

**UNIVERSIDAD CARLOS III DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**

**INGENIERÍA DE TELECOMUNICACIÓN**



**PROYECTO FIN DE CARRERA**

**Desarrollo de una Aplicación Web de Gestión de Flotas  
basada en la plataforma de localización de un operador  
de telefonía móvil**

**AUTOR: GUILLERMO PELAZ MARTÍN  
TUTOR: VICENTE PALACIOS MADRID**

**Diciembre 2010**



*A toda la gente que me ha apoyado y ayudado en el desarrollo de este proyecto*

# Resumen

Con el despliegue y evolución de las redes de comunicaciones móviles se abre un universo de nuevos servicios y entre ellos destacan aquellos basados en la localización del usuario (Location Based Services – LBS). Este trabajo trata de desarrollar una solución que permite la gestión de la flota de una empresa de una manera sencilla e intuitiva, basada en la facilidades que ofrecen actualmente plataformas comerciales de localización de algunos operadores de telefonía móvil.

# Índice general

|  |           |
|--|-----------|
| <b>CAPÍTULO 1: INTRODUCCIÓN .....</b>                                  | <b>1</b>  |
| 1.1. <i>Objetivos</i> .....  | 3         |
| 1.2. <i>Estructura del proyecto</i> .....                              | 4         |
| <b>CAPÍTULO 2: ESTADO DEL ARTE.....</b>                                | <b>5</b>  |
| 2.1. <i>Servicios Basados en Localización</i> .....                    | 5         |
| 2.1.1. <i>Introducción</i> .....                                       | 5         |
| 2.1.2. <i>Tecnologías de Localización</i> .....                        | 6         |
| 2.1.3. <i>La cadena del valor en la localización</i> .....             | 17        |
| 2.1.4. <i>Tipología de servicios basados en localización</i> .....     | 20        |
| 2.1.5. <i>Ejemplos de servicios basados en localización</i> .....      | 22        |
| 2.2. <i>World Wide Web</i> .....                                       | 26        |
| 2.2.1. <i>La usabilidad en la Web</i> .....                            | 27        |
| 2.2.2. <i>Elementos clave de sitios Web centrados en usuario</i> ..... | 29        |
| 2.3. <i>Desarrollo de Software</i> .....                               | 30        |
| 2.3.1. <i>Proceso de desarrollo de software</i> .....                  | 30        |
| 2.3.2. <i>Ciclo de vida del Software</i> .....                         | 33        |
| 2.3.3. <i>UML</i> .....  | 38        |
| <b>CAPÍTULO 3: HERRAMIENTAS PARA LA ELABORACIÓN DEL PROYECTO .....</b> | <b>45</b> |
| 3.1. <i>Infraestructura de servidor</i> .....                          | 45        |
| 3.1.1. <i>Servidor Web</i> .....                                       | 45        |
| 3.1.2. <i>Sistema Gestor de Bases de Datos</i> .....                   | 46        |
| 3.1.3. <i>Páginas web dinámicas</i> .....                              | 50        |
| 3.2. <i>Entorno de trabajo</i> .....                                   | 54        |
| 3.2.1. <i>Eclipse</i> .....  | 54        |
| 3.3. <i>Librerías</i> .....  | 56        |
| 3.3.1. <i>Smarty</i> .....   | 56        |
| <b>CAPÍTULO 4: DESARROLLO DEL PROYECTO .....</b>                       | <b>58</b> |
| 4.1. <i>Fase inicial</i> .....   | 58        |
| 4.1.1. <i>Proceso de desarrollo</i> .....                              | 59        |

|  |            |
|--|------------|
| 4.1.2. Especificación de requisitos de Usuario.....                | 60         |
| 4.1.3. Diagrama de casos de uso inicial .....                      | 61         |
| 4.2. Fase de Análisis.....   | 62         |
| 4.2.1. Especificación de requisitos en la Fase de Análisis.....    | 62         |
| 4.2.2. Diagrama de casos de uso en la Fase de Análisis.....        | 67         |
| 4.2.3. Modelo de Base de Datos .....                               | 70         |
| 4.3. Estudio de la API de localización.....                        | 72         |
| 4.3.1. Alta de usuarios .....                                      | 73         |
| 4.3.2. Operaciones de provisión .....                              | 73         |
| 4.3.3. Invocando a la API de localización .....                    | 77         |
| 4.4. Estudio de la API de cartografía .....                        | 85         |
| 4.5. Diseño arquitectónico.....                                    | 91         |
| 4.5.1. Infraestructura Hardware .....                              | 91         |
| 4.5.2. Infraestructura Software.....                               | 92         |
| 4.6. Diseño detallado .....  | 92         |
| 4.6.1. Implementación y pruebas de la Fase Final del Sistema ..... | 93         |
| 4.6.2. Consideraciones importantes sobre el desarrollo .....       | 102        |
| 4.7. Resumen del proyecto.....                                     | 106        |
| 4.7.1. Planificación Final .....                                   | 107        |
| 4.7.2. Resumen de costes .....                                     | 107        |
| 4.7.3. Estado del proyecto .....                                   | 110        |
| <b>CAPÍTULO 5: CONCLUSIONES.....</b>                               | <b>111</b> |
| 5.1. Futuras líneas de desarrollo.....                             | 112        |
| <b>CAPÍTULO 6: REFERENCIAS .....</b>                               | <b>113</b> |
| <b>CAPÍTULO 7: ACRÓNIMOS .....</b>                                 | <b>115</b> |

# Lista de Figuras

|   |    |
|---|----|
| Figura 2.1. Zonas de ubicación en función de tipo de la antena .....      | 8  |
| Figura 2.2. Localización en base al TOA.....                              | 9  |
| Figura 2.3. Localización basada en E-OTD.....                             | 11 |
| Figura 2.4. Cobertura satelital en GPS.....                               | 12 |
| Figura 2.5. A-GPS .....   | 14 |
| Figura 2.6. Captura Google Maps Mobile .....                              | 15 |
| Figura 2.7. Ejemplo Wifi-Mapping.....                                     | 16 |
| Figura 2.8. Cadena de valor en la localización .....                      | 18 |
| Figura 2.9. Logo de Ivesfleet .....                                       | 23 |
| Figura 2.10. Ejemplo de dispositivo embarcado.....                        | 24 |
| Figura 2.11. Captura de FLOTASNET .....                                   | 26 |
| Figura 2.12. Representación de un proceso de desarrollo de software ..... | 31 |
| Figura 2.13. Ciclo de vida en cascada .....                               | 37 |
| Figura 2.14. Ciclo de vida iterativo-incremental.....                     | 38 |
| Figura 2.15. Ejemplo de diagrama de casos de uso .....                    | 41 |
| Figura 2.16. Figura de un paquete.....                                    | 41 |
| Figura 2.17. Figura de una clase .....                                    | 42 |
| Figura 2.18. Figura de una interfaz.....                                  | 42 |
| Figura 2.19. Ejemplo de un diagrama de clases.....                        | 43 |
| Figura 2.20. Ejemplo de diagrama de secuencia.....                        | 44 |
| Figura 3.1. Logo del servidor web Apache.....                             | 46 |
| Figura 3.2. Comparativa de servidores Web fuente (NETCRAFT,2010) .....    | 46 |
| Figura 3.3. Logo de MySQL .....   | 50 |
| Figura 3.4. Ejemplo generación dinámica de páginas html.....              | 51 |
| Figura 3.5. Logo de PHP .....   | 51 |
| Figura 3.6. Ejemplo código fuente PHP .....                               | 52 |
| Figura 3.7. Captura del IDE Eclipse PDT .....                             | 55 |
| Figura 3.8. Logo de Smarty.....   | 56 |
| Figura 3.9. Ejemplo de código utilizando Smarty.....                      | 57 |
| Figura 4.1. Casos de uso iniciales .....                                  | 61 |
| Figura 4.2. Casos de uso en la Fase de Análisis.....                      | 67 |
| Figura 4.3. Modelo de la Base de Datos .....                              | 71 |
| Figura 4.4. Ejemplo de estructura de la provisión en la GPP.....          | 72 |
| Figura 4.5. Representación la localización en Google Maps .....           | 84 |
| Figura 4.6. Detalles de la localización.....                              | 84 |

|  |     |
|--|-----|
| Figura 4.7. Captura de Google Earth .....                                | 86  |
| Figura 4.8. Captura de Google Maps .....                                 | 87  |
| Figura 4.9. Ejemplo API de Google Maps.....                              | 90  |
| Figura 4.10. Arquitectura del servicio .....                             | 91  |
| Figura 4.11. Página principal de la aplicación .....                     | 93  |
| Figura 4.12. Pantalla de Inicio para un perfil de administrador .....    | 94  |
| Figura 4.13. Pantalla de Gestión de la GPP.....                          | 95  |
| Figura 4.14. Pantalla de Gestión de Elementos .....                      | 96  |
| Figura 4.15. Pantalla de Gestión de usuarios.....                        | 96  |
| Figura 4.16. Pantalla para “Ver Registros” .....                         | 97  |
| Figura 4.17. Pantalla de Inicio para un perfil de usuario.....           | 98  |
| Figura 4.18. Pantalla de Gestión de mi Flota .....                       | 98  |
| Figura 4.19. Pantalla de Envío de SMS .....                              | 99  |
| Figura 4.20. Pantalla Localizar.....                                     | 99  |
| Figura 4.21. Pantalla con resultado de localización .....                | 100 |
| Figura 4.22. Pantalla para Ver Registros por usuario.....                | 101 |
| Figura 4.23. Pantalla Mis datos .....                                    | 101 |
| Figura 4.24. Planificación final del proyecto en Diagrama de Gantt ..... | 107 |

# Lista de Tablas

|  |     |
|--|-----|
| Tabla 2.1. Comparativa mecanismos de localización..... | 17  |
| Tabla 2.2. Fases de desarrollo del proyecto .....      | 34  |
| Tabla 4.1. RU-000001.....                              | 63  |
| Tabla 4.2. RU-000002.....                              | 63  |
| Tabla 4.3. RU-000003.....                              | 63  |
| Tabla 4.4. RU-000004.....                              | 64  |
| Tabla 4.5. RU-000005.....                              | 64  |
| Tabla 4.6. RU-000006.....                              | 64  |
| Tabla 4.7. RU-000007.....                              | 65  |
| Tabla 4.8. RU-000008.....                              | 65  |
| Tabla 4.9. RU-000009.....                              | 65  |
| Tabla 4.10. RU-000010.....                             | 66  |
| Tabla 4.11. RU-000011.....                             | 66  |
| Tabla 4.12. RU-000012.....                             | 66  |
| Tabla 4.13. CU-001.....                                | 68  |
| Tabla 4.14. CU-002.....                                | 68  |
| Tabla 4.15. CU-003.....                                | 68  |
| Tabla 4.16. CU-004.....                                | 69  |
| Tabla 4.17. CU-005.....                                | 69  |
| Tabla 4.18. CU-006.....                                | 69  |
| Tabla 4.19. CU-007.....                                | 70  |
| Tabla 4.20. CU-008.....                                | 70  |
| Tabla 4.21. Resumen de horas por fase.....             | 108 |
| Tabla 4.22. Resumen de costes .....                    | 109 |

# Lista de Ejemplos

|   |     |
|---|-----|
| Ejemplo 4.1. XML Operación CreateGroupS.....                                | 74  |
| Ejemplo 4.2. XML Resultado CreateGroupS .....                               | 74  |
| Ejemplo 4.3. XML Operación CreateUser .....                                 | 75  |
| Ejemplo 4.4. XML Resultado CreateUser .....                                 | 75  |
| Ejemplo 4.5. XML Operación CreateMemberRec .....                            | 76  |
| Ejemplo 4.6. XML Resultado CreateMemberRec.....                             | 76  |
| Ejemplo 4.7. Formato URL Plataforma de localización .....                   | 77  |
| Ejemplo 4.8. Resultado MFIND_IT primera sección.....                        | 79  |
| Ejemplo 4.9. Resultado MFIND_IT segunda sección .....                       | 80  |
| Ejemplo 4.10. Resultado MFIND_IT tercera sección .....                      | 81  |
| Ejemplo 4.11. Resultado MFIND_IT cuarta sección .....                       | 82  |
| Ejemplo 4.12. Resultado MFIND_IT última sección .....                       | 83  |
| Ejemplo 4.13. XML Resultado erróneo en la localización .....                | 85  |
| Ejemplo 4.14. Inclusión clave API Google Maps .....                         | 88  |
| Ejemplo 4.15. Tabla HTML con identificador del mapa .....                   | 89  |
| Ejemplo 4.16. JavaScript para generación mapa, marcador y descriptor.....   | 90  |
| Ejemplo 4.17. Fichero config.php.....                                       | 102 |
| Ejemplo 4.18. Función generate_xml del fichero lib_gpp.php.....             | 103 |
| Ejemplo 4.19. Función generate_url y localiza del fichero lib_gpp.php ..... | 103 |
| Ejemplo 4.20. Función parse_gpp del fichero lib_parse.php .....             | 104 |
| Ejemplo 4.21. Script de invocación a Google Maps .....                      | 106 |
| Ejemplo 4.22. Ejemplo de invocación a Smarty en el fichero main2.php.....   | 106 |



# CAPÍTULO 1: INTRODUCCIÓN

Con el despliegue de las redes de los operadores móviles surge la necesidad de ofrecer sobre sus capacidades básicas (llamadas, mensajería, navegación) el desarrollo de interfaces para permitir, tanto internamente como a terceros, desarrollar servicios de valor añadido sobre estas capacidades.

Aunque inicialmente la mayoría de servicios son desarrollados por el propio operador y se basan en el acceso a contenidos premium de terceros (logos y tonos), poco a poco se van desarrollando servicios más complejos y se van integrando las distintas capacidades del operador entre sí (mensajería localizada, navegación georeferenciada, etc...) que permiten acceder a directorios de servicios en base a la ubicación del usuario.

Llega un momento en el que se produce un cambio en el modelo de negocio, en el que los terceros dejan de ser proveedores de contenidos y pasan a ser desarrolladores de sus propios servicios sobre las capacidades que ofrece el operador en sus plataformas de acceso. De esta manera, el operador se ahorra el tener que personalizar sus servicios para tantos perfiles de cliente (grandes empresas, pymes, autónomos o residencial) y delega en estos terceros la relación con aquellos clientes que no tienen un volumen de tráfico que permita considerarlos grandes cuentas. Así surgen los primeros brokers de mensajería y localización que poco a poco van enriqueciendo el ecosistema de servicios entorno al operador.

En el ámbito de la localización, el despliegue de las antenas cubre prácticamente la mayor parte del país y la arquitectura de las redes del Sistema Global para Comunicaciones Móviles (en adelante GSM) incluye entidades Home Location Registers (se suelen representar por su acrónimo HLR) que permiten conocer en cada momento la ubicación en la red de los clientes. Se introduce además un nodo de red (Location Server, LS en adelante) que ofrece una interfaz interna al operador para consultar la ubicación de la celda en la que está cada abonado y poder desarrollar servicios internos basados en la ubicación del usuario.



Poco a poco va evolucionando el mercado de los servicios basados en la localización y surge la demanda, por parte de grandes clientes e integradores de poder, desarrollar servicios en base a la localización de usuarios del operador.

Surge un nuevo nodo, el Location Enabled Server (LES en adelante), que ofrece un recubrimiento a la capacidad anterior, ya que permite al operador o a un tercero el asociar la celda en la que está el abonado con unas coordenadas geográficas e incluso más información georeferenciada, como podría ser municipio, código postal, calle, servicios, etc.

La aparición de la plataforma de localización para terceros, se traduce en la apertura de una interfaz para la programación de aplicaciones (en adelante API) de localización, basada en la consulta al LS por la ubicación de la celda en la que se encuentra el abonado. Es el LES el que se encarga de transformar la información de la celda en la que está el abonado por la información geográfica que demandan los desarrolladores.

Esta API de localización se identifica como la capacidad básica del operador para ofrecer servicios de localización. Sobre ella se va a desarrollar un ecosistema de servicios y aplicaciones.

