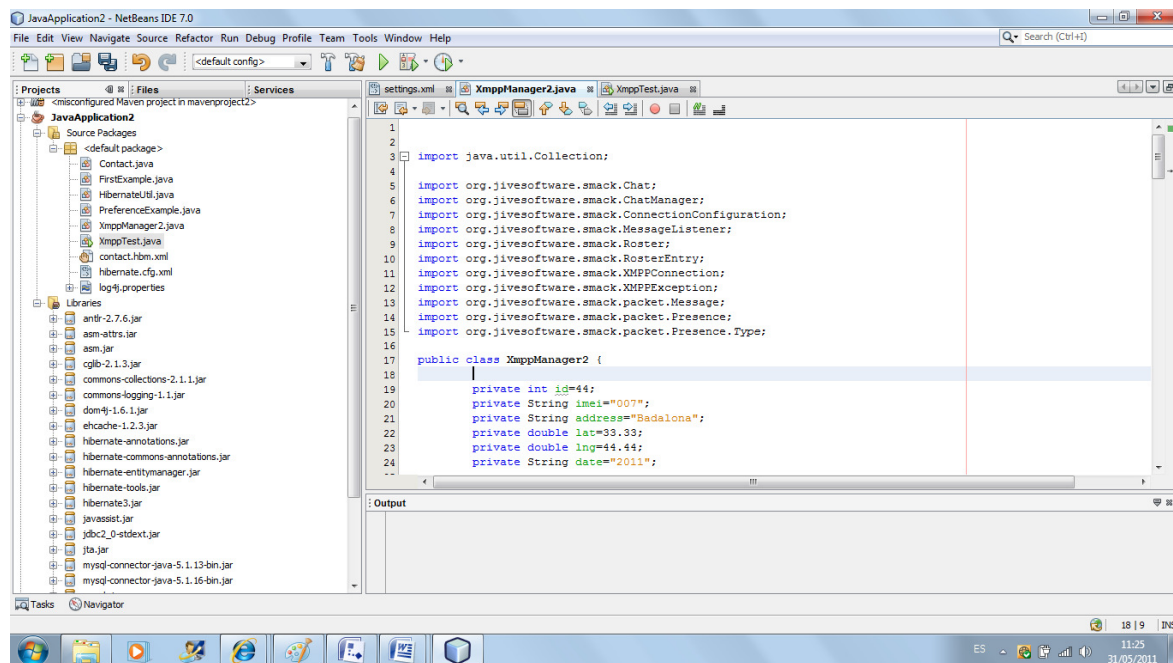


Fig. 4.6. Imagen que muestra de las librerías Smack.

La segunda parte del módulo está implementada con una utilidad que proporciona Java, se trata de Hibernate.

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML).

Existen tres archivos importantes en una aplicación con Hibernate:

- **hibernate.cfg.xml:** es donde se define sobre qué base de datos se trabajara y se configura la conexión con la base de datos.

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD//EN"
4     "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
5 <hibernate-configuration>
6   <session-factory>
7     <!-- Database connection settings -->
8     <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
9     <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/puntos</property>
10    <property name="hibernate.connection.username">root</property>
11    <property name="hibernate.connection.password"></property>
12
13    <!-- JDBC connection pool (use the built-in) -->
14    <property name="hibernate.connection.pool_size">1</property>
15
16    <!-- SQL dialect -->
17    <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
18
19    <!-- Echo all executed SQL to stdout -->
20    <property name="show_sql">True</property>
21
22    <!-- Mapping files -->
23
24    <mapping resource="contact.hbm.xml"/>
25    <!--mapping resource="Person.hbm.xml"/-->
26  </session-factory>
27 </hibernate-configuration>
```

Fig. 4.7. Imagen del archivo hibernate.cfg.xml.

- **contact.hbm.xml:** es donde se definen los elementos del objeto que utilizaremos, se puede apreciar cómo se define el nombre de la clase y el nombre de la tabla donde se guardarán los datos, así como el nombre de las variables utilizadas en el módulo Java y el nombre de los campos en los registros de la base de datos.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name="Contact" table="markers">
    <id name="jid" type="int" column="id" >
      <generator class="increment"/>
    </id>
    <property name="jimei">
      <column name="imei"/>
    </property>
    <property name="jaddress">
      <column name="address"/>
    </property>
    <property name="jlat">
      <column name="lat"/>
    </property>
    <property name="jlng">
      <column name="lng"/>
    </property>
    <property name="jdate">
      <column name="date"/>
    </property>
  </class>
```

Fig. 4.8. Imagen del archivo contact.hbm.xml.

- Contact.java: se trata del archivo donde se definen las funciones que son necesarias para el control de los elementos del objetos, se definen un get y un set por cada elemento, de esta forma se podrá manipular la información de la base de datos, tanto para insertar la información o para leerla.

```
package com.mycompany.pfc2;
public class Contact {
    private int jid;
    private String jimei;
    private String jaddress;
    private double jlat;
    private double jlng;
    private String jdate;

    public Contact() {...}
    /**...*/
    public String getjimei() {...}
    /**...*/
    public String getjaddress() {...}
    /**...*/
    public double getjlat() {...}
    /**...*/
    public double getjlng() {...}
    /**...*/
    public String getjdate() {
        return jdate;
    }
    /**...*/
    public void setjimei(String string) {
        jimei = string;
        System.out.println(jimei);
    }
}
```

Fig. 4.9. Imagen del archivo Contact.java

La tercera fase de la implementación del módulo Java es la creación de un paquete .jar, para ello se ha utilizado otra de las herramientas que proporciona NetBeans 7.0, se trata de Maven una herramienta de software para la gestión y construcción de proyectos Java.