Aparecen los primeros servicios basados en localización, servicios de mensajería localizada que permiten a un usuario, mediante un mensaje corto, solicitar la dirección de un comercio cercano (por ejemplo una farmacia) así como la navegación georeferenciada, que permite navegar por un portal wap o web de servicios y que se muestre información en base a la localización de nuestro terminal.

Las empresas empiezan a ser conscientes del potencial que ofrece este nuevo sistema de localización, ya que no requiere el uso de dispositivos específicos por parte del individuo a ser localizado, más que un terminal móvil encendido, y además se obtiene la ventaja de que para su localización no es necesario tener visión directa con los satélites, como es el caso de los sistemas de localización basados en el Global Positioning System (en adelante GPS), pudiendo ser utilizado en zonas sin visión directa satelital.



Todo esto supone una reducción en la barrera de entrada de cualquier empresa que quiera gestionar la localización de sus trabajadores, ya que no tiene que reequipar a todos ellos con nuevos dispositivos, sino que puede hacerlo localizando sus terminales móviles actuales.

Uno de los sectores que se interesa desde un primer momento por esta tecnología es el del Transporte, tanto de mercancías como de personas, ya que puede abaratar la gestión de la localización flotas. De esta manera empieza a surgir en el mercado la demanda de sistemas de localización de flotas basándose en esta nueva tecnología de localización.

## **1.1. Objetivos**

Este proyecto tiene como objetivo el desarrollar una herramienta web de localización de usuarios móviles de un operador de telefonía, basado en las APIs de localización que éste ofrece. Así se permitirá a una empresa o tercero el desplegarlo en sus instalaciones y, de una manera centralizada, conocer la localización de la flota de usuarios que éste gestione.

La aplicación debe permitir a un administrador definir diversos usuarios que podrán gestionar una flota de individuos, localizable en base a las capacidades de la plataforma de localización del operador.

Cada uno de estos usuarios podrá gestionar los individuos de la flota (altas, bajas, modificar datos), localizarlos e incluso enviarles mensajes cortos para indicarles nuevas órdenes.

No sólo se pretende que los usuarios tengan acceso a las coordenadas actuales, sino que, en base a éstas, se represente en un sistema de cartografía actualizado y se complemente con la información de localización que pueda proveer la plataforma.

Para facilitar la usabilidad y reducir la curva de aprendizaje, se optará por desarrollar la aplicación sobre una interfaz web sencilla y versátil, que permita al usuario desde el primer momento sentirse cómodo y comprender su funcionamiento.

Además, se utilizará como sistema de cartografía la API de Google Maps ya que su uso está ampliamente extendido y es conocido por los usuarios.



De esta manera, se podrá implementar una solución de gestión de flotas sin necesidad de realizar un desembolso importante inicial, al evitarse la necesidad de proveer a toda la flota con sistemas de localización embarcados.

## 1.2. Estructura del proyecto

En este capítulo se ha ofrecido una breve introducción al proyecto presentándose la motivación y objetivos de éste.

En el *Capítulo 2: Estado del Arte*, se abordará el estado del arte, es decir, se describirá brevemente la situación actual de los servicios basados en la localización y el desarrollo de aplicaciones web.

En el *Capítulo 3: Herramientas para la elaboración del proyecto*, se describirán con cierto detalle las tecnologías, APIs, entornos de desarrollo y lenguaje de programación utilizados para la implementación de la aplicación.

En el *Capítulo 4: Desarrollo del proyecto*, se especificarán y analizarán los requisitos de la aplicación así como la descripción de problemas encontrados durante su desarrollo y las soluciones tomadas para la resolución de los mismos. De igual manera, se presentarán aspectos relevantes dentro de la fase de diseño tales como la arquitectura seleccionada o el diseño de la interfaz. Finalmente se incluirá el estado actual del proyecto y un resumen de éste.

El *Capítulo 5: Conclusiones*, se dedicará a las conclusiones del proyecto y a posibles recomendaciones futuras para una nueva versión de esta aplicación. Así mismo, se realizará una valoración final del proceso de desarrollo desde un punto de vista personal.

Por último, se ha incluido el *Capítulo 6: Referencias* donde se identifican cuáles han sido utilizadas durante el desarrollo del proyecto.



# CAPÍTULO 2: ESTADO DEL ARTE

Este Proyecto Fin de Carrera (en adelante PFC) consiste en el desarrollo de una aplicación informática, accesible vía Web, que permita la gestión y localización de una flota de individuos.

En este capítulo se realizará la presentación de las distintas tecnologías relacionadas con el PFC. Partiendo con una introducción a los servicios basados en localización se hará una revisión del estado actual de estos, que tecnologías utilizan, que tipo de servicios se pueden desarrollar y hacia donde evolucionan.

Como el proyecto se ha desarrollado para tener una interfaz Web, se realizará una descripción de la Web actual junto con los problemas asociados a ella y cómo está evolucionando para superarlos.

A continuación se hablará de las aplicaciones Web y los requisitos que necesitan cumplir para poder potenciar su usabilidad, objetivo a cumplir por la herramienta que gestionará la flota.

Finalmente se presentan los conceptos de desarrollo de software y lenguaje de modelado unificado que se han utilizado en el diseño del sistema.

## 2.1. Servicios Basados en Localización

A continuación se describe qué son los servicios basados en localización, cuales son las tecnologías que se utilizan actualmente para poder desplegarlos en las redes de los operadores móviles, cuales son los agentes de valor que intervienen en los servicios basados en localización y algunos ejemplos de estos.

### 2.1.1. Introducción

Los Servicios Basados en Localización (en inglés Location Based Services, LBS) se definen como servicios de que integran una localización o ubicación de un dispositivo móvil con otra información para proveer un valor agregado a un usuario.



Los servicios de localización han sido ampliamente analizados y diseñados por los organismos de estandarización, como por ejemplo el 3th Generation Partnership Project (en adelante 3GPP) encargado de especificar los fundamentos técnicos para la evolución a la tercera generación desde GSM que adoptó posteriormente la responsabilidad del mantenimiento y evolución de la red GSM actual. Se puede consultar la información relativa a LBS en (3GPP, 2000).

También se puede consultar en (OMA, 2010) información relativa a la especificación del Secure User Plan Location (SUPL) relativa a los servicios de localización. por Open Mobile Alliance (OMA). El principal objetivo de OMA es la adopción temprana del usuario de la necesidad de servicios de datos en movilidad. OMA se encarga de especificar, desde el punto de vista de la industria, la interoperabilidad de servicios, redes y operadores para favorecer usabilidad por parte de los usuarios.

### **2.1.2. Tecnologías de Localización**

Las diversas tecnologías de localización presentan distintas características en base a diversos factores:

- Grado de precisión, cada una tendrá un límite de resolución, por ejemplo en sistemas GPS con visión directa se puede llegar a obtener una precisión del orden de 3m, mientras que en localización via celdas GSM en entornos rurales se puede llegar a los 20km.
- Tiempo de respuesta de la tecnología para obtener la localización
- Consumo de recursos de red, limitarán el número de usuarios máximo que puedan hacer uso de la tecnología simultáneamente.
- Equipamiento (HW) necesario
- Datos internos/externos requeridos, etc, la tecnología puede requerir de acceso a datos externos para facilitar la localización, por ejemplo en A-GPS.
- Arquitectura:



- Plano de Control (Control Plane ): El terminal a localizar (Mobile Station, MS) es pasivo y no interviene en el proceso de su localización. El servidor (Location Server, LS) realiza el proceso de localización del terminal a partir de los recursos de red disponibles, en el caso de redes GSM (Home Location Register, Visitor Location Register, ...)
- Plano de Usuario (User Plane): El terminal móvil (MS) participa en el proceso de obtención de su posición ya sea enviando datos al servidor para que éste efectúe el cálculo o recibiendo datos de asistencia del servidor para que el terminal obtenga la posición final. Por ejemplo en el caso de servicios como Google Maps My Location o Google Latitude, que envía a un servidor el identificador de la celda, y este se encarga de devolverle unas coordenadas aproximadas de su ubicación

#### 2.1.2.1. Cell Global Identity

Los primeros servicios de localización de los operadores estaban basados en CGI (Cell Global Identity). Es la tecnología más sencilla y más utilizada por los operadores ya que utiliza la celda servidora a la que está conectado el terminal móvil para su localización

La precisión de este método depende del radio de cobertura de la celda que puede variar en 100-400 metros en áreas urbanas a 3-20km en áreas rurales.

El funcionamiento es el siguiente:

1. Cuando el dispositivo cambia de celda se informa al Visitor Location Register (en adelante VLR)
2. El VLR apunta al MS y al CellId donde está registrado
3. Si es el primer apunte en el VLR, provoca mensaje hacia el Home Location Register (HLR) indicando el nuevo VLR que da servicio
4. El HLR apunta el MS junto con el VLR actual y envía mensaje al VLR anterior para eliminación de su lista

5. El sistema Location Server (LS) accede al HLR y obtiene el VLRId actual. Con este dato, accede a dicho VLR para obtener la celda

6. A partir del CellId, el LS accede a la BBDD de celdas de la Operadora donde se encuentra la descripción geométrica del CellId asociado

A partir de esta tecnología se desarrolla **CGI-TA (Cell Global Identity-Timing Advance)** que mejora la precisión teniendo en cuenta el parámetro de avance temporal TA, que da una estimación de la distancia desde el MS a la antena. Esta nueva técnica mejora considerablemente la precisión en entornos rurales.

Dependiendo del tipo de la antena de la estación base (omnidireccional o de tres sectores) y de la disponibilidad del parámetro Timing Advance (TA), las zonas posibles de ubicación pueden variar como se ve en la siguiente figura:

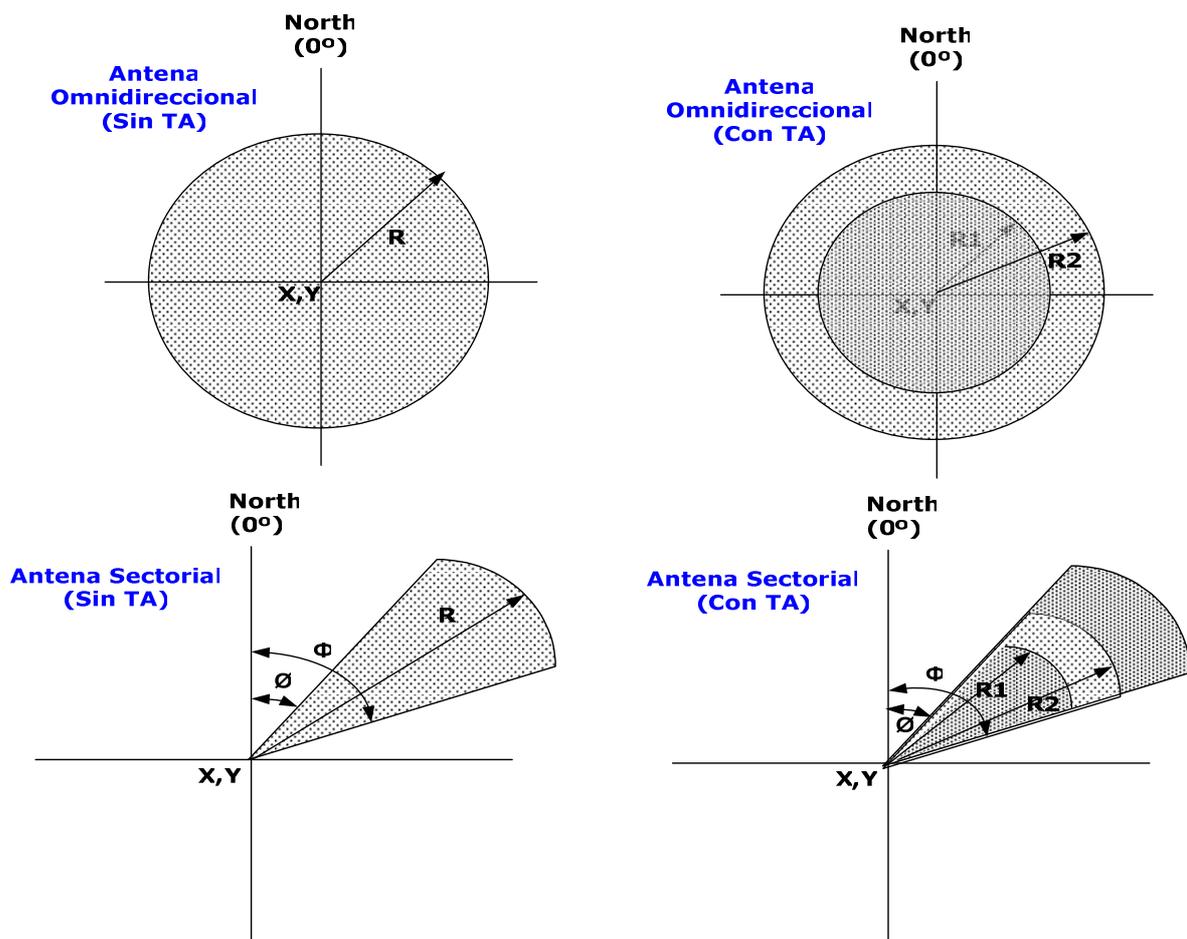


Figura 2.1. Zonas de ubicación en función de tipo de la antena

Para estaciones base con antenas omnidireccionales, no podemos reducir a un sector por el que se capta la señal del terminal, así que la zona de ubicación del terminal será esférica (caso superior izquierdo). En caso de tener acceso al parámetro Timing Advance (que depende de si el fabricante de la estación base lo soporta), se puede calcular aproximadamente la distancia a la que se encuentra el terminal de la estación base, por lo que reducimos el área en el que puede estar el terminal de un círculo a un anillo circular de radio interior  $R_1$  y exterior  $R_2$  (caso superior derecho).

Si la estación base utiliza antenas de tres sectores, cada una de estas sólo da cobertura a un ángulo horizontal de  $120^\circ$ . De esta manera identificando el sector por el que se capta la señal del terminal se reduce la zona probable de la ubicación a una porción de la anterior. En caso de que la estación base sea trisectoral y poseamos el dato del Timing Advance, podremos acotar el radio interior y exterior de la porción anterior.

#### 2.1.2.2. TOA (Time of Arrival)

Está basada en medición del tiempo de llegada de una señal transmitida por MS a diferentes BTS (mínimo 3 BTS).

La distancia recorrida se calcula como producto del tiempo empleado en llegar a BTS y la velocidad de la luz.

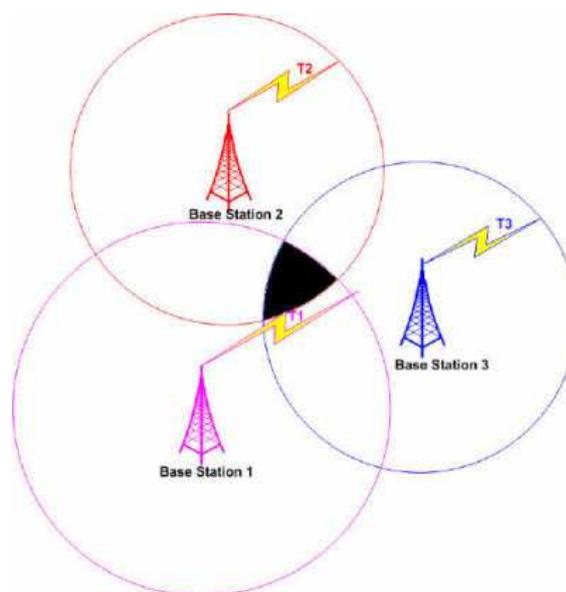


Figura 2.2. Localización en base al TOA



Las medidas permiten trazar circunferencias con centro en cada una de las BTS, dando su intersección el punto de localización del terminal.

La precisión depende de la sincronización en el tiempo de las BTS y la visión directa entre el MS y la estación base.

Una variante mejorada de la tecnología TOA es **TDOA (Time Difference Of Arrival)**. Se basa en la medición de la diferencia del tiempo de llegada de una señal transmitida por MS a distintos pares de BTS, de esta manera no es necesario conocer los tiempos de llegada sino que basta con conocer la diferencia de tiempos.

Al igual que en TOA, la sincronización entre estaciones base es muy importante, pues la falta de sincronía se traduce en errores de precisión y a diferencia de TOA, puede funcionar sin la visión directa entre el MS y la estación base ya que la diferencia de tiempos cancela posibles errores por reflexiones. El inconveniente es que en entornos urbanos hay que realizar mediciones con hasta cuatro BTS para evitar los efectos de las reflexiones.

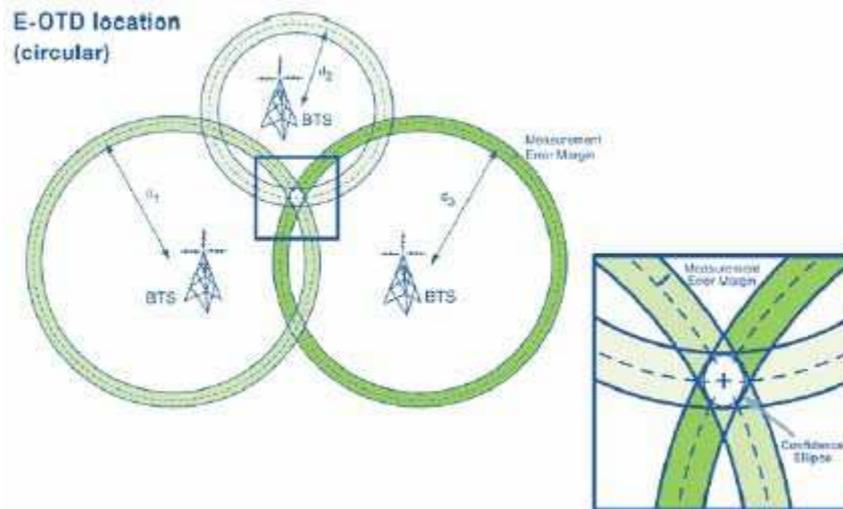
#### **2.1.2.3 E-OTD (Enhanced Observed Time Difference)**

Es una tecnología que reside tanto en el terminal móvil como en la red. La solución de red similar a la utilizada en TDOA, pero el sistema necesita de la instalación de unidades de medida de posición (LMU Location Measurement Units) en cada BTS.

Además requiere que el terminal sea compatible con la tecnología lo que complica el despliegue en los parques móviles de los operadores por tener que sufragar el cambio de terminal.

Los receptores LMU y MS habilitados con software E-OTD realizan medidas de señales procedentes de tres o más BTS periódicamente.

Las diferencias temporales de llegada de la señal a los dos puntos (LMU y terminal) se combinan para triangular la posición del terminal móvil, obteniendo una precisión de 50m en el mejor caso y 100 metros en el peor.



**Figura 2.3. Localización basada en E-OTD**

#### 2.1.2.4 GPS (Global Positioning System)

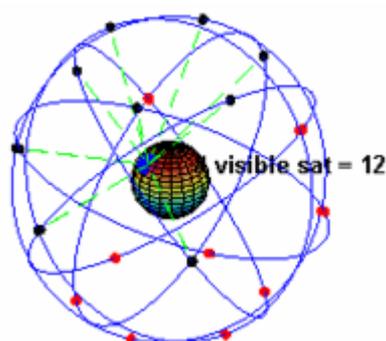
El GPS (Global Positioning System: sistema de posicionamiento global) o NAVSTAR-GPS es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de 24 satélites en órbita sobre el globo, a 20.200 km, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante "triangulación" (método de trilateración inversa), la cual se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o las coordenadas reales del punto de medición. También se

consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

El proceso inicial de búsqueda de satélites es muy lento y además la señal se ve atenuada por los edificios en ciudad, ya que requiere de visión directa con los satélites.

Aunque oficialmente los sistemas GPS indican aproximadamente 15 m (en el 95% del tiempo). En la realidad un GPS portátil monofrecuencia de 12 canales paralelos ofrece una precisión de 2,5 a 3 metros en más del 95% del tiempo. Con el WAAS / EGNOS / MSAS activado (sistema diferencial), la precisión asciende de 1 a 2 metros.



**Figura 2.4. Cobertura satelital en GPS**

#### **2.1.2.5 A-GPS (Assisted Global Positioning System)**

El GPS convencional presenta dificultades a la hora de proporcionar posiciones precisas en condiciones de baja señal. Por ejemplo, cuando el aparato está rodeado de edificios altos (como consecuencia de la recepción de múltiples señales rebotadas) o cuando la señal del satélite se ve atenuada por encontrarnos con obstáculos, dentro de edificios o debajo de árboles. De todos modos algunos de los nuevos aparatos GPS reciben mejor las señales de poca potencia y funcionan mejor en estas condiciones que aparatos más antiguos y menos sensibles.

Además, la primera vez que los receptores GPS se encienden en tales condiciones, algunos sistemas no asistidos no son capaces de descargar información de los satélites GPS como el "almanaque" y la "efemérides" (términos traducidos del inglés), haciéndolos incapaces de funcionar, triangular o posicionarse hasta que se reciba una señal clara durante al menos un minuto. Este proceso inicial, denominado primer posicionamiento o posicionamiento inicial (del inglés TTFF (Time To First Fix)



o tiempo para el primer posicionamiento), suele ser muy largo en general, incluso según las condiciones, de minutos.

Un receptor A-GPS o GPS asistido puede solucionar estos problemas de diversas formas mediante el acceso a un Servidor de Asistencia en línea (modo "on-line") o fuera de línea (modo "off-line"). Los modos en línea acceden a los datos en tiempo real, por lo que tienen la necesidad de tener una conexión de datos activa con el consiguiente coste de la conexión. Por contra, los sistemas fuera de línea permiten utilizar datos descargados previamente.

Por tanto, algunos dispositivos A-GPS requieren una conexión activa (modo en línea) a una red celular de teléfono (como GSM) para funcionar, mientras que en otros simplemente se hace el posicionamiento más rápido y preciso, pero no se requiere conexión (modo fuera de línea). Los dispositivos que funcionan en modo fuera de línea ("off-line"), descargan un fichero mientras tienen acceso a la red (ya sea a través de una conexión de datos GPRS, Ethernet, WIFI, ActiveSync o similar) que se almacena en el dispositivo y puede ser utilizado por éste durante varios días hasta que la información se vuelve obsoleta y se nos avisa de que es preciso actualizar los datos o en lugares sin conexión de datos.

En cualquier caso, el sistema de GPS asistido utilizará los datos obtenidos, de una u otra forma, de un servidor externo y lo combinará con la información de la celda o antena de telefonía móvil para conocer la posición y saber qué satélites tiene encima. Todos estos datos de los satélites están almacenados en el servidor externo o en el fichero descargado, y según nuestra posición dada por la red de telefonía, el GPS dispondrá de los datos de unos satélites u otros y completará a los que esté recibiendo a través del receptor convencional de GPS, de manera que la puesta en marcha de la navegación es notablemente más rápida y precisa.

Es una tecnología enmarcada dentro de "Plano de Usuario". El chipset embarcado en MS es capaz de obtener datos de su posición en base a satélites visibles en cada momento.

La "asistencia" que este sistema proporciona respecto al GPS tradicional radica en el uso de receptores de referencia.

Estos receptores recogen información de navegación y datos para los satélites GPS que están en la zona de cobertura del servidor de localización.

Gracias a estos datos de asistencia se evita el rastreo de todos los satélites posibles. Para obtener esta lista de satélites visibles, los sistemas LS que soportan esta tecnología se basan en el dato CGI que da cobertura al MS.

Las precisiones obtenidas a través de A-GPS llegan a márgenes de 5/10 metros.



Figura 2.5. A-GPS

#### 2.1.2.6 Near Location Based Services (NLBS)

Está basada en tecnologías de corto alcance (bluetooth, wi-fi, infrarrojo o RFID). Mediante una base de datos que asigna coordenadas geográficas a distintos puntos de acceso o identificadores de las tecnologías anteriores se puede implementar un sistema rudimentario de localización.

Un ejemplo de ello es la localización que se realiza en base a los puntos de acceso wi-fi que utiliza la versión móvil de Google Maps, ya que las tecnologías para red de área local (LAN) inalámbrica están experimentando un boom de implantación.



**Figura 2.6. Captura Google Maps Mobile**

Con la implantación de puntos de acceso wifi en hoteles, cafés, aeropuertos y otros edificios han desplegado una red de identificadores únicos (la MAC del punto de acceso) que cubren prácticamente las grandes urbes.

Mediante la generación de bases de datos que almacenan las MAC de los puntos de acceso junto con la ubicación donde se ha detectado el punto de acceso se permite mapear la ubicación en función de la cobertura de las redes wifi de alrededor.

Las redes inalámbricas cubren áreas de hasta 75 metros en el interior de edificios y de 300 metros en el exterior

La localización se lleva a cabo de diferentes maneras:

- Punto de acceso más cercano:
  - Más sencilla.
  - Método conlleva a errores. La antena más cercana puede ser la misma que un usuario situado en la planta superior

- La señal es vulnerable debido a las interferencias, afecta a la precisión.

- Wi-Fi mapping:

Basada en el almacenamiento de medida de potencia de señal en diferentes puntos del recinto cubierto (Sensible a cambios de altura)



**Figura 2.7. Ejemplo Wifi-Mapping**

**2.1.2.7 Resumen**

En la siguiente tabla se resumen la precisión, cobertura y coste de cada una de las tecnologías descritas anteriormente:



| Tecnología           | Precisión                       | Cobertura                                      | Coste despliegue                                   |
|----------------------|---------------------------------|--|--|
| CGI<br>CGI-TA        | 100-500m urbano<br>3-20km rural | Muy buena                                      | Bajo al estar basado en la red móvil ya desplegada |
| TOA<br>TDOA<br>E-OTD | 10-30m                          | Buena  | Alto   |
| GPS                  | 10-30m                          | Atenuada en zonas urbanas y en entornos indoor | Alto   |
| NLBS                 | 75-300m                         | Buena en interiores<br>Sólo zonas urbanas      | Medio  |

**Tabla 2.1. Comparativa mecanismos de localización**

### 2.1.3. La cadena del valor en la localización

La cadena de valor empresarial, o cadena de valor, es un modelo teórico que permite describir el desarrollo de las actividades de una organización empresarial generando valor al cliente final descrito y popularizado por Michael Porter en su obra (PORTER, 1998).

La cadena de valor ayuda a determinar las actividades o competencias distintivas que permiten generar una ventaja competitiva, concepto introducido también por Michael Porter. Tener una ventaja competitiva es tener una rentabilidad relativa superior a los rivales en el sector industrial en el cual se compite, la cual tiene que ser sustentable en el tiempo. Rentabilidad significa un margen entre los ingresos y los

costos. Cada actividad que realiza la empresa debe generar el mayor posible. De no ser así, debe costar lo menos posible, con el fin de obtener un margen superior al de los rivales. Las actividades de la cadena de valor son múltiples y además complementarias (relacionadas). El conjunto de actividades de valor que decide realizar una unidad de negocio es a lo que se le llama estrategia competitiva o estrategia del negocio, diferente a las estrategias corporativas o a las estrategias de un área funcional

El concepto ha sido extendido más allá de las organizaciones individuales y se aplica también en el proceso de puesta a disposición de un conjunto de servicios al consumidor final, ya que moviliza diferentes actores económicos, cada uno de los cuales gestiona su cadena de valor. Capturar el valor generado a lo largo de la cadena es la nueva aproximación que han adoptado muchos estrategias de la gestión.

En el caso de los Servicios Basados en Localización se identifican los siguientes agentes en la cadena de valor:



**Figura 2.8. Cadena de valor en la localización**

- Network: la red del operador móvil
- El Location Server (LS): es un la infraestructura de localización necesaria cuyo objetivo se centra en obtener la posición geográfica del dispositivo móvil. Ofrece un interfaz (API) simple para solicitar la posición del MS y, en su caso, con el grado de precisión requerido.
- El Location Enabling Server (LES): es la plataforma de intermediación entre las Aplicaciones LBS y los sistemas externos del Operador incluyendo:



- Sistema/s LS
  - Gateways de Mensajería (SMSC, MMSC,...)
  - Facturación (Prepago y Postpago)
  - Provisión
  - Estadísticas
- Posteriormente se añade una capa que añade información geográfica (calles, municipios, códigos postales, etc...) que permite en el siguiente agente el desarrollo de aplicaciones más enriquecidas
  - Aplicaciones: que se encargan de atacar contra el LES y se integran con los sistemas de información georeferenciada en base a la localización servida por el LES
  - El usuario o empresa, como agente final de esta cadena de valor que le da valor al servicio georeferenciado del que hace uso.

En el caso de una aplicación de gestión de flotas se identifican los siguientes agentes:

- Network: La red GSM del operador
- Location Server: El Mobile Positioning Service del operador, que se integra con la red y consulta su base de datos de estaciones base para obtener las coordenadas del usuario a localizar (MS)
- Location Enabling Server: La GPP o plataforma de localización del operador que se encarga de la autenticación y autorización de los usuario que tienen acceso a localizar equipos y la generación de registros para el cobro de las localizaciones. También se integra con otros elementos como pasarelas de mensajería o bases de datos geográficas que es la siguiente capa
- Aplicaciones: Las que se desarrollan sobre la API que ofrece el LES, en este caso el objetivo de este proyecto



- Usuario: el cliente que utilice la aplicación para localizar a algún elemento de su flota.

#### **2.1.4. Tipología de servicios basados en localización**

Se suelen categorizar en base a:

- Quién solicita la posición del MS:
  - El propio usuario (Auto localización)
  - Otro usuario inscrito en el servicio LBS (Localización de terceros):
    - Usuarios Conocidos
    - Usuarios Desconocidos
    - El servidor de aplicación
- Cuál es el flujo de la petición de localización:
  - Petición directa desde el terminal del usuario (localización activa)
  - Petición Indirecta a través de servidores de Aplicación (Internos o Externos al Operador)
- Quién conoce la identidad real del usuario (MSISDN):
  - Sólo es conocida por los sistemas del Operador
  - Es conocida por el usuario que solicita mi posición
  - Es conocida por el servidor de aplicación

La mayoría de los servicios de gestión de flotas se instalan en vehículos de empresas de transporte de mercancías y/o personas.

Suelen existir dos tipologías básicas de servicios de gestión de flotas:



- El dispositivo embarcado se encarga de mandar la posición del vehículo a intervalos regulares de tiempo vía SMS o mediante una conexión GPRS/UMTS
- La central solicita una localización del vehículo al sistema de gestión de flotas.

El primer tipo requiere de dispositivos integrados en el camión que se encarguen de mandar la localización. Suelen ser bastante caros y requieren un consumo de datos elevado, bien en volumen de SMS o por las conexiones de datos.

En el segundo caso hay menor tráfico de datos y existen dos posibilidades:

- Dispositivos embarcados que responden a peticiones SMS enviadas desde la central y enviando como respuesta las coordenadas del GPS con el que están integrados
- Terminales móviles que son localizados desde el operador móvil

En el caso de los dispositivos embarcados se reduce el volumen de información que se transmite y abarata el servicio, pero sigue existiendo el problema de que ante situaciones en las que no haya visión directa con los satélites GPS (túneles, parkings, interior de edificios o calles estrechas con edificios altos) no podrán ser localizados.

Con la opción de los terminales móviles se opta por una opción más barata en su despliegue (cualquier terminal del operador, hasta el más básico, puede ser localizado y se evita el problema de las zonas interiores con falta de visión directa satelital. Estos beneficios tienen la contraprestación de que la precisión de la localización será menor y que habrá que realizar un pago al operador móvil por cada operación de localización. Este modelo está más adaptado para PYMES que no pueden asumir un coste elevado de implantación de una solución de gestión de flotas y les permite racionalizar el consumo del tráfico de localizaciones.