Maven es una herramienta que crea un ejecutable .jar para ello es necesario que se le indique que librerías son necesarias para el correcto funcionamiento de la aplicación, así como la clase main. Estos puntos son definidos en el archivo pom.xml.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany</groupId>
  <artifactId>PFC2</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>PFC2</name>
  <url>http://maven.apache.org</url>
  <!--> Desde aqui<!-->
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <pluginRepositories>
    <pluginRepository>
      <id>freehep-maven</id>
      <name>Maven FreeHEP</name>
      <url>http://java.freehep.org/maven2</url>
    </pluginRepository>
  </pluginRepositories>

  <build>
    <plugins>
      <plugin>
        <groupId> org.apache.maven.plugins</groupId>
```

Fig. 4.10. Imagen del archivo pom.xml

5. APACHE

5.1. ¿Qué es?

Es un servidor web de distribución libre y de código abierto, siendo el más popular del mundo desde 1995. Acapara el 60% de la cuota de mercado.

Apache es mantenido y desarrollado por una comunidad abierta de desarrolladores bajo la Apache Software Foundation que están ubicados por todo el mundo. La Aplicación permite ejecutarse en múltiples sistemas operativos tanto Windows, Mac OS y Linux.

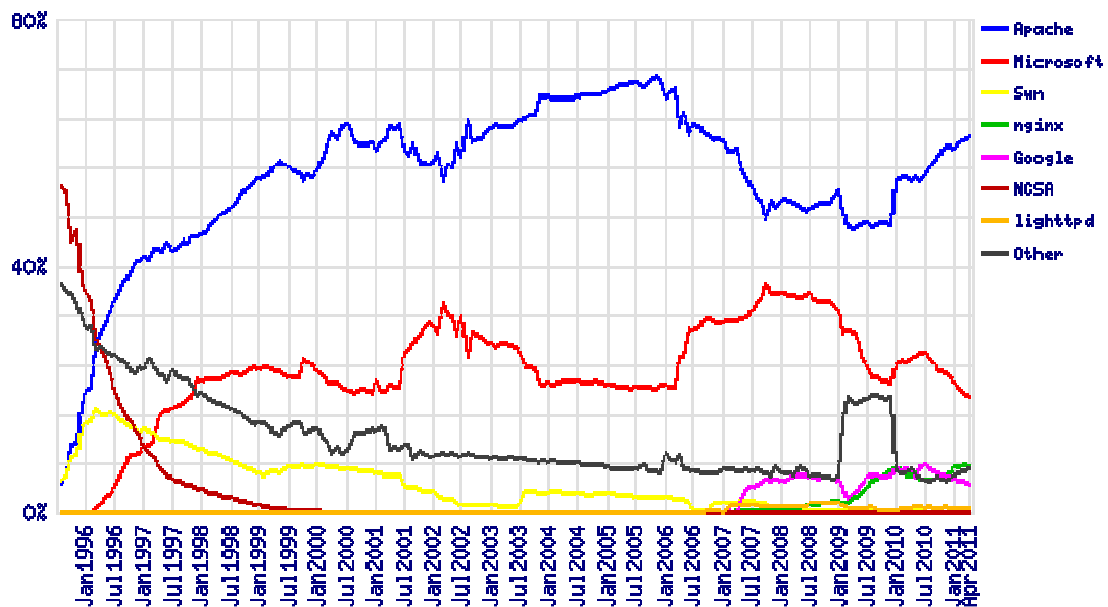


Fig. 5.1. Grafica de evolución de los diferentes servidores.

5.2. Características principales

Entre las características más importantes destacamos las siguientes:

- Páginas protegidas con contraseña para múltiples usuarios.
- Páginas de errores personalizados.
- Soporta los lenguajes Java, Jsp, Perl, Python y PHP.

- Es multiplataforma.
- La tecnología que usa es gratuita.
- Soporta SSL y TLS

5.3. Funcionalidades

El uso principal es el alojamiento de páginas web y/o contenido en un equipo externo accesible mediante internet. Se utiliza también junto a MySQL⁵ y MyPHP⁶, lo que nos proporciona una gran versatilidad.

Apache puede ser usado para otras tareas donde el contenido necesita ser puesto a disposición de los usuarios de una forma fiable y segura. Una de las utilidades que se le da es la de compartir archivos desde un ordenador hacia internet, también se puede utilizar como servidor local para realizar todas las pruebas que se necesitan mientras se está desarrollando.

Curiosidades:

El servidor de Microsoft es Internet Information Server que es el principal competidor de Apache, seguido de Sun (como se puede observar en la gráfica anterior). Las páginas web más importantes están siendo ejecutadas por Apache.

La capa frontal (front end) del motor de búsqueda de Google está basada en Apache, denominada Google Web Server (GWS).

5.4. Caso práctico

En este caso el uso del servidor Apache da la posibilidad de albergar la página web, que proporciona una serie de funcionalidades como son: registrar los usuarios, administrar la bases de datos, ver la última posición de un usuario y ver todos los puntos registrados. Tanto el último punto como todos los registros que se disponen son mostrados sobre un

⁵ MySQL: Producto de código abierto para la gestión de bases de datos relacional.

⁶ MyPHP: lenguaje de programación que permite crear webs dinámicas.

mapa, para la utilización del mapa se ha recurrido a la implementación de la Api de Google Maps, y la implementación de paginas PHP, que proporcionan dinamismo e interacción con el usuario.



Fig. 5.2. Web del Sistema de geolocalización.