### 2.1.5. Ejemplos de servicios basados en localización

A continuación se muestran algunas soluciones de gestión de flotas comerciales:

- Invesfleet (<http://www.invesfleet.com>) tiene una solución de gestión de flotas basada en dispositivos embarcados con conexión a GPS que guarda históricos de desplazamientos y realiza gestión de los conductores y vehículos, así como posibilidad de visualizar los recorridos en sistemas de cartografía. Entre sus principales características destacan:
  - Posicionamiento de vehículos y puntos de actuación: Permite visualizar donde están los vehículos y crear ordenes de trabajo indicando la dirección exacta donde hay que prestar el servicio.
  - Asignación inteligente de órdenes de trabajo: InvesFleet le ayuda a determinar cual es el mejor vehículo a asignar a una orden de trabajo nueva (según la distancia, tipo de orden de trabajo, por tipo de vehículo, etc. ).
  - Gestión de vehículos: Establezca las características de cada vehículo de la flota.
  - Gestión de conductores: Administre qué conductores estarán disponibles.
  - Comunicación con sus conductores: Permite enviar las órdenes de trabajo a los conductores y recibir de éstos el detalle de la finalización de las órdenes en tiempo real.
  - Estado de sus vehículos: Permite conocer el estado de cada vehículo: Libre / Ocupado/ etc. Alertas. Velocidad. Etc.
  - Avisos y Alertas: Permite configurar cuales son los avisos y alertas que se desea que InvesFleet informe, tales como: Movimiento de vehículo no autorizado, sensor del vehículo activado/desactivado, cambio de aceite, etc.

- Navegación con TomTom: El conductor al recibir el parte de trabajo, automáticamente TomTom le guiará al destino especificado sin que tema perderse.
- Zonas Permitidas: Se puede establecer las zonas geográficas en las que está permitida la conducción de cada uno de los vehículos.
- Histórico de recorridos: Permite conocer el trayecto de cada uno de sus vehículos. Indicando los puntos de parada, tiempos de parada, velocidad, kms recorridos, etc.
- Gestión de usuarios: Se puede administrar los usuarios de la aplicación, otorgándoles permisos a cada característica de la aplicación.
- Cartografía siempre actualizada: Mediante la cartografía de Google Maps se dispondrá de la última cartografía disponible.



**Figura 2.9. Logo de Ivesfleet**

- Zenithal (<http://www.zenithal.es>) oferta una solución basada en otro dispositivo embarcado con integración con el tacógrafo digital y permite realizar informes de actividad de toda la flota. Tiene las siguientes funcionalidades:
  - Localización de la flota en tiempo real
  - Históricos (Recorridos de la flota) en tiempo real sobre cartografía vectorial o fotografía aérea
  - Posicionamiento de puntos de Interés (POIs)
  - Informes de actividad de la flota describiendo:

- Recorridos en texto
- Kilometrajes
- Tiempos
- Tiempo en clientes
- Tiempo en zonas NO autorizadas
- Velocidades
- Id conductor
- Temperatura

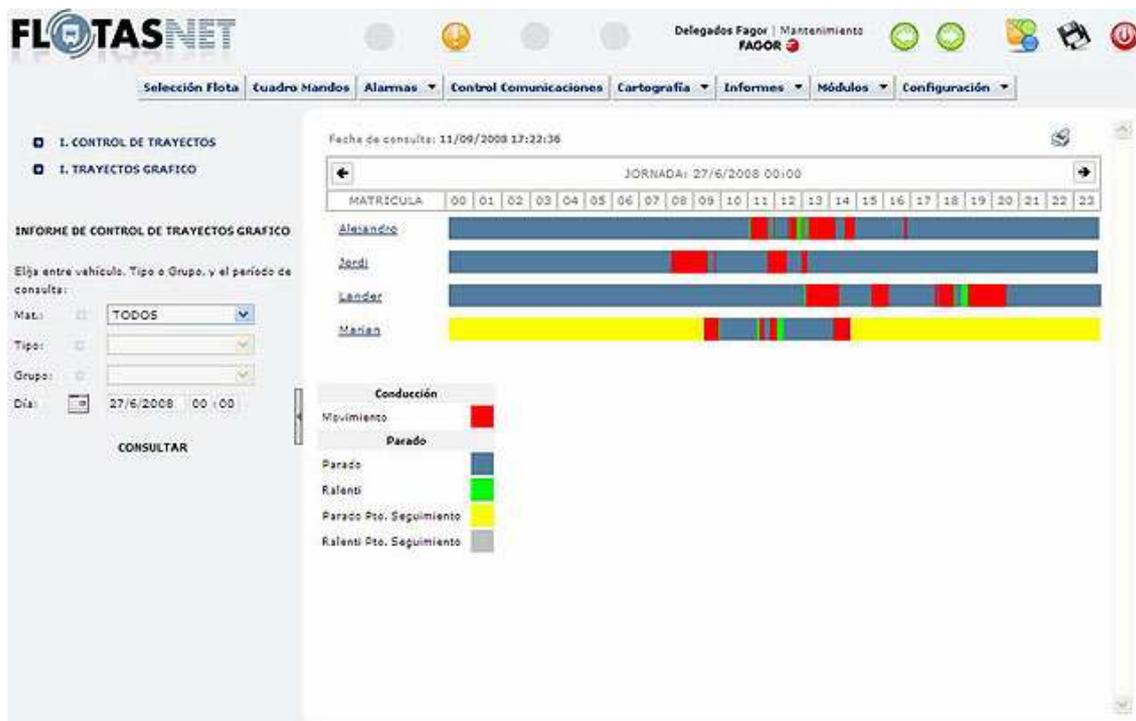


**Figura 2.10. Ejemplo de dispositivo embarcado**

- Flotasnet (<http://www.fagorelectronica.es/flotas/flotasnet.html>) de Fagor Electrónica ofrece un producto con las siguientes funcionalidades:
  - Selección de la Flota
  - Cuadro de Mandos
  - Información textual de la situación geográfica y del estado de la flota.
  - Información sobre tiempos de conducción y parada, actuales y acumulados.
  - Gestión de Alarmas
  - Gestión de Mensajes
  - Gestión de Eventos



- Gestión de Alarmas Especiales
- Gestión de Alarma de Velocidad
- Redireccionamiento de las Comunicaciones
- Configuración de las Comunicaciones
- Envío de Mensajería a la Flota.
- Cartografía a nivel de callejeros, con sentidos, numeración de las calles, y códigos postales.
- Búsquedas por calle, número, localidad y código postal.
- Búsqueda de vehículos más cercanos a un vehículo o a otro punto.
- Búsqueda de puntos más cercanos a vehículos y a otro punto.
- Medición de distancias.
- Gestión eficiente de zonas geográficas (vistas).
- Gestión ágil y eficaz de los zooms y visualización.
- Representación del vehículo diferencia con respecto a la velocidad que lleva.
- Gestión de históricos
- Integración con GoogleEarth
- Creación de Puntos Geográficos Personalizados. Puntos de carga, descarga, proveedores, clientes, talleres, etc.
- Informes
- Configuración de equipamiento embarcado y sistema



**Figura 2.11. Captura de FLOTASNET**

Al contrario que los sistemas anteriores, se va a optar por desarrollar un sistema basado en localizaciones realizadas por el operador. De esta manera abarataremos el despliegue inicial de la solución, ya que cualquier terminal móvil del operador puede ser localizado y sin tener que recurrir a dispositivos embarcados integrados con GPS cuyos costes se dispararían.

## 2.2. World Wide Web

De un tiempo a esta parte (CASTEJÓN, 2004), la rápida expansión de Internet ha supuesto una transformación en las necesidades de información de las organizaciones. No existe prácticamente ninguna empresa u organismo que no necesite que la información esté accesible tanto dentro como fuera de su edificio o que ésta sea compartida entre todas las partes interesadas en ella, ya sea en su totalidad o en aquella parte que corresponda según su función.

Estas necesidades han sido las causantes de un movimiento creciente de cambio desde las clásicas aplicaciones de escritorio a aplicaciones Web, que las satisfacen ampliamente. Así, las páginas Web, que antes tan sólo se dedicaban a mostrar información, se han convertido en aplicaciones con las cuales el usuario interactúa.



Hoy el foco se encuentra en la construcción de sitios Web potentes que proporcionen valor real y ofrezcan una experiencia positiva al usuario o cliente. Cuando los visitantes coinciden en evaluar con alta puntuación el contenido, facilidad de uso, rendimiento, fiabilidad y satisfacción general, se denomina sitio Web centrado en el usuario.

Frente a las imposiciones de diseñadores, tecnología o compañía, el diseño de un sitio Web debe centrarse en el cliente o usuario. Construir sitios Web centrados en el usuario se basa en proporcionar una experiencia positiva para todos los visitantes, ya sea en la búsqueda de información, formar parte de una comunidad, comprar artículos o entretenerse. Este enfoque enfatiza la importancia de comprender la necesidad del usuario, las herramientas y tecnologías que utiliza, y su contexto social y organizativo. Además, concierne al modo en que ese conocimiento del usuario se plasma en los diseños, que deberán ser probados para asegurar que las necesidades detectadas son cubiertas.

Incluso los desarrollos de sitios en los que no hay competidores directos, como es el caso de instituciones educativas e intranets corporativas, pueden beneficiarse de la orientación a usuario. Sitios Web sencillos, claros y bien diseñados pueden reducir drásticamente el tiempo de trabajo de usuarios, reducir los costes de mantenimiento y mejorar la satisfacción general.

### **2.2.1. La usabilidad en la Web**

La usabilidad (dentro del campo del desarrollo web) es la disciplina que estudia la forma de diseñar sitios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible. Dado que la aplicación que se está desarrollando está claramente orientada al usuario se seguirán, en el grado en que sea posible, los diez principios básicos de la usabilidad, enunciados por el gurú en esta materia, Jakob Nielsen. Estos principios son (NIELSEN, 1994):

**Visibilidad del estado del sistema para los usuarios** El sistema debe mantener a los usuarios informados en todo momento sobre lo que está sucediendo, utilizando para ello sistemas adecuados en tiempos razonables.



**Adecuación entre el sistema y el mundo real** El sistema debe hablar el lenguaje de sus usuarios con palabras y conceptos comprensibles por ellos. Se deben evitar, por tanto, términos orientados al sistema. Son necesarias las convenciones del mundo real y sobre todo, hacer que la información aparezca en un orden lógico.

**Libertad y control por parte del usuario** Es necesario implementar un mecanismo de hacer / deshacer, de esta manera el usuario no se verá obligado a buscar puertas de emergencia cuando piense que ha cometido un error.

**Consistencia y estándares** Los usuarios no tienen que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Es necesario, por tanto, seguir los estándares de plataforma.

**Prevención de errores** El lugar de buenos y vistosos mensajes de error es mejor tener un buen diseño, que sea cuidadoso con los errores en el primer intento.

**Reconocimiento antes que recuerdo** Las instrucciones de uso del sistema deben encontrarse siempre visibles o fácilmente recuperables cuando sean necesarias.

**Flexibilidad y eficiencia en el uso** El sistema tiene que estar preparado para usuarios expertos e inexpertos de igual manera.

**Diseño estético y minimalista** Las alertas o mensajes que se muestren no deben contener información irrelevante o innecesaria. Cada unidad de información extra compite con la información realmente relevante, minimizando su visibilidad frente al resto.

**Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores** Los mensajes de error deben estar expresados en un lenguaje entendible para el usuario, describiendo de forma precisa el problema y proporcionando una solución clara.

**Ayuda y documentación** Es necesario dotar al sistema de un módulo de ayuda y documentación. Cualquier información debe ser fácil de buscar, estar centrada en la acción de usuario, proporcionar una lista de pasos a seguir y no ser demasiado extensa.



### 2.2.2. Elementos clave de sitios Web centrados en usuario

Dado que la aplicación que se quiere desarrollar está claramente orientada al usuario se intentará seguir las siguientes pautas:

#### **Facilidad de uso**

La experiencia en el uso de las tecnologías de la información varía enormemente entre los usuarios. Es necesario que el sitio sea tan sencillo de usar como sea posible. El diseño de la interfaz de usuario ocupa muchos libros, pero se ofrecen unas líneas básicas:

- Mantener el sitio tan sencillo como sea posible. Cuantas más opciones, publicidad y distracciones ocupen la pantalla, más fácilmente se confundirá el usuario.
- Mantener el texto claro. Utilizar fuentes sencillas y claras. No utilizar tamaños de fuente muy pequeños y mantener en mente que el tamaño puede variar entre diferentes tipos de máquinas.
- Simplificar los procesos de interacción con el usuario. Tanto la intuición como la evidencia soportan la idea de que cuanto mayor es el número de clicks para completar un proceso (por ejemplo una hoja de pedido), menor probabilidad hay de que el usuario lo complete. Se debe mantener el número de pasos al mínimo.
- No permitir que el usuario se pierda. Deben proporcionarse señales y pistas de navegación que indiquen al usuario dónde se encuentra. Por ejemplo, si se utiliza la metáfora de la cesta de compra, que permite al usuario acumular compras en un contenedor virtual antes de finalizar el pedido, se debe mantener un enlace a la cesta visible en todo momento.

#### **Rendimiento**

No sólo referido al tiempo de descarga de las páginas, sino a la implementación de los procesos en los que el usuario tenga que interaccionar con el sistema. La navegación debe ser fluida y debe minimizarse el retardo introducido por el



procesamiento en el sitio Web. Este parámetro puede ser especialmente crítico en intranets donde el usuario espera la misma respuesta que ofrecen las aplicaciones locales.

### **Satisfacción**

La medida en que el sitio Web se adecua a sus objetivos es proporcional a la satisfacción de los usuarios en su uso. Para comprender la importancia de este elemento, sólo tendríamos que recordar cuál fue el último portal Web visitado que nos sorprendió positivamente, ya fuera por su presentación o eficacia.

### **Contenido**

La información publicada en el sitio Web debe orientarse al usuario. El contenido, tanto en su estructura, como redacción y mensaje, será fiel al valor de marca. El sitio debe ofrecer información de utilidad al cliente sin sobrecargarle con datos innecesarios.

### **Incompatibilidad entre navegadores**

Se debe verificar el funcionamiento del sitio en diferentes navegadores y sistemas operativos. Si el portal no funciona para un navegador popular, el sitio parecerá poco profesional y se perderá una sección de usuarios potenciales.

## **2.3. Desarrollo de Software**

En esta sección se explicará qué es un proceso de desarrollo software, se mostrarán sus actividades y su organización en modelos de ciclo de vida. Se describirá brevemente el lenguaje de modelado unificado y, por último, se presentarán los estilos y patrones de diseño que se utilizarán en este proyecto.

### **2.3.1. Proceso de desarrollo de software**

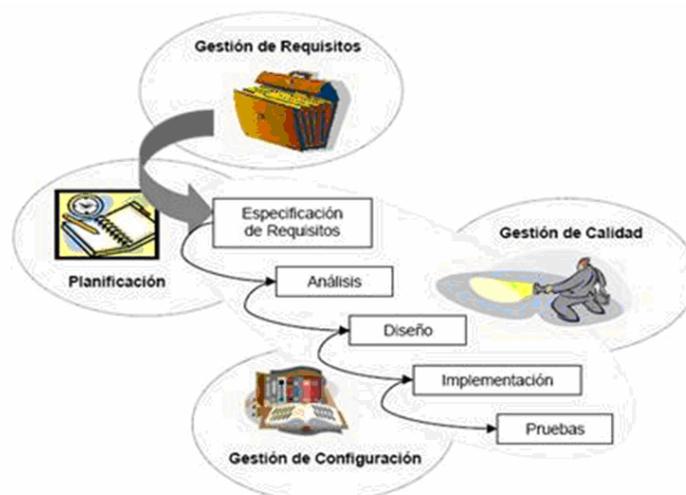
La idea de aplicar el concepto de proceso al desarrollo software proviene del campo de la fabricación, donde los procesos (pasos en la fabricación) están definidos y se controlan de manera continua. Este enfoque se puede aplicar al mundo del software

para la gestión del proceso a nivel de proyecto y para mejorar las capacidades de los grupos de desarrollo. Según (CUEVAS, 2003), el proceso software establece el marco de trabajo tanto técnico como de gestión para poder aplicar los métodos, herramientas y personas a la tarea de desarrollo de software. La definición del proceso identifica los roles y las tareas específicas y establece medidas para el control de ejecución de cada paso.

Se denomina definición del proceso software a la descripción del proceso que se sigue para elaborar el software. La definición adecuada del proceso permitirá a una organización dedicada al desarrollo de software asegurar que cada elemento de trabajo se asigna apropiadamente y que se conoce su estado en cada momento. A su vez, esta definición oficial del proceso en la organización, permitirá incorporar nuevos métodos que lo mejoren.

Un proceso definido permite que cada nuevo proyecto sea construido en base a la propia experiencia y la de sus predecesores. Los problemas recogidos a medida que se utiliza el proceso permiten identificar sus causas y corregirlos. De este modo, tanto el proceso, como su definición y la infraestructura de soporte evolucionarán con la experiencia.

En la figura 2.12 se muestra el proceso de desarrollo donde se han incorporado las actividades de gestión: gestión de requisitos, planificación, gestión de la configuración y gestión de calidad.



**Figura 2.12. Representación de un proceso de desarrollo de software**



### **2.3.1.1. Actividades de gestión del proceso de desarrollo de software**

Las actividades de gestión están orientadas a controlar el desarrollo del proceso software y a corregir las desviaciones con respecto a los parámetros de calidad establecidos. Así, son tareas que se desarrollan a lo largo del ciclo de vida del software.

#### **Gestión de requisitos**

La gestión de requisitos, una vez obtenida la especificación de requisitos, se ocupa de fijar el marco de referencia, de carácter contractual, que determinará el desarrollo. Incluirá todas las actividades de control que mantengan la integridad y exactitud del acuerdo sobre los requisitos a medida que progrese el proyecto. Para ello, llevará a cabo el tratamiento y control de las actualizaciones y cambios a los mismos garantizando su trazabilidad en el ciclo de vida.

#### **Gestión de configuración**

La gestión de configuración del software es el medio que permite conocer, en todo momento, qué componentes y versiones, tanto de un producto como de sus elementos, son las correctas. De forma más concreta, la gestión de configuración es el proceso de identificar y definir los elementos de la configuración para:

- Controlar la liberación de versiones y cambios durante todo el ciclo de vida.
- Registrar e informar de su estado y de las peticiones de cambios.
- Verificar la corrección y acabado de los elementos.

#### **Gestión de calidad**

Una vez establecidos los objetivos de calidad de un proyecto software y especificados aquellos procesos operativos y recursos necesarios para satisfacer dichos objetivos, la gestión de calidad será el conjunto de actividades diseñadas para evaluar el proceso por el que se desarrollan los productos. Deberá garantizar que el sistema, componente o proceso cumple los requisitos especificados y cubre las necesidades o expectativas del cliente o usuario.

#### **Planificación**

La planificación del proyecto consistirá en establecer la estructura temporal de las fases, actividades y tareas del proyecto, en función de los recursos de que se



disponga. El seguimiento y control de evolución determinarán la necesidad de nuevas planificaciones que puedan alterar la fecha de finalización de proyecto.

### 2.3.2. Ciclo de vida del Software

El ciclo de vida software es el periodo que comienza cuando un producto software es concebido y termina cuando deja de estar disponible (CUEVAS, 2003). El ciclo de vida se divide normalmente en fases que estructuran y organizan las etapas de concepción, desarrollo y mantenimiento. Un modelo de ciclo de vida es la descripción de las distintas formas de desarrollo de un proyecto, es decir, la orientación que debe seguirse para obtener a partir de los requerimientos del cliente, sistemas que puedan ser utilizados por dicho cliente.

Las funciones principales de un ciclo de vida software son:

- Determinar el orden de las fases y procesos involucrados en el desarrollo del software y su evolución incluyendo la explotación y el mantenimiento.
- Establecer los criterios de transición para pasar de una fase a la siguiente (productos intermedios). Todo ello incluye los criterios para verificar la terminación de la fase actual y los criterios para seleccionar e iniciar la fase siguiente.
- El modelo de ciclo de vida se escoge en la planificación inicial del proyecto. Cada proyecto se estructura en actividades, que serán previamente planificadas y agrupadas en fases. Además, los productos deben ser terminados y entregados, de acuerdo al plan, y funcionar correctamente. Todo ciclo de vida debe incluir las fases incluidas en la tabla 2.2.

| FASE      | DEFINICIÓN                                | ENTREGABLE            |
|-----------|---|-----------------------|
| <b>UR</b> | Definición de Requerimientos de Usuario   | <b>URD</b>            |
| <b>SR</b> | Definición de Requerimientos del Software | <b>SRD</b>            |
| <b>AD</b> | Diseño de la Arquitectura                 | <b>ADD</b>            |
| <b>DD</b> | Diseño Detallado y codificación           | <b>DDD, SUM, CODE</b> |
| <b>TR</b> | Transferencia del software                | <b>STD</b>            |
| <b>OM</b> | Operación y mantenimiento                 | <b>PHD</b>            |

**Tabla 2.2. Fases de desarrollo del proyecto**

Las primeras cuatro fases finalizan con una revisión. Estas fases han de aparecer siempre, independientemente del tamaño, tipo, software empleado, o equipo de desarrollo, ya sea propio o externo. El ciclo de vida del software, siguiendo la notación del estándar PSS-05 de la ESA (ESA, 2008), empieza realmente con la entrega del Documento de Requisitos de Usuario (URD, User Requirements Document) a los desarrolladores. Por lo tanto, la revisión del URD es la primera actividad del ciclo de vida. Junto con esta revisión, ha de producirse la entrega de un plan de gestión de proyecto, incluyendo una planificación y una estimación de costes para el proyecto. Los entregables de cada fase deben de ser revisados y aprobados antes de proceder con la siguiente fase. Hay seis hitos que indican el progreso del ciclo de vida:

- Aprobación del Documento de Requisitos de Usuario (User Requirements Document: URD).
- Aprobación del Documento de Requisitos de Software (Software Requirements Document: SRD).



- Aprobación del Documento de Diseño Arquitectónico (Architectural Design Document: ADD).
- Aprobación del Documento de Diseño Detallado (Detailed Design Document: DDD), Manual de Usuario del Software (Software User Manual: SUM) y Código, así como confirmación de disponibilidad para las pruebas de aceptación provisional.
- Entrega del Documento de Transferencia del Software (Software Transfer Document: STD) y confirmación de aceptación provisional.
- Entrega del Documento Histórico del Proyecto (Project History Document: PHD) y confirmación de aceptación final.

A continuación se describen las fases del ciclo de vida del software.

### **Definición de requisitos del usuario**

En esta fase se trata de plantear el problema que deberá ser solucionado mediante un sistema software. Para ello, se realizarán entrevistas con el cliente, de forma que se obtienen los requisitos que debe tener el sistema para resolver sus necesidades.

### **Definición de requisitos del software**

Esta es la fase de análisis del problema. A partir de las necesidades que han originado la puesta en marcha del proyecto, se trata de comprender qué es lo que tiene que realizar el sistema en su conjunto. Se analizará el entorno en el que se desarrollará el sistema y la manera en que éste interactuará con él.

### **Diseño de la arquitectura**

Esta fase corresponde al planteamiento de la solución. El sistema se descompone en un conjunto de subsistemas, de manera que se simplifica el sistema global. Se definen las relaciones que tienen cada uno de los subsistemas con los demás.



### **Diseño detallado y codificación**

En esta fase se refina el diseño realizado en la fase anterior. Además se realiza la implementación del sistema, se valida y se verifica, de manera que el sistema realizado cumpla con los requisitos definidos en las fases iniciales.

### **Transferencia del software**

En esta fase, el sistema es transferido a su entorno de explotación. Puede haber un periodo de enseñanza para que el usuario conozca la manera de utilizar el sistema.

### **Operación y mantenimiento**

Esta suele ser la fase más larga del ciclo de vida. Transcurre durante el uso del sistema en su entorno de explotación. Además tendrán lugar actuaciones de mantenimiento y mejora del sistema.

El proyecto se basa en un ciclo de vida “híbrido” entre el modelo de ciclo de vida en cascada y el modelo de ciclo de vida iterativo-incremental. A continuación se describirán estos dos ciclos de vida por su relación con este trabajo.

#### **2.3.2.1. Ciclo de vida en cascada**

El ciclo de vida en cascada es el ciclo más simple. Históricamente, fue el primer ciclo en aparecer. La vida de los sistemas pasa por una serie de fases consecutivas y separables. Las fases se desarrollan en secuencia y sólo una vez, aunque puede haber iteraciones dentro de cada fase (CUEVAS, 2003).

Este ciclo de vida se ha utilizado para realizar un primer enfoque de los requisitos y análisis del proyecto a nivel global. En la Figura 2.10 se esquematiza este ciclo de vida.

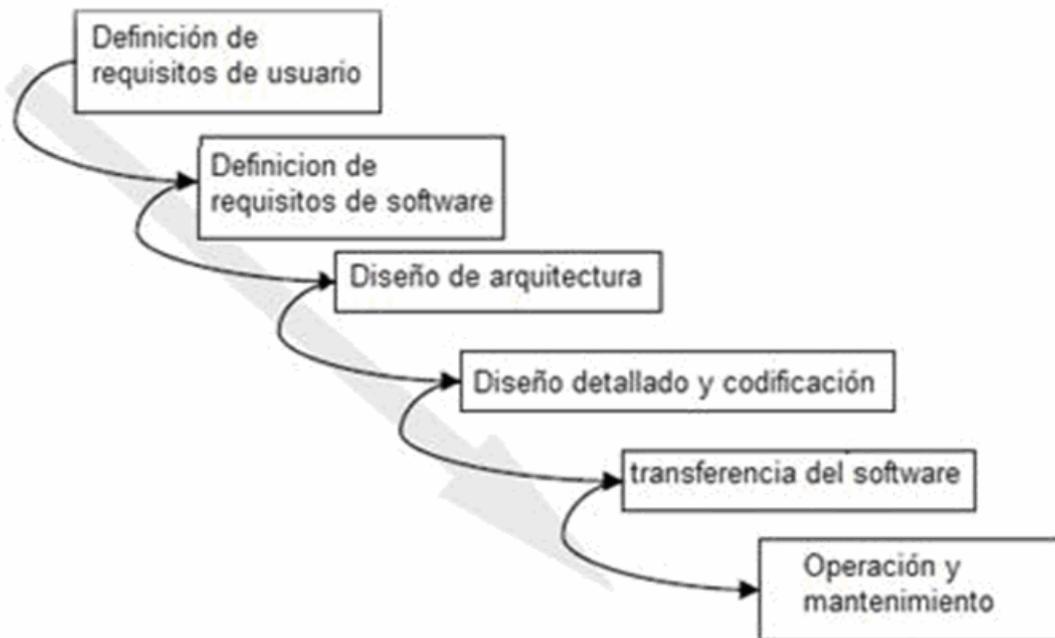
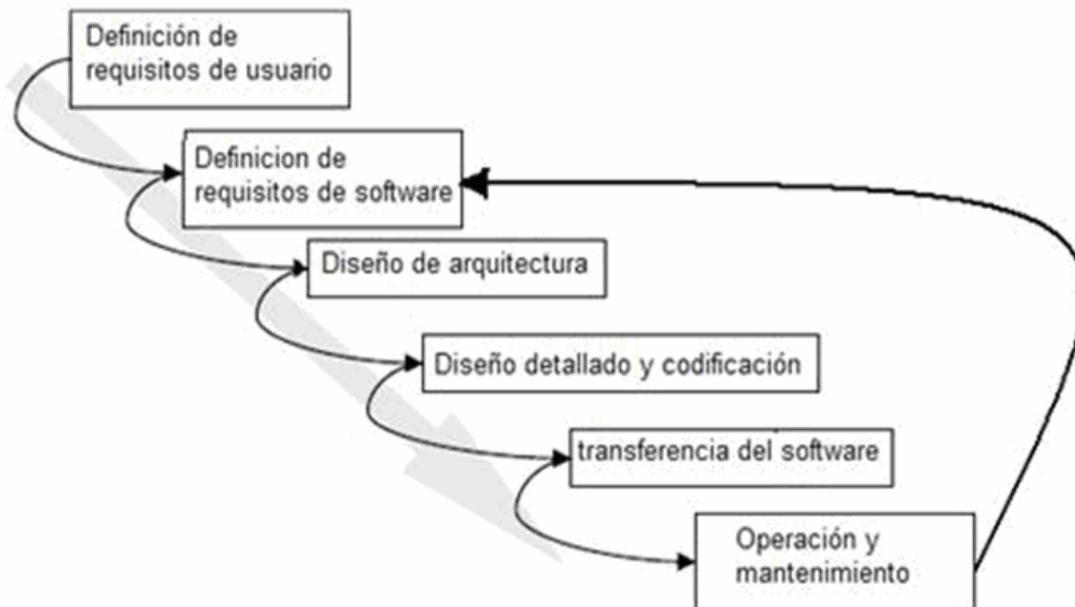


Figura 2.13. Ciclo de vida en cascada

### 2.3.2.2. Ciclo de vida iterativo-incremental

Este ciclo de vida comienza determinando la especificación de requisitos y realizando un análisis del sistema, para ello se recogen las necesidades del usuario y se realiza el diseño del sistema y la planificación de la construcción. A partir de ese momento se realiza el resto del desarrollo como una secuencia de entregables en que cada entregable incorpora una parte de las capacidades planificadas. En otras palabras en el ciclo de vida iterativo-incremental se va creando el sistema software añadiendo componentes funcionales al sistema.

En la ejecución de este proyecto inicialmente se realizó una especificación de requisitos de usuario, así como los requisitos de software y el diseño de la arquitectura basado en el ciclo de vida en cascada. Posteriormente se ha realizado el diseño detallado y la codificación, así como la transferencia de software de forma iterativa/incremental. En la Figura 2.11 se muestra el ciclo de vida iterativo-incremental.



**Figura 2.14. Ciclo de vida iterativo-incremental**

### 2.3.3. UML

El Lenguaje Unificado de Modelado (UML) (BOOCH, 2006) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende ser apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

(ALARCÓN, 2007) UML es un lenguaje que nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad. En este contexto, UML sirve para especificar, modelos concretos, no ambiguos y completos.



UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos.

En la definición de UML, se establecieron como objetivos principales la consecución de un método que aunara los mejores aspectos de sus predecesores. Para ello, se plantearon las siguientes características:

- El método debe ser capaz de modelar no sólo sistemas de software, sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).
- El lenguaje de modelado que se pretendía definir, debía poder ser utilizado, a la vez, por máquinas y por personas.
- Establece un acoplamiento explícito de los conceptos y los artefactos ejecutables.
- Maneja los problemas típicos de los sistemas complejos de tiempo real.

Lo que se pretende con esto, es lograr que los lenguajes que se aplican siguiendo los métodos más utilizados sigan evolucionando en conjunto y no por separado. Y además, unificar las perspectivas entre diferentes tipos de sistemas, no sólo software, al facilitar la comprensión de las fases de desarrollo, los requerimientos de análisis, el diseño, la implementación y los conceptos propios de la orientación a objetos.

### **2.3.3.1. Diagramas UML**

Un diagrama (HERNANDEZ, 2005) es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia



variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

Los diagramas más utilizados y empleados para el desarrollo del proyecto son, los de casos de uso por lo que nos centraremos en éstos.

### **Diagrama de caso de uso**

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema, mediante su interacción con los usuarios, u otros sistemas. Esto quiere decir que, estos diagramas muestran la relación entre los actores y los casos de uso. En este tipo de diagrama intervienen algunos conceptos nuevos. Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema, en respuesta a un evento que inicia un actor sobre el propio sistema. Una relación es una conexión entre los elementos del modelo. Un actor es una entidad externa al sistema, que se modela y que puede interactuar con él. Un ejemplo de actor, podría ser un usuario o cualquier otro sistema.

Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un caso de uso

- Un caso de uso extiende otro caso de uso (extend)
- Un caso de uso usa otro caso de uso (include)

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema, al mostrar la reacción producida como respuesta a los eventos que se producen en el mismo, en la Figura 2.12. “Ejemplo de diagrama de casos de uso”, se muestra un ejemplo.