Otra de las funcionalidades que ofrece Apache y que se ha hecho servir es albergar las bases de datos necesaria tanto para el registro y control de los usuarios, como para la base de datos que guarda los movimientos del usuario.

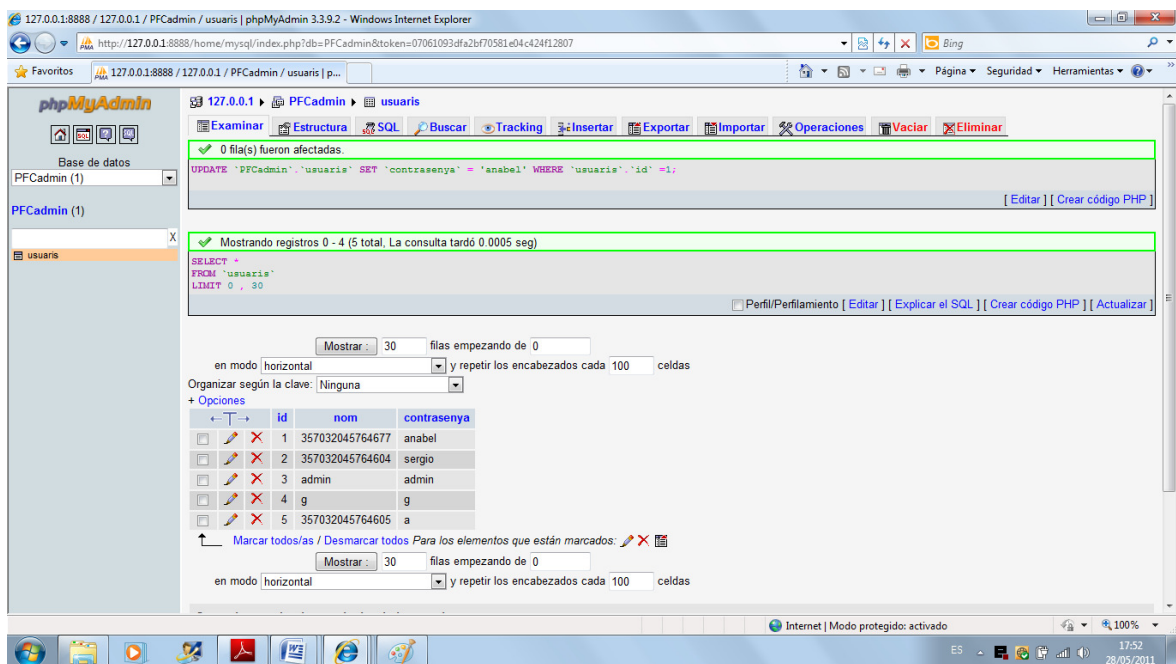
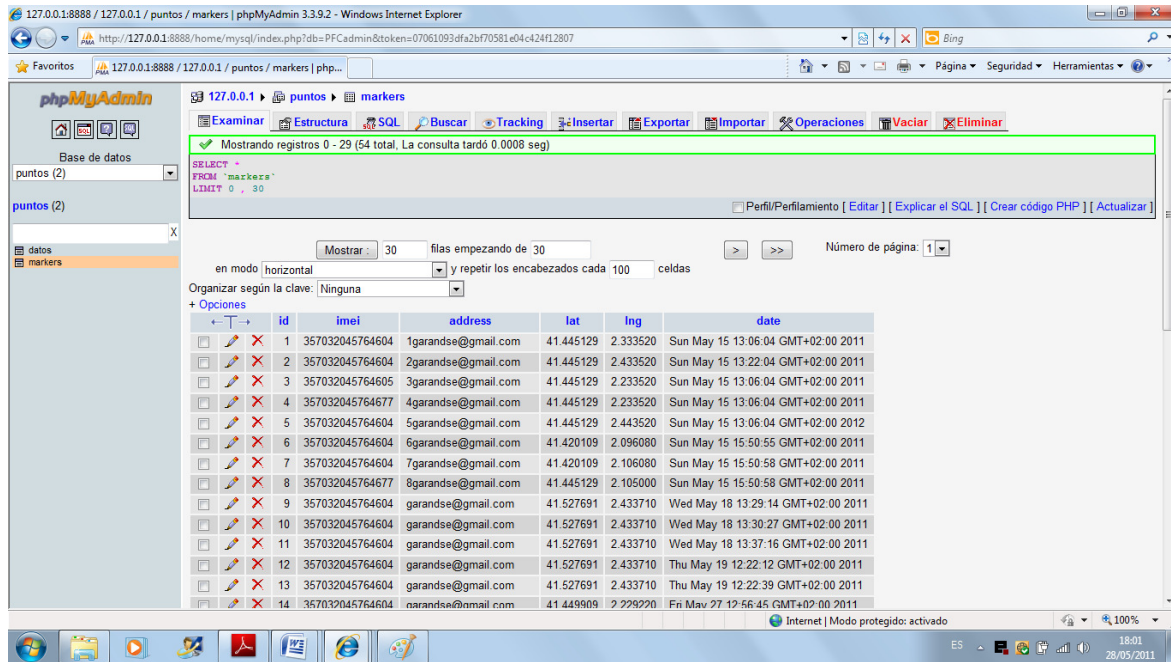


Fig. 5.3. Bases de datos de usuarios.



The screenshot shows the phpMyAdmin interface for a database named 'puntos'. The 'markers' table is selected, and a query is executed: `SELECT * FROM 'markers' LIMIT 0, 30`. The table displays 14 records of user location data.