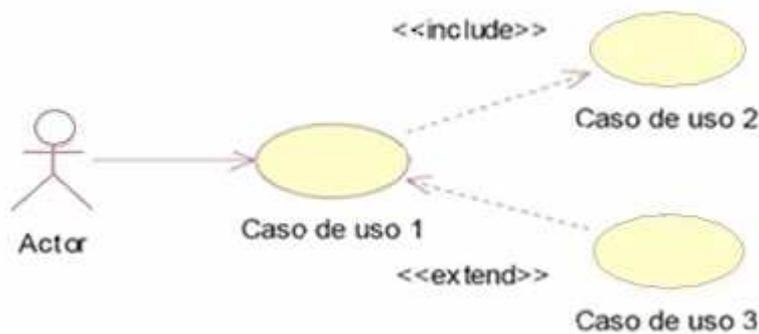


Figura 2.15. Ejemplo de diagrama de casos de uso

### Diagrama de clases

Los diagramas de clases representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos. Algunos de los elementos que se pueden clasificar como estáticos en UML, son los siguientes:

- Paquete: Es el mecanismo de que dispone UML para organizar sus elementos en grupos.

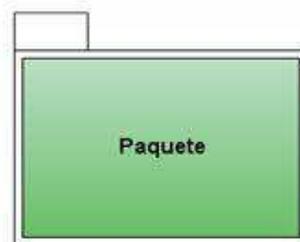


Figura 2.16. Figura de un paquete

- Clase: Describe un conjunto de objetos que comparte los mismos atributos, operaciones, métodos, relaciones y significado. Los componentes de una clase son:
  - Atributo: Se corresponde con las propiedades de una clase.
  - Operación: También conocido como método, es un servicio proporcionado por la clase que puede ser solicitado por otras clases y que produce un comportamiento en ellas cuando se realiza.

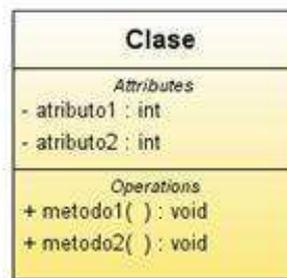


Figura 2.17. Figura de una clase

- Interfaz: Representa a la funcionalidad que proporciona uno o varios elementos del modelo al resto de elementos. Esta interfaz no tiene asociado un comportamiento, por lo que tendrá que ser implementado por otro elemento.

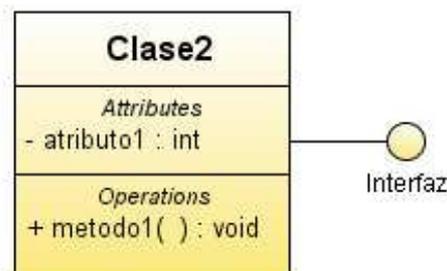
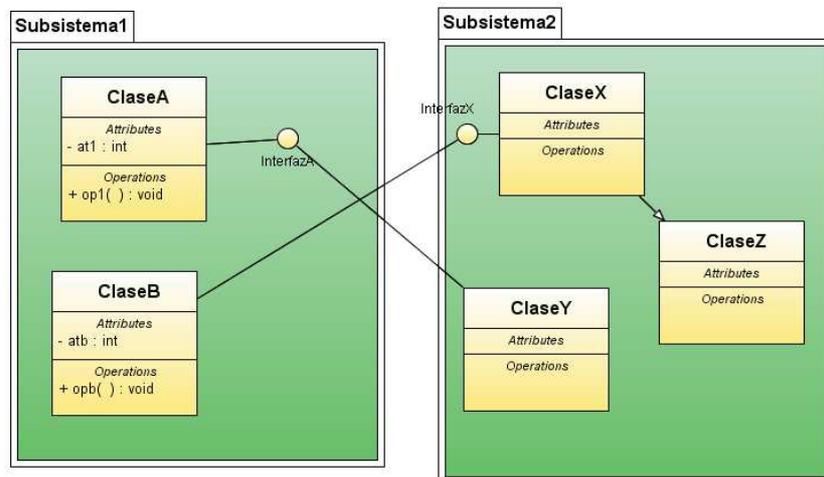


Figura 2.18. Figura de una interfaz

En estos diagramas se muestra, además, las relaciones estructurales existentes entre los elementos descritos anteriormente. Un ejemplo de un diagrama de clases completo, puede ser el de la Figura 2.16. “Ejemplo de un diagrama de clases”.



**Figura 2.19. Ejemplo de un diagrama de clases**

### Diagrama de secuencia

Muestran las interacciones entre un conjunto de objetos, ordenadas según el momento del tiempo en que tienen lugar. El objeto puede existir sólo durante la ejecución de la interacción, momento en el que puede ser creado o destruido. Un diagrama de secuencia representa una forma de indicar el periodo durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.

En este tipo de diagramas también intervienen los mensajes, que son la forma en que se comunican los objetos. Los mensajes consisten en la solicitud de una operación de un objeto origen a un objeto destino. Existen distintos tipos de mensajes según cómo se producen en el tiempo: simples, síncronos, y asíncronos.

Los diagramas de secuencia permiten indicar cuál es el momento en el que se envía o se completa un mensaje mediante el tiempo de transición, especificado en el diagrama.

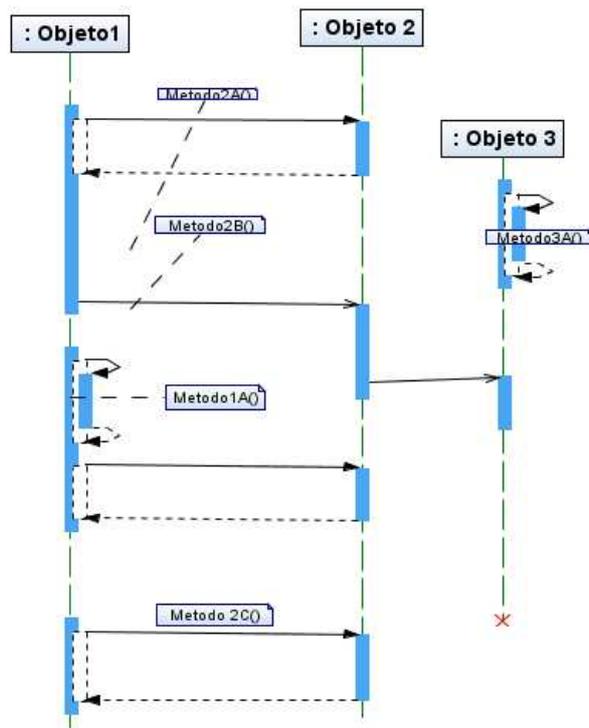


Figura 2.20. Ejemplo de diagrama de secuencia



# CAPÍTULO 3: HERRAMIENTAS PARA LA ELABORACIÓN DEL PROYECTO

Se describen en este capítulo las herramientas utilizadas agrupadas en tres secciones: en primer lugar se describe la infraestructura del servidor elegido para el despliegue del servicio; a continuación se tratan las herramientas que se han utilizado para el desarrollo de la aplicación, tecnologías involucradas e IDE (en inglés, Entorno de desarrollo integrado).

## 3.1. Infraestructura de servidor.

### 3.1.1. Servidor Web.

Un servidor Web (WIKIPEDIA,2010) es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados.

Es muy común utilizar el término Servidor Web para identificar tanto al programa como a la máquina en la que se ejecuta el programa. Este programa se está ejecutando continuamente, manteniéndose a la espera de peticiones por parte de un cliente (un navegador o browser) y respondiendo a estas peticiones adecuadamente, mediante el envío de una página web que se mostrará en el navegador que realizó la petición.

#### 3.1.1.1. Apache

Para el desarrollo del proyecto se ha seleccionado Apache como servidor web. Hoy en día es el servidor web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales, el 59.36% de los servidores web utilizan Apache. Es un software de código abierto que funciona sobre cualquier plataforma.



Figura 3.1. Logo del servidor web Apache

Tiene capacidad para servir páginas tanto de contenido estático como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información. Siendo capaz de soportar HTTP 1.1.

Haciendo una breve comparativa con el resto de servidores Web, se puede ver que Apache se encuentra en la primera posición en cuanto a su implantación. Le sigue el servidor web IIS de Microsoft.

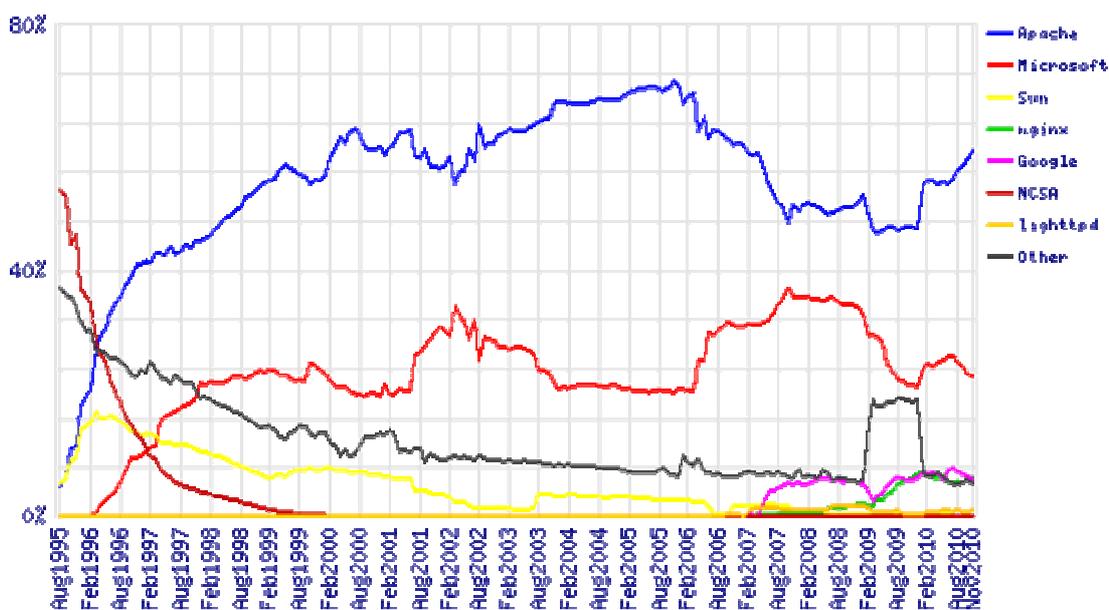


Figura 3.2. Comparativa de servidores Web fuente (NETCRAFT,2010)

### 3.1.2. Sistema Gestor de Bases de Datos.

Los sistemas de gestión de base de datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de



datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos.

Existen distintos objetivos que deben cumplir los SGBD:

**Abstracción de la información** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

**Independencia** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

**Consistencia** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.

**Seguridad** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

**Integridad** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. Los SGBD



proveen mecanismos para garantizar la recuperación de la base de datos hasta un estado consistente (ver Consistencia, más arriba) conocido de forma automática.

**Respaldo** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

**Control de la concurrencia** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

**Manejo de Transacciones** Una Transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que el estado después de una ejecución en la que se produce un error es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

**Tiempo de respuesta** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

### 3.1.2.1. MySQL

MySQL es un sistema de gestión para bases de datos relacionales RDBMS que proporciona una solución fiable y robusta a los usuarios con poderosas herramientas multi-usuario, soluciones de base de datos SQL multi-threaded.

- Ventajas:
  - Es rápida. El objetivo principal en el desarrollo de MySQL fue la velocidad, lo que lleva a un mayor rendimiento.



- Es barata. MySQL está desarrollada bajo política de código abierto con licencia GPL.
  - Es fácil de usar. Se puede interactuar con una base de datos MySQL usando declaraciones simples en lenguaje SQL, lenguaje estándar para la comunicación con una base de datos de tipo RDBMS.
  - Funciona en distintos sistemas operativos. MySQL funciona sobre una amplia variedad de sistemas operativos - Windows, Linux, Mac OS, en la mayor parte de las variantes de UNIX (incluyendo Solaris, AIX, y DEC UNIX), FreeBSD, OS/2, Irix,....
  - El soporte técnico es accesible. La gran cantidad de usuarios que utilizan MySQL garantiza el acceso a la información necesaria a través de foros. También se puede acceder al soporte técnico a través de MySQL AB a precios razonables. Además existen gran cantidad de manuales en la red con los que se puede aclarar cualquier tipo de duda.
  - Es segura. MySQL tiene un sistema flexible para la el control de acceso, la gestión de los privilegios de los usuarios. Se puede especificar en todo momento que privilegios tiene cada usuario o grupos de usuarios. Además las contraseñas están cifradas.
  - Soporta bases de datos grandes. MySQL gestiona bases de datos de hasta 50 millones de entradas o más. El límite de tamaño del archivo de una Tabla es 4GB, pero se puede incrementar, siempre que sea soportado por el sistema operativo, hasta un límite teórico de 8 millones de Terabytes (TB).
  - Es adaptable. La licencia de software libre GPL permite modificar el código de MySQL para adaptar la base de datos a entornos específicos.
- Inconvenientes:
- MySQL no soporta transacciones, "roll-backs" ni subselects.

- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.



**Figura 3.3. Logo de MySQL**

Para la gestión y administración de la base de datos se ha utilizado las MYSQL GUI Tools que incluye el mySQL Administrador y el mySQL Query Browser.

### **3.1.3. Páginas web dinámicas**

Las páginas dinámicas del servidor, son reconocidas, interpretadas y ejecutadas por el propio servidor.

Es importante destacar que las páginas dinámicas de servidor son necesarias ya que para realizar la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos al ordenador del cliente, principalmente bases de datos alojadas en servidores y otros recursos web.

Las ventajas de este tipo de programación son:

- transparente para cliente ya que no ve los scripts que se ejecutan remotamente en el servidor y generan una salida HTML interpretada por el navegador del cliente
- independiente del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable.

Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas estáticas HTML. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno.



**Figura 3.4. Ejemplo generación dinámica de páginas html**

Para permitir mostrar información dinámica en nuestra aplicación web utilizaremos un lenguaje de scripting que nos permita generar dinámicamente las páginas web en base al contenido de nuestra base de datos o información extraída de otros sistemas.

El lenguaje de scripting seleccionado para la realización del proyecto será PHP (PHP: Hypertext Preprocessor)

### 3.1.3.1. PHP

Según (PHP GROUP,2010) se define PHP como *“un lenguaje de scripting, el cual puede ser embebido dentro de páginas HTML. Gran parte de su sintaxis fue tomada de C, Java y Perl con un par de características específicas propias de PHP. El objetivo del lenguaje es permitir a Desarrolladores Web escribir páginas generadas dinámicamente con rapidez.”*



**Figura 3.5. Logo de PHP**

Los motivos por los que se ha elegido PHP para el desarrollo del proyecto son:

- Es un lenguaje de programación del lado del servidor.



- Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor Web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.
- Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones web actuales.
- Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.
- Es modulable y hay multitud de módulos disponibles que realizan multitud de funciones desde envío de correos electrónicos a generación dinámica de imágenes
- Compatibilidad con las bases de datos más comunes, como MySQL, Oracle, Informix, SQLite, ODBC, etc...

```
<html>
  <head>
    <title>Ejemplo PHP</title>
  </head>
  <body>
    <?php echo '<p>Hola Mundo</p>'; ?>
  </body>
</html>
```

**Figura 3.6. Ejemplo código fuente PHP**



### 3.1.3.2. Javascript

JavaScript es un lenguaje de programación desarrollado por Netscape Corporation para su navegador Netscape Navigator 2.0, para permitir la ejecución de código dentro de las páginas en HTML.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

El lenguaje JavaScript imita a Java, pero sin verificación estática de tipos ni restricciones fuertes en tiempo de ejecución.

Las principales características de JavaScript son:

- Interpretado en el cliente.
- Basado en objetos. El código usa objetos incorporados en el sistema, pero no provee creación de clases o herencia
- Código embebido en HTML.
- Variables y tipos de main no se declaran.
- Enlazado dinámico. Las referencias a objetos se validan en tiempo de ejecución.
- No puede escribir archivos en disco

### 3.1.3.3. CSS

CSS, es una tecnología que permite crear páginas web de una manera más exacta, intentando separar en un documento el estilo lógico (estructura) del estilo físico (presentación).

La finalidad de las hojas de estilo es crear unos estilos físicos, separados de las etiquetas HTML (en lugar de como parámetros de las etiquetas), y aplicarlos en los



bloques de texto en los que se quieran aplicar. Estos estilos se pueden modificar además desde JavaScript.

El estilo lógico se refiere a la lógica del documento: cabeceras, párrafos,... no se preocupa de la apariencia final, sino de la estructura del documento. Por el contrario, el estilo físico no se preocupa de la estructura del documento, sino por la apariencia final: párrafos con un cierto tipo de letra, tablas con un determinado color de fondo...