	id	imei	address	lat	lng	date
<input type="checkbox"/>	1	357032045764604	1garandse@gmail.com	41.445129	2.333520	Sun May 15 13:06:04 GMT+02:00 2011
<input type="checkbox"/>	2	357032045764604	2garandse@gmail.com	41.445129	2.433520	Sun May 15 13:22:04 GMT+02:00 2011
<input type="checkbox"/>	3	357032045764605	3garandse@gmail.com	41.445129	2.233520	Sun May 15 13:06:04 GMT+02:00 2011
<input type="checkbox"/>	4	357032045764677	4garandse@gmail.com	41.445129	2.233520	Sun May 15 13:06:04 GMT+02:00 2011
<input type="checkbox"/>	5	357032045764604	5garandse@gmail.com	41.445129	2.443520	Sun May 15 13:06:04 GMT+02:00 2012
<input type="checkbox"/>	6	357032045764604	6garandse@gmail.com	41.420109	2.096080	Sun May 15 15:50:55 GMT+02:00 2011
<input type="checkbox"/>	7	357032045764604	7garandse@gmail.com	41.420109	2.106080	Sun May 15 15:50:58 GMT+02:00 2011
<input type="checkbox"/>	8	357032045764677	8garandse@gmail.com	41.445129	2.105000	Sun May 15 15:50:58 GMT+02:00 2011
<input type="checkbox"/>	9	357032045764604	garandse@gmail.com	41.527691	2.433710	Wed May 18 13:29:14 GMT+02:00 2011
<input type="checkbox"/>	10	357032045764604	garandse@gmail.com	41.527691	2.433710	Wed May 18 13:30:27 GMT+02:00 2011
<input type="checkbox"/>	11	357032045764604	garandse@gmail.com	41.527691	2.433710	Wed May 18 13:37:16 GMT+02:00 2011
<input type="checkbox"/>	12	357032045764604	garandse@gmail.com	41.527691	2.433710	Thu May 19 12:22:12 GMT+02:00 2011
<input type="checkbox"/>	13	357032045764604	garandse@gmail.com	41.527691	2.433710	Thu May 19 12:22:39 GMT+02:00 2011
<input type="checkbox"/>	14	357032045764604	narandse@nmail.com	41.449909	2.229220	Fri May 27 12:56:45 GMT+02:00 2011

Fig. 5.3. Bases de datos para el registro de la posición de los usuarios.

6. Api de Google Maps

6.1. ¿Qué es Api de Google Maps?

Una API (Application Programming Interface) consiste en un conjunto de librerías de clases que permiten ser usadas de forma fácil y rápida. De esta manera se pueden utilizar y manejar muchos elementos en las aplicaciones que ya han sido programados sin tener que empezar desde cero.

6.2.Utilidades

La API de Google Maps es un complemento de Google que permite insertar y usar los mapas de Google en las páginas web. Se trata de un servicio gratuito dónde únicamente se debe estar registrado para poderlo usar las aplicaciones. Al registrarse se obtiene una clave, que se requerirá en la parte del código de la aplicación. Unas de las características más importantes que nos ofrece es poder controlar los mapas y manejarlos a gusto del usuario. Por ejemplo, se puede cambiar el zoom para alejar o acercar un punto en concreto, indicar unas coordenadas específicas y elegir el tipo de vista a mostrar (satélite o calles).

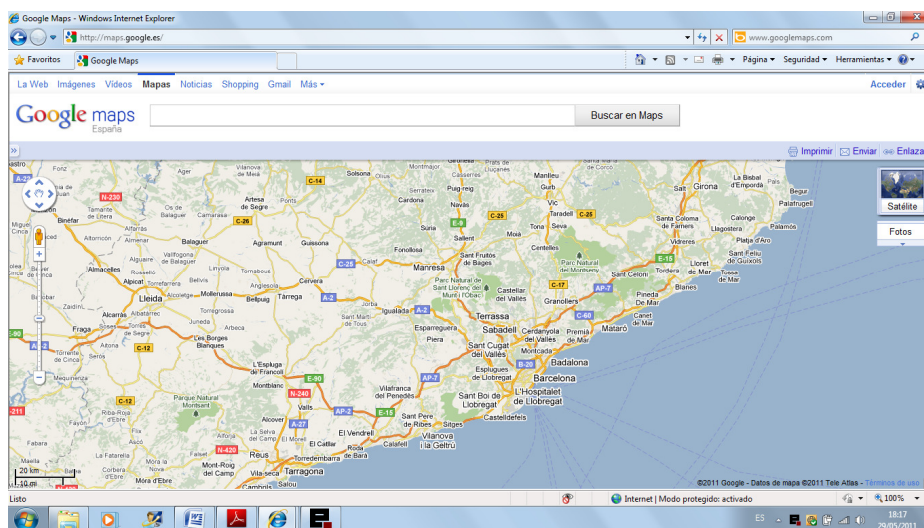


Fig. 6.1. Ejemplo de Googlemaps.

6.3 Obtención de API Key

Para poder usar Google Maps es necesario registrarse en la siguiente página: <http://code.google.com/intl/es-ES/apis/maps/signup.html> y aceptar los términos del servicio.