## **3.2. Entorno de trabajo**

Se ha utilizado un entorno de trabajo bajo el sistema operativo Microsoft Windows XP.

### **3.2.1. Eclipse**

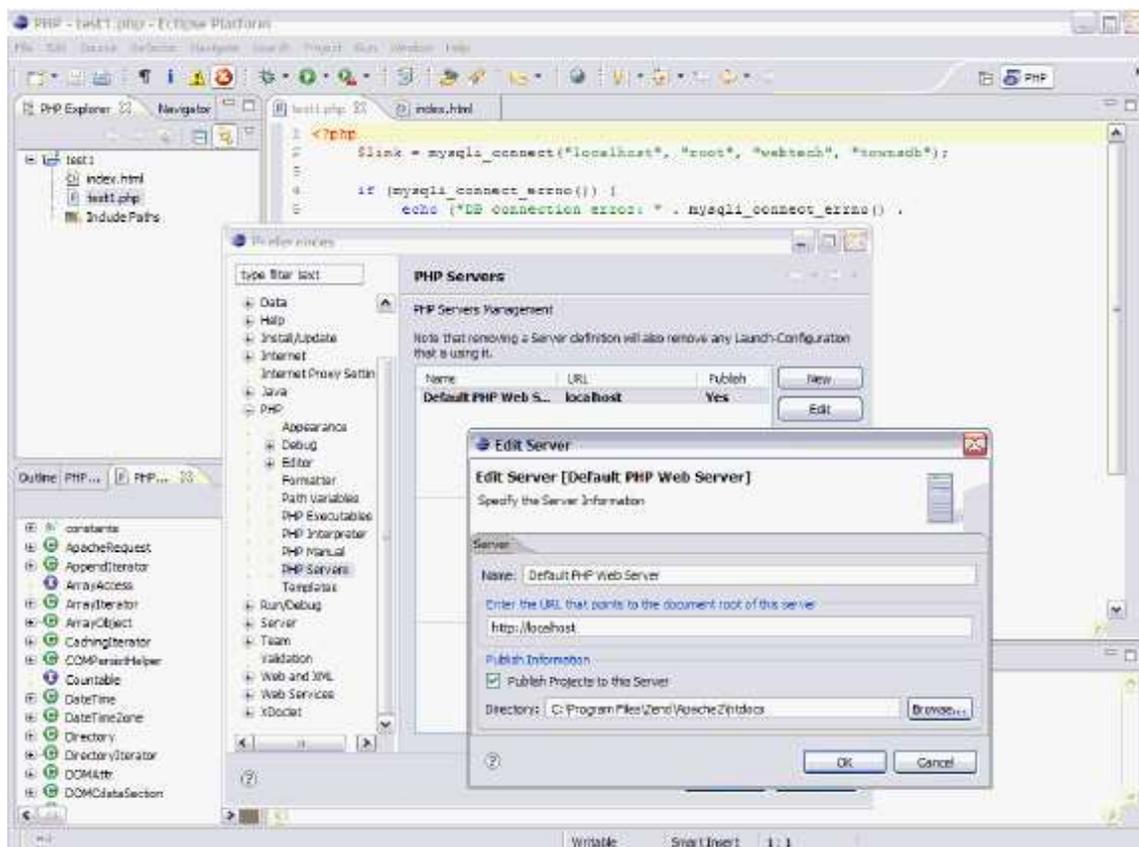
Eclipse (ECLIPSE FOUNDATION,2010) es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

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

Como el lenguaje utilizado para el desarrollo será PHP se utilizará el PHP Development Tools framework for the Eclipse (ECLIPSE FOUNDATION,2010), que es una caracterización de Eclipse para permitir el desarrollo de aplicaciones sobre PHP con los siguientes principios:

- Intuitivo y fácil aprendizaje
- Integración sin problemas con otros proyectos de Eclipse
- Sigue los estándares de Eclipse
- Extensibilidad

- Soporte continuo a futuras versiones de PHP



**Figura 3.7. Captura del IDE Eclipse PDT**

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesamiento de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos.

La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

La versión actual de Eclipse dispone de las siguientes características: editor de texto, resaltado de sintaxis, compilación en tiempo real, pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para la creación

de proyectos, clases, tests, etc., refactorización de clases, métodos, etc., gestión de plugins y muchas otras.

### 3.3. Librerías

Una librería contiene funciones o rutinas que una aplicación puede invocar. Para el desarrollo del proyecto se ha utilizado un motor de plantillas para PHP llamado Smarty (SMARTY,2010).



Figura 3.8. Logo de Smarty

#### 3.3.1. Smarty

Smarty es un motor de plantillas para PHP que permite obtener los siguientes beneficios:

- Separación del plano de aplicación del plano de presentación
- Facilidad en la gestión, administración y modificación de las plantillas
- Permite modificar múltiples páginas web simultáneamente que usen la misma plantilla
- Mayor seguridad al separar la parte que ejecuta código PHP de la parte de presentación
- Se reduce el uso de delimitadores  $\langle \rangle$  que son compartidos entre PHP y HTML reduciendo los posibles casos de confusión.



| PHP  |
|--|
| <pre>&lt;a href="&lt;?php echo \$bar['zig']; ?&gt;"&gt;&lt;?php echo \$bar['zag']; ?&gt;&lt;/a&gt; &lt;a href="&lt;?php echo \$bar['zig2']; ?&gt;"&gt;&lt;?php echo \$bar['zag2']; ?&gt;&lt;/a&gt; &lt;a href="&lt;?php echo \$bar['zig3']; ?&gt;"&gt;&lt;?php echo \$bar['zag3']; ?&gt;&lt;/a&gt;</pre> |
| Smarty   |
| <pre>&lt;a href="{ \$bar.zig }"&gt;{ \$bar.zag }&lt;/a&gt; &lt;a href="{ \$bar.zig2 }"&gt;{ \$bar.zag2 }&lt;/a&gt; &lt;a href="{ \$bar.zig3 }"&gt;{ \$bar.zag3 }&lt;/a&gt;</pre>   |

Figura 3.9. Ejemplo de código utilizando Smarty



# CAPÍTULO 4: DESARROLLO DEL PROYECTO

Este proyecto es un desarrollo a medida en base a las necesidades de creación de una aplicación web que sirva como ejemplo de las capacidades de la plataforma de localización de un operador móvil.

Sus principales objetivos es que esté orientado al usuario, buscando la máxima usabilidad y sencillez que permita reducir la curva de aprendizaje para desenvolverse con comodidad en ella.

## 4.1. Fase inicial

El principal propósito de esta sección es reflejar las necesidades de crear un aplicación que mostrase las posibilidades de gestión de una flota en base a la capacidad que presentan las plataformas del operador. Para ello, esta aplicación y su entorno deben contemplar las siguientes capacidades:

**Usabilidad** La aplicación debe reducir al mínimo el esfuerzo del usuario a la hora de llevar a cabo sus tareas, presentando una interfaz clara y sencilla.

**Fiabilidad** Las herramientas escogidas para el servidor en producción deberán ser estables y confiables para garantizar el buen funcionamiento en todo momento.

**Escalabilidad** En un principio no se prevé un gran crecimiento de los datos de la aplicación pero se pensará en una solución que no nos ate a un Sistema Gestor de Base de Datos para, en caso de ser necesario, poder ampliar la funcionalidad a otros de mayor capacidad sin efectuar cambios en el código de la aplicación.

**Innovación** Se utilizarán todas las herramientas posibles para que el desarrollo de la aplicación se adecue a los estándares de la Web 2.0, teniendo especial atención en la información proporcionada al usuario.

Seguidamente se comentarán las distintas fases del proyecto, describiendo detalladamente el desarrollo y los productos finales de cada una de ellas.



Por último, se dará una visión del producto final puesto en producción, presentando los resultados del trabajo, una descripción de la interfaz de usuario y un resumen de costes.

#### 4.1.1. Proceso de desarrollo

Debido a la naturaleza del proyecto y dado que los requerimientos de usuario fueron bien especificados desde un principio se ha optado por un modelo de Ciclo de Vida en Cascada, con el objetivo de delimitar claramente cada fase de desarrollo.

Como es conocido, el inicio de cada etapa de este ciclo de vida debe esperar a la finalización de la etapa inmediatamente anterior. Estas etapas o fases son :

**Análisis de requisitos** Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos. Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema.

**Diseño del sistema** Se descompone y organiza el sistema en elementos que puedan elaborarse por separado. El resultado es la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

**Diseño del programa** Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de codificación.

**Codificación** Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las librerías y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

**Pruebas** Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación.



**Implantación** El software obtenido se pone en producción. La implantación es la fase con más duración y con más cambios en el ciclo de elaboración de un proyecto. Durante la explotación del sistema software pueden surgir cambios, bien para corregir errores o bien para introducir mejoras.

#### **4.1.2. Especificación de requisitos de Usuario**

A continuación se definen los requisitos que se recogieron de manera informal en las primeras reuniones con el objetivo de evaluar la viabilidad de la aplicación Web a desarrollar así como de realizar una estimación de costes inicial.

##### *Especificaciones Informales*

El objetivo principal de la aplicación deberá ser el permitir gestionar una flota de vehículos una empresa y permitir la localización de cada uno de los individuos en base a las capacidades de la plataforma de localización del operador. Para ello se definen distintos roles de usuario, un administrador y el perfil de usuario que a su vez gestiona su propia flota de vehículos.

La aplicación permitirá al usuario dar de alta/baja/modificar los individuos de la flota que gestiona, así como realizar localización de estos y consultar el histórico de localizaciones.

Al usuario administrador debe permitirle dar de alta/baja usuarios, así como modificarlos, realizar localizaciones de flotas, consultar el histórico de localizaciones y modificar los datos de los usuarios.

Además el usuario administrador debe poder acceder a un módulo de funcionalidades avanzadas que le permita enviar primitivas en modo debug contra la plataforma para poder depurar cualquier problema con la aplicación.

En cuanto a requisitos de tipo no funcional, se establece que la aplicación Web sea desarrollada con una interfaz sencilla y orientada al usuario, permitiendo su instalación en cualquier sistema operativo

### 4.1.3. Diagrama de casos de uso inicial

Definidos los objetivos y requisitos del proyecto se inicia el proceso de análisis del sistema correspondiente a la fase actual. En la primera iteración del proceso se estudia el sistema como un todo y se determinan sus límites y contenido.

La aplicación Web será utilizada por usuarios que podrán tener dos tipos de roles. El rol más permisivo es el rol de Administrador, que tendrá acceso a todas las funcionalidades de la aplicación así como al módulo de funcionalidades avanzadas. Existirá también el rol de usuario normal que permitirá gestionar una flota específica y realizar localizaciones sobre los individuos de su flota. Este rol no tendrá acceso al módulo de funcionalidades avanzadas.

Los requisitos funcionales del sistema se muestran a continuación mediante diagramas de casos de uso. Al encontrarnos en la fase inicial, se muestra el diagrama de casos de uso para un usuario genérico, sin tener en cuenta el rol de dicho usuario dentro de la aplicación.

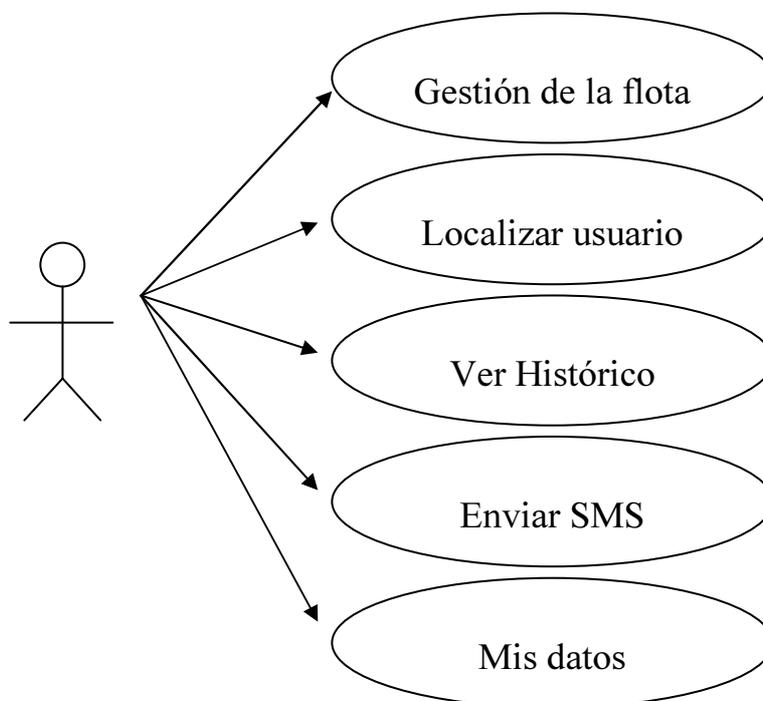


Figura 4.1. Casos de uso iniciales



A continuación se ofrece una breve descripción de los casos de uso mostrados en el diagrama anterior.

**Gestión de la flota** Permite crear nuevos individuos de la flota, consultar, cancelar o modificar alguno de los existentes

**Localizar usuario** Permite localizar un usuario de los existentes en la flota del usuario

**Ver Histórico** Se podrá consultar las localizaciones realizadas y el histórico de ubicaciones

**Enviar SMS** Permite enviar un SMS a la lista de usuarios definida de la flota

**Mis datos** Permite cambiar la clave de acceso del usuario actual.

## 4.2. Fase de Análisis

Se presenta en esta fase del proyecto una visión general del proceso de análisis, paso previo a la implementación de la aplicación. Como es natural, los requisitos expuestos en la Fase Inicial distan considerablemente de los que se presentan a continuación. De igual manera, se presenta un diagrama de casos de uso mucho más completo, incluyendo también la jerarquía definitiva de usuarios.

### 4.2.1. Especificación de requisitos en la Fase de Análisis.

A continuación se enumeran los requisitos que la aplicación debe cumplir generados a partir del diagrama de casos de uso.



| <b>RU-000001: Gestión de Usuarios</b> |   |             |                      |
|---------------------------------------|---|-------------|----------------------|
| Author:                               |   | Kind:       | Requisito de usuario |
| Version:                              | 1.0   | State:      | Proposed             |
| Priority:                             | Medium  | Complexity: | Medium               |
| Stability:                            | Normal  | Verifiable: | True                 |
| Goal:                                 | Permitirá la gestión de los usuarios del sistema.                                     |             |                      |
| Related Requirements:                 | RU-000004: Administración<br>RU-000005: Alta de Usuario<br>RU-000006: Baja de usuario |             |                      |
| Description:                          | Permitirá la gestión de los usuarios del sistema.                                     |             |                      |

**Tabla 4.1. RU-000001**

| <b>RU-000002: Gestión de Flota</b> |  |             |                      |
|------------------------------------|--|-------------|----------------------|
| Author:                            |  | Kind:       | Requisito de usuario |
| Version:                           | 1.0  | State:      | Proposed             |
| Priority:                          | Medium   | Complexity: | Medium               |
| Stability:                         | Normal   | Verifiable: | True                 |
| Goal:                              | Permitirá la gestión de la flota   |             |                      |
| Related Requirements:              | RU-000004: Administración<br>RU-000007: Alta de Flota<br>RU-000008: Desactivar Flota |             |                      |
| Description:                       | Permitirá la gestión de los individuos de la flota                                   |             |                      |

**Tabla 4.2. RU-000002**

| <b>RU-000003: Ver registros</b> |  |             |                      |
|---------------------------------|--|-------------|----------------------|
| Author:                         |  | Kind:       | Requisito de usuario |
| Version:                        | 1.0  | State:      | Proposed             |
| Priority:                       | Medium   | Complexity: | Medium               |
| Stability:                      | Normal   | Verifiable: | True                 |
| Goal:                           | Permitirá la gestión de emails de confirmación de recepción. |             |                      |
| Related Requirements:           | RU-000004: Administración                                    |             |                      |
| Description:                    | Permitirá la consulta del histórico de acciones              |             |                      |