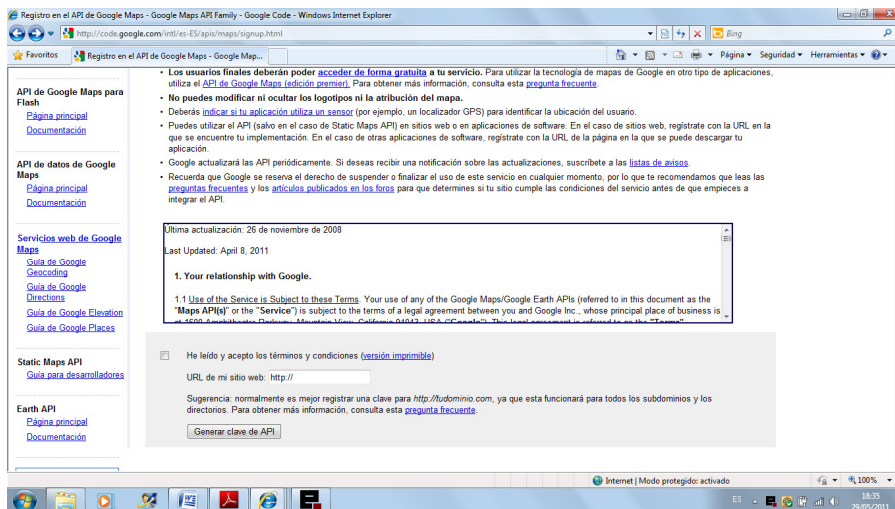


Fig. 6.2. Página para generar la Key de Google Maps.

Una vez aceptados los términos y pulsado el botón de generar clave de API, se nos mostrará una serie de claves dependiendo del proyecto que vayamos a generar.

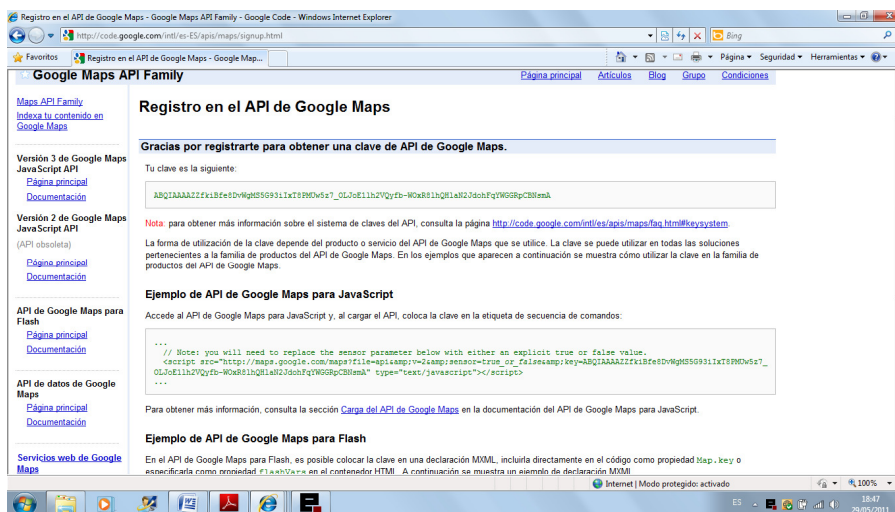


Fig. 6.3. Página de registro de Google Maps.

Una vez obtenida la clave, ésta se deberá insertar en el código de la aplicación para que cada vez que se utilicen los mapas el servidor de Google pueda ver que está registrado con el servicio de Google Maps”.

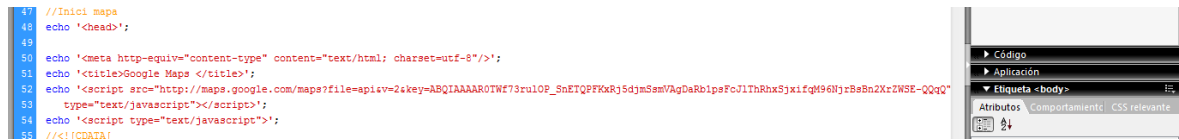


Fig. 6.4. Parte del código donde está la clave de Google Maps.

En el caso de que no se disponga de una clave o ésta no sea válida, la aplicación seguiría funcionando, simplemente no se mostrarían los mapas.

6.4. Aplicación web

En este caso la Api de Google Maps ha sido una herramienta más para la creación de la página web de este proyecto.

La web es una aplicación desarrollada con php en la que el usuario tendrá diversas funciones para poder interactuar: ver la última posición en la que se encuentra el usuario y disponer de la posibilidad de ver el recorrido que ha efectuado el usuario que mediante su dispositivo móvil ha estado enviado su posición.

La web también dispone de unas funciones destinadas a la gestión de usuarios que puedan usar la aplicación, así como gestionar la base de datos donde cada usuario envía su posición.

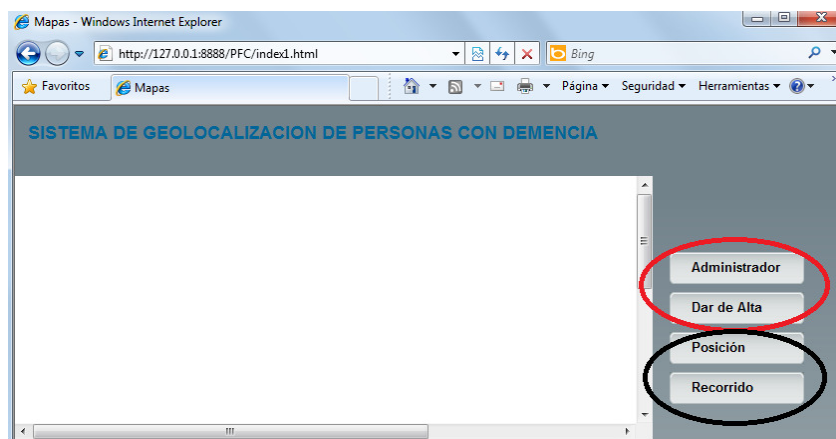


Fig. 6.5. Pantalla principal de la aplicación web, partes de menú.

6.4.1. Administrador

En el apartado de administrador después de introducir un usuario y un password valido se encuentra una serie de utilidades para la gestión de la base de datos.

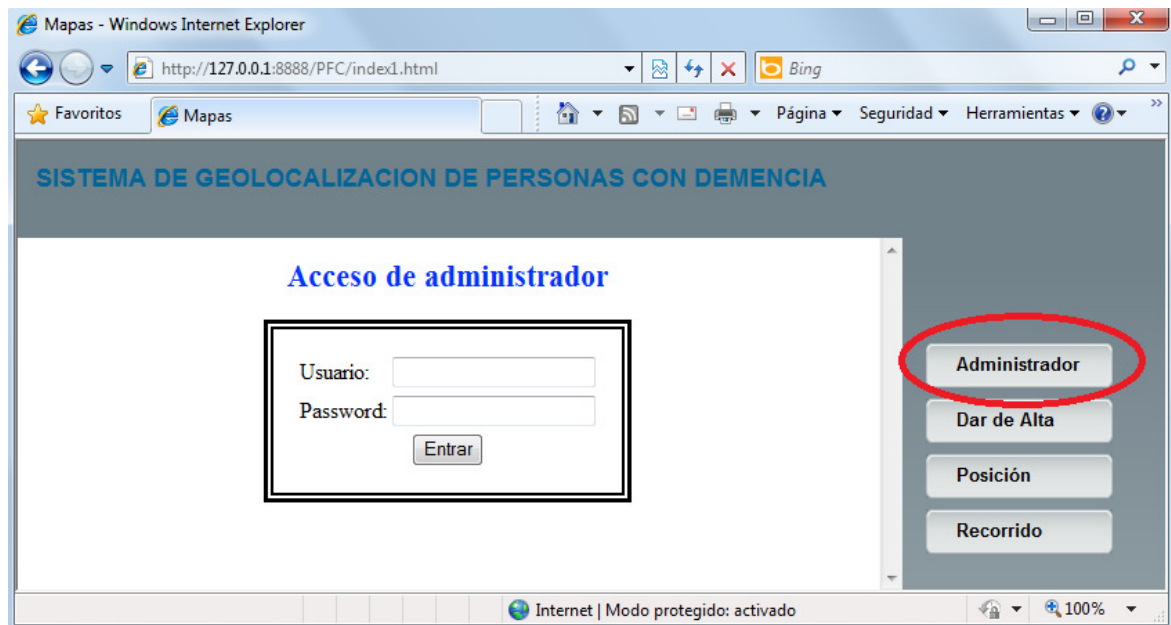


Fig. 6.6. Pantalla de la validación como administrador web.

Las operaciones que el administrador puede hacer sobre la base de datos son aquellas que le permiten una gestión del día a día:

- Insertar un registro
- Modificar un registro
- Borrar un registro.

Desde esta pantalla el administrador podrá modificar si es necesario algún registro, eliminar los registros de aquellos usuarios del sistema que se hayan dados de baja, también podrá insertar algún registro aunque no sea el funcionamiento normal de inserción de datos, ya que las información que se almacenan son los provenientes del envío de las coordenadas desde los terminales móviles.

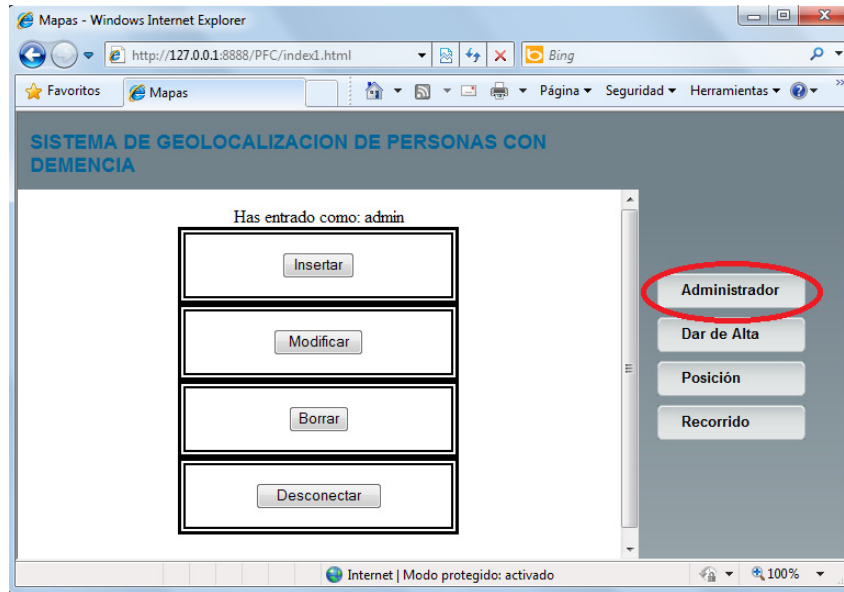


Fig. 6.7. Pantalla con las funciones del administrador.



Fig. 6.8. Ejemplo de borrado de un registro.

6.4.2. Dar de alta

La función dar de alta incorpora nuevos usuario al servicio de localización. Los usuarios deberán introducir como nombre el identificador único de su teléfono (Imei), de esta manera cuando el usuario desee localizar la posición del dispositivo estará identificado por un elemento único. También deberá introducir una contraseña para poder acceder a las funciones propias del usuario, como serán ver la última posición donde está el dispositivo móvil o ver todo el recorrido.

La información es almacenada en una base de datos de usuarios que se puede ver en la figura 5.3

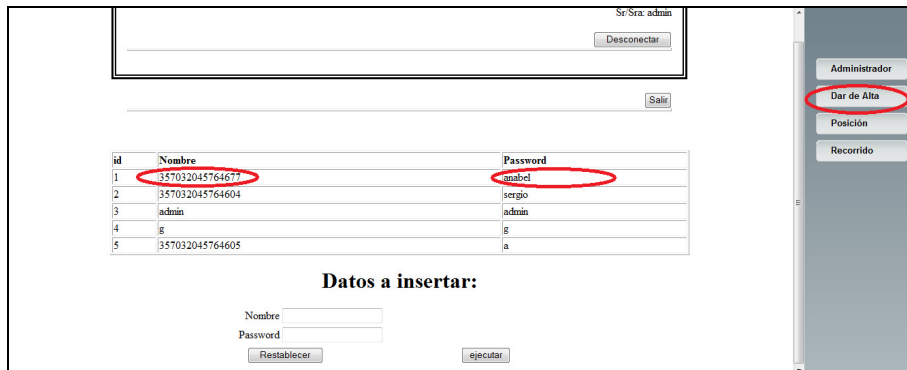


Fig. 6.9. Ejemplo de borrado de un registro.

6.4.3. Posición

La posición es una función que está destinada propiamente a los usuarios del servicio de geolocalización, el usuario que habiendo sido registrado en el apartado anterior, al introducir el imei del teléfono que desea localizar y la contraseña, accederá a ver la posición del dispositivo móvil sobre un mapa que proporciona la Api de Google Maps.

La posición estará marcada por un globo o punto de color rojo que al pinchar sobre éste nos mostrara una serie de datos relacionados con el usuario del sistema, el identificador imei, la fecha y la hora del envío de la posición y la dirección del cliente XMPP que se ha empleado para el envío de la coordenadas (longitud y latitud).

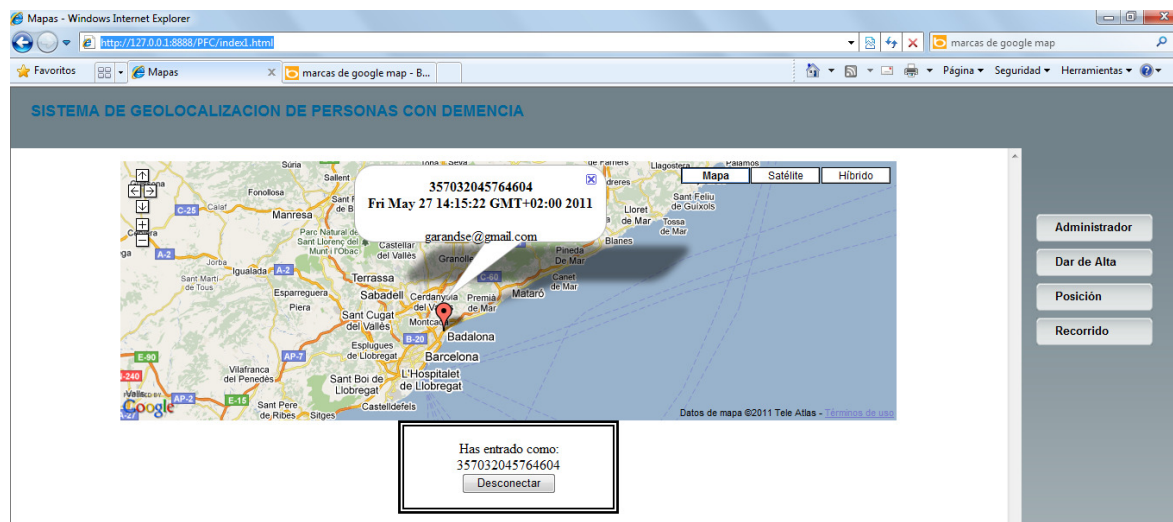


Fig. 6.10. Pantalla donde se aprecia la última posición del dispositivo.

6.4.4. Recorrido

El recorrido al igual que el caso anterior se trata de un servicio que proporciona la web de geolocalización, dicho servicio se muestra tras haberse identificado con el imei y la contraseña. El recorrido que muestra es el que previamente ha sido almacenado en la base de datos que podemos ver en la figura 5.3. Dicha base de datos esta implementada con Mysql y funcionando con un servidor Apache.



Fig. 6.11. Pantalla donde se puede visualizar el recorrido del dispositivo.

7. Conclusiones.

El objetivo del trabajo era el diseñar y construir un sistema que facilitara el día a día a las personas con algún tipo de demencia, para ello se trataba de diseñar un software que incorporado al Smartphone del usuario permitirá ser localizado por sus familiares desde casa a través de una web.

Los puntos que se habían detallado al inicio del proyecto, que eran la creación de una aplicación para una plataforma Android y una web para la visualización de la posición del usuario han sido realizados con éxito.

En este proyecto ha proporcionado una amplia diversidad de puntos a tocar.

El primero ha sido la de aprender a desarrollar aplicaciones para un Smartphone, más concretamente para la plataforma o sistema operativo Android, cabe destacar que la amplia difusión de la información que existe sobre Android, ha facilitado mucho el trabajo.

La aplicación Android y el módulo receptor XMPP nos ha permitido introducirnos en el campo de las comunicaciones, aprovechando las conexiones de datos que nos proporcionan las operadoras de telefonía.

La creación del módulo receptor XMPP también ha permitido la posibilidad de experimentar con utilidades como hibernate, que nos ha permitido la interacción entre Java y las bases de datos. También ha sido muy interesante la utilización de Maven, se trata de otra utilidad que nos proporciona la posibilidad de crear un ejecutable para el módulo receptor XMPP .

La aplicación web ha proporciona la posibilidad de trabajar con un servidor local como el EasyPhp, que incorpora las posibilidades de gestionar bases de datos Mysql y un servidor capaz de albergar páginas PHP.

Para realizar la web se ha tenido que realizar una programación con páginas activas, es por ellos que se decidió usar páginas PHP junto con la Api de Google Maps que nos proporciona la posibilidad de utilizar un mapa con todas sus funciones.

8. Mejoras futuras.

A causa de la limitación del tiempo para la realización del proyecto, hay diversas mejoras o ampliaciones que podrían realizarse en la aplicación.

En la aplicación de Android algunas de las mejoras serían:

La incorporación de un menú donde poder guardar el nombre de usuario y password para poder configurar los parámetros necesarios para la comunicación XMPP. Para hacerlo más versátil también podría añadir los datos del servidor, de esta manera la aplicación no está limitada a un solo servidor XMPP.

También se podría incorporar una base de datos, que en caso de no poder enviar los datos por algún problema en las comunicaciones, el sistema almacenara la información durante ese tiempo.

Para mejorar el ahorro de energía del Smartphone se podría programar un alarmservice, se trata de hacer que la aplicación sea un proceso programado lo más parecido a una alarma, que se ejecutaría por ejemplo cada 20 minutos y durante el resto del tiempo dejaría el procesador del teléfono libre para realizar otras tareas.

La aplicación web que proporciona la ubicación se le podría incorporar nuevas funciones como pueden ser ver el recorrido de un usuario en un periodo de tiempo acotado.

Para ampliar la cobertura a otros usuarios se podría desarrollar aplicaciones similares para otros sistemas operativos como puede ser: iOS, Blackberry o Windows Phone.

9. Referencias

- [1] Web oficial de Google latitude, <http://www.google.es/latitude>
Fecha de última consulta: 20-2-2011
- [2] Página de localizatodo, <http://www.localizatodo.com/Moviles/intro.htm>
Fecha de última consulta: 20-2-2011
- [3] Página de asociación de Alzheimer, <http://www.afal.es/AFAL/index.php>
Fecha de última consulta: 21-2-2011
- [4] Página de estadísticas, <http://news.netcraft.com/archives/2011/>
Fecha de última consulta: 24-2-2011
- [5] Libro, Anaya multimedia: Fundamentos Desarrollo web con PHP, Apache y MySQL.
- [6] Página oficial de apache, http://httpd.apache.org/ABOUT_APACHE.html
- [7] Página oficial de keruve, www.keruve.com Fecha de última consulta: 25-2-2011
- [8] Página dedicada a la Historia del GPS
<http://homepages.mty.itesm.mx/al584299/mypaper.htm>
Fecha de última consulta: 26-2-2011
- [9] Página dedicada al GPS,
http://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global
Fecha de última consulta: 26-2-2011
- [10] Página dedicada a la telefonía móvil,
http://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil

Fecha de última consulta: 10-3-2011

[11] Libro, Android, Editorial :Anaya multimedia Escritor: Ed Burnette

[12] Web oficial programadores Android de Google

<http://developer.android.com/index.html>

Fecha de última consulta: 15-4-2011

[13] Foro especializado en Android

<http://groups.google.com/group/desarrolladores-android/web>

Fecha de última consulta: 28-4-2011

[14] Página de hibernate, <http://www.roseindia.net/hibernate/>

Fecha de última consulta: 3-5-2011

[15] Página de Maven, <http://maven.apache.org/>

Fecha de última consulta: 12-5-2011

10. Anexos

10.1. Presupuesto

A continuación se realiza un desglose del coste total de la realización del proyecto.

En él se incluyen los estudios previos necesarios para llevar a cabo el desarrollo del proyecto.

Para la realización del proyecto se ha tenido en cuenta los siguientes puntos:

El coste de una hora de un Ingeniero de Telecomunicaciones es de 40€, aquí se incluyen gastos que puedan desprenderse de del proyecto como pueden ser, gastos de teléfono, impresiones o electricidad.

Todos los precios especificados a continuación no incorporan IVA.

El número total de horas de dedicación ha sido aproximado un promedio de 4,5 horas diarias lo que nos proporciona un total de 405 horas, de las cuales se desglosan de la siguiente manera:

Desglose de actividades	
Estudio previo	50 horas
Programación	
Módulo Android	100 horas
Módulo Java	100 horas
Aplicación web	100 horas
Documentación	55 horas

Tabla 10.1: Desglose de actividades.

10.1.1. Coste de personal

Coste de actividades				
	Personas	Precio/hora	Horas Totales	Total
Estudio previo	1	40 €	50 horas	2.000 €
Programación				
Módulo Android	1	40 €	100 horas	4.000 €
Módulo Java	1	40 €	100 horas	4.000 €
Aplicación web	1	40 €	100 horas	4.000 €
Documentación	1	40 €	55 horas	2.200 €
			Total	16.200 €

Tabla 10.2: Coste de actividades.

10.1.2. Coste material informático

Coste de material informático	
	Coste
1 PC intel Core i5-2410M HD 640Gb 4 Gb RAM	559 €
Mobil HTC Wilfire	199 €
Netbeans 7.0	0 €
Eclipse	0 €
Adobe Dreamweaver CS5.5	199 €
EasyPHP 5.3.6.0	0 €
Navegadores (Mozilla, Internet explorer)	0 €
Herramientas ofimáticas	699 €
Windows 7 Home Premium 64-bit ⁷	0 €
Total	1.656 €

Tabla 10.3: Coste de material informático.

⁷ El coste del sistema operativo está incluido en el precio del PC

10.1.3. Coste total del proyecto

Coste total	
	Coste
Coste de personal	16.200 €
Coste de material informático	1.656 €
Total	17.856 €

Tabla 10.4: Coste total.