**Tabla 4.3. RU-000003**



| <b>RU-000004: Administración</b> |  |             |                      |
|----------------------------------|--|-------------|----------------------|
| Author:                          |  | Kind:       | Requisito de usuario |
| Version:                         | 1.0  | State:      | Proposed             |
| Priority:                        | Medium   | Complexity: | Medium               |
| Stability:                       | Normal   | Verifiable: | True                 |
| Goal:                            | Permitirá la administración tanto de usuarios como de las flotas                           |             |                      |
| Related Requirements:            | RU-000001: Gestión de Usuarios.<br>RU-000002: Gestión de Flota<br>RU-000003: Ver registros |             |                      |
| Description:                     | Permitirá la administración tanto de los maestros como de las flotas                       |             |                      |

Tabla 4.4. RU-000004

| <b>RU-000005: Alta Usuario</b> |   |             |                      |
|--------------------------------|---|-------------|----------------------|
| Author:                        |   | Kind:       | Requisito de usuario |
| Version:                       | 1.0   | State:      | Proposed             |
| Priority:                      | Medium  | Complexity: | Medium               |
| Stability:                     | Normal  | Verifiable: | True                 |
| Goal:                          | Permite dar de alta un Usuario.                   |             |                      |
| Normal Flow:                   | Introducir login válido.<br>Guardar nuevos datos. |             |                      |
| Related Requirements:          | RU-000001: Gestión de Usuarios.                   |             |                      |
| Description:                   | Permite dar de alta un Usuario.                   |             |                      |

Tabla 4.5. RU-000005

| <b>RU-000006: Baja Usuario</b> |   |             |                      |
|--------------------------------|---|-------------|----------------------|
| Author:                        |   | Kind:       | Requisito de usuario |
| Version:                       | 1.0   | State:      | Proposed             |
| Priority:                      | Medium  | Complexity: | Medium               |
| Stability:                     | Normal  | Verifiable: | True                 |
| Goal:                          | Permite dar de baja un usuario                    |             |                      |
| Normal Flow:                   | Seleccionar usuario a borrar<br>Borrar usuario/s. |             |                      |
| Related Requirements:          | RU-000001: Gestión de Usuarios.                   |             |                      |
| Description:                   | Permite dar de baja un usuario                    |             |                      |

Tabla 4.6. RU-000006



| <b>RU-000007: Alta Flota</b> |   |             |                      |
|------------------------------|---|-------------|----------------------|
| Author:                      |   | Kind:       | Requisito de usuario |
| Version:                     | 1.0   | State:      | Proposed             |
| Priority:                    | Medium  | Complexity: | Medium               |
| Stability:                   | Normal  | Verifiable: | True                 |
| Goal:                        | Permite dar de alta un individuo de la flota  |             |                      |
| Normal Flow:                 | Introducir datos del individuo (nombre, teléfono, y habilitar localización).<br>Guardar nuevos datos. |             |                      |
| Related Requirements:        | RU-000002: Gestión de Flota   |             |                      |
| Description:                 | Permite dar de alta un individuo en la flota.   |             |                      |

Tabla 4.7. RU-000007

| <b>RU-000008: Desactivar flota</b> |   |             |                      |
|------------------------------------|---|-------------|----------------------|
| Author:                            |   | Kind:       | Requisito de usuario |
| Version:                           | 1.0   | State:      | Proposed             |
| Priority:                          | Medium  | Complexity: | Medium               |
| Stability:                         | Normal  | Verifiable: | True                 |
| Goal:                              | Permite desactivar la localización de una flota                               |             |                      |
| Normal Flow:                       | Seleccionar individuo, desactivar la opción de localizable<br>Guardar cambios |             |                      |
| Related Requirements:              | RU-000002: Gestión de Flota   |             |                      |
| Description:                       | Permite desactivar la localización de una flota                               |             |                      |

Tabla 4.8. RU-000008

| <b>RU-000009: Gestión avanzada GPP</b> |   |             |                      |
|--|---|-------------|----------------------|
| Author:                                |   | Kind:       | Requisito de usuario |
| Version:                               | 1.0   | State:      | Proposed             |
| Priority:                              | Medium  | Complexity: | Medium               |
| Stability:                             | Normal  | Verifiable: | True                 |
| Goal:                                  | Acceso a las funciones de la API de la GPP para detectar y solucionar errores   |             |                      |
| Related Requirements:                  | RU-000004: Administración   |             |                      |
| Description:                           | El administrador podrá acceder a la API de funciones de la GPP para consultar o modificar la configuración en la plataforma de localización |             |                      |

Tabla 4.9. RU-000009



| <b>RU-000010: Localizar usuario</b> |   |             |                      |
|-------------------------------------|---|-------------|----------------------|
| Author:                             |   | Kind:       | Requisito de usuario |
| Version:                            | 1.0   | State:      | Proposed             |
| Priority:                           | Medium  | Complexity: | Medium               |
| Stability:                          | Normal  | Verifiable: | True                 |
| Goal:                               | Permitir localizar a un usuario de la flota   |             |                      |
| Related Requirements:               | RU-000002: Gestión de Flota   |             |                      |
| Description:                        | El usuario podrá localizar a alguno de los individuos de la flota que tenga activada la localización. |             |                      |

**Tabla 4.10. RU-000010**

| <b>RU-000011: Cambiar contraseña</b> |   |             |                      |
|--------------------------------------|---|-------------|----------------------|
| Author:                              |   | Kind:       | Requisito de usuario |
| Version:                             | 1.0   | State:      | Proposed             |
| Priority:                            | Medium  | Complexity: | Medium               |
| Stability:                           | Normal  | Verifiable: | True                 |
| Goal:                                | Permitirá el cambio de contraseña del usuario y administrador |             |                      |
| Related Requirements:                | RU-000001: Gestión de Usuarios<br>RU-000004: Administración   |             |                      |
| Description:                         | Permitirá el cambio de contraseña del usuario y administrador |             |                      |

**Tabla 4.11. RU-000011**

| <b>RU-000012: Salida del sistema</b> |  |             |                      |
|--------------------------------------|--|-------------|----------------------|
| Author:                              |  | Kind:       | Requisito de usuario |
| Version:                             | 1.0  | State:      | Proposed             |
| Priority:                            | Medium   | Complexity: | Medium               |
| Stability:                           | Normal   | Verifiable: | True                 |
| Goal:                                | Permitirá salir del sistema cerrando la sesión |             |                      |
| Description:                         | Permitirá salir del sistema cerrando la sesión |             |                      |

**Tabla 4.12. RU-000012**

### 4.2.2. Diagrama de casos de uso en la Fase de Análisis.

Los casos de uso representados para el Sistema en la Fase de Análisis son:

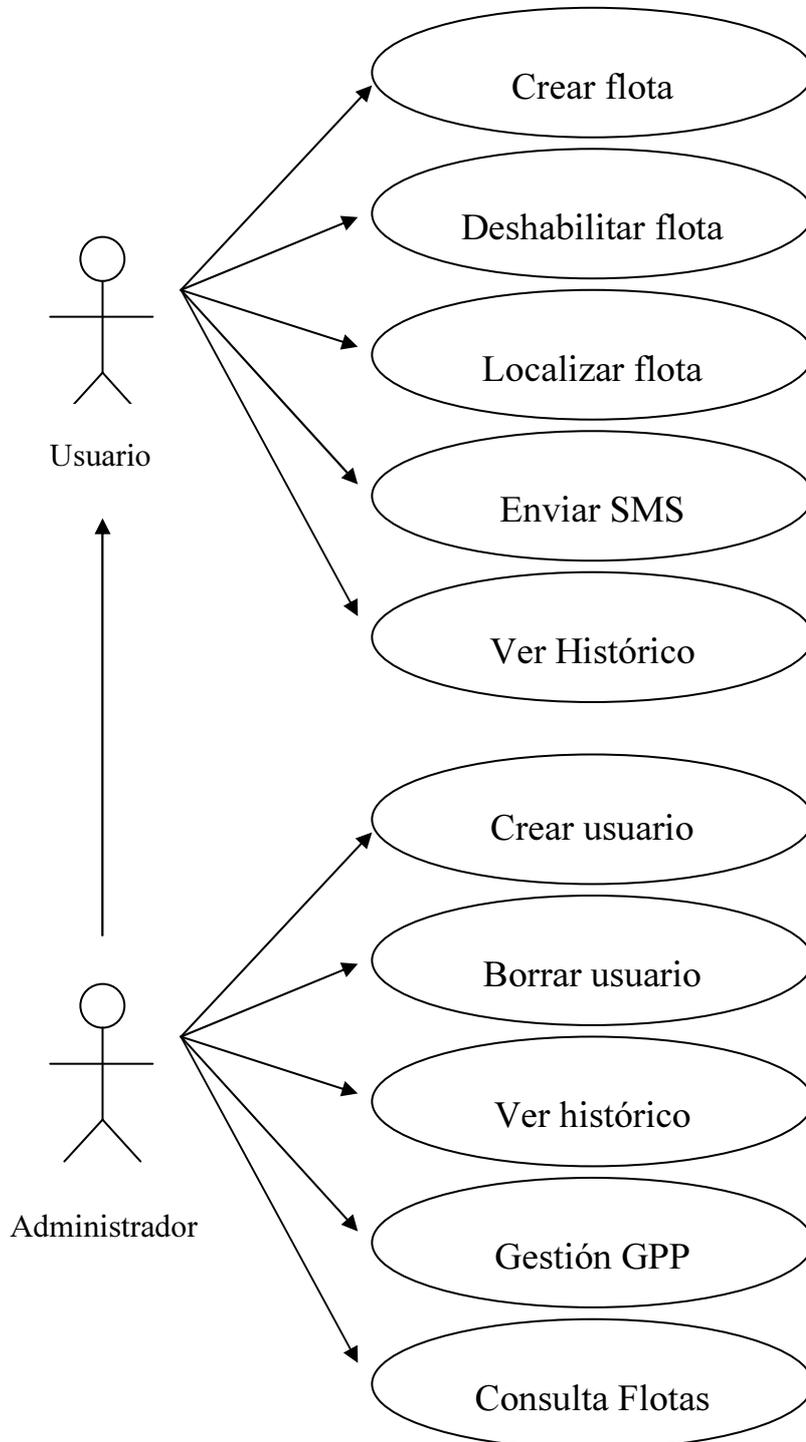


Figura 4.2. Casos de uso en la Fase de Análisis



Se ofrece a continuación una descripción en profundidad de los casos de uso representados en la figura anterior. Para cada caso de uso se ofrece una descripción del mismo, los actores que están involucrados en él, las precondiciones (estado del sistema que se tiene que cumplir para que el caso de uso se pueda instanciar), las postcondiciones (estado del sistema una vez instanciado el caso de uso) y el escenario básico (pasos principales del caso de uso ordenados).

| Nombre          | Crear flota.   |
|-----------------|--|
| Descripción     | El sistema ha de permitir el incluir un nuevo individuo dentro de la flota que se gestiona   |
| Actores         | Usuario, Administrador   |
| Precondiciones  | Usuario registrado en la aplicación.   |
| Postcondiciones | Flota creada   |
| Escenario       | Hacer login en la aplicación.<br>Seleccionar Gestión mi Flota<br>Seleccionar Añadir nuevo equipo<br>Introducir los datos<br>Y confirmar la inserción de la nueva flota |

**Tabla 4.13. CU-001**

| Nombre          | Deshabilitar flota  |
|-----------------|---|
| Descripción     | El sistema deberá permitir deshabilitar la localización de un equipo de la flota para que no se permite localizarlo                             |
| Actores         | Usuario, Administrador  |
| Precondiciones  | Usuario registrado en el sistema.   |
| Postcondiciones | Flota modificada  |
| Escenario       | Hacer login en la aplicación.<br>Seleccionar Gestión mi Flota<br>Seleccionar el equipo a deshabilitar<br>Confirmar la modificación de los datos |

**Tabla 4.14. CU-002**

| Nombre          | Localizar flota  |
|-----------------|--|
| Descripción     | El sistema deberá permitir realizar consultas de localización de los equipos con el flag de localizables de la flota del usuario         |
| Actores         | Usuario  |
| Precondiciones  | Usuario registrado   |
| Postcondiciones | Se modifica el histórico de eventos  |
| Escenario       | Hacer login en la aplicación.<br>Seleccionar Localizar<br>Seleccionar el equipo a localizar del desplegable<br>Confirmar la localización |

**Tabla 4.15. CU-003**



| Nombre          | Enviar SMS   |
|-----------------|--|
| Descripción     | El sistema deberá permitir enviar SMS a cualquier de los equipos de la flota activos   |
| Actores         | Usuario  |
| Precondiciones  | Usuario  |
| Postcondiciones | El sistema permanece inalterado.   |
| Escenario       | Hacer login en la aplicación.<br>Seleccionar Envío SMS<br>Seleccionar el destinatario del desplegable e introducir el texto<br>Confirmar el envío del mensaje. |

Tabla 4.16. CU-004

| Nombre          | Ver histórico   |
|-----------------|---|
| Descripción     | El sistema debe permitir consultar el histórico de localizaciones que ha realizado el usuario |
| Actores         | Usuario, Administrador.   |
| Precondiciones  | Usuario   |
| Postcondiciones | El sistema permanece inalterado.  |
| Escenario       | Hacer login en la aplicación.<br>Seleccionar Ver histórico                                    |

Tabla 4.17. CU-005

| Nombre          | Crear usuario  |
|-----------------|--|
| Descripción     | El sistema deberá permitir añadir usuarios para gestionar la flota   |
| Actores         | Administrador.   |
| Precondiciones  | Usuario registrado como Administrador.   |
| Postcondiciones | Inserción de usuario   |
| Escenario       | Hacer login como Administrador.<br>Seleccionar Gestión de usuarios<br>Elegir Añadir usuario<br>Introducir los datos del usuario<br>Confirmar la inserción. |

Tabla 4.18. CU-006



| Nombre          | Borrado de usuario  |
|-----------------|---|
| Descripción     | El sistema deberá permitir borrar usuarios de la gestión de la flota                                      |
| Actores         | Administrador.  |
| Precondiciones  | Usuario registrado como Administrador.  |
| Postcondiciones | Borrado de usuario de usuario   |
| Escenario       | Hacer login como Administrador.<br>Seleccionar Gestión de usuarios<br>Elegir usuario<br>Confirmar borrado |

Tabla 4.19. CU-007

| Nombre          | Gestión GPP  |
|-----------------|--|
| Descripción     | El sistema deberá permitir al usuario administrador el envío de todas las operaciones soportadas por la plataforma de localización para permitir solucionar cualquier incidencia relacionada con la provisión en la plataforma de localización |
| Actores         | Administrador.   |
| Precondiciones  | Usuario registrado como Administrador.   |
| Postcondiciones | El sistema permanece inalterado  |
| Escenario       | Hacer login como Administrador.<br>Seleccionar Gestión GPP<br>Seleccionar operación y parámetros a enviar<br>Confirmar el envío de la operación contra la plataforma de localización   |

Tabla 4.20. CU-008

### 4.2.3. Modelo de Base de Datos

A continuación se describe el modelo de base de datos utilizado en la aplicación.

El diagrama se compone de tres entidades o tablas, que son: USUARIOS, FLOTAS y CDRS.

En la entidad **USUARIOS** se almacenan los usuarios que tienen acceso a la aplicación en cualquier de sus perfiles (administrador o normal). Tiene una clave principal autonómica para identificar a cada usuario. Se guardan diversos datos del usuario como nombre, alias, password, telefono, email, perfil y el grupo de la GPP al que puede administrar.

La entidad **FLOTAS** contiene todos los elementos localizables por la plataforma. Cada flota tiene una clave principal autonómica y única. Cada individuo de la flota pertenece a un grupo tiene un msisdn que lo identifica para localizarlo y un

grupo de la GPP resultado de su provisión en la GPP. Los individuos de la flota sólo pueden ser localizados por usuarios que tengan asignado el mismo grupo en la entidad USUARIOS. Se define un atributo llamado activo que permite desactivar el localizar a un usuario para que no salga en el desplegable a la hora de localizar a usuarios. Se añade un campo descrip para permitir identificar con una descripción la flota.

La entidad **CDRS** contiene los registros por cada localización que realizan los usuarios. Se almacena un registro con un identificador autonumérico y se referencia a dos claves externas (foreign key) para identificar el usuario que realizó la localización y el individuo localizado. Además se incluyen dos campos con los datos de la localización y la fecha y hora en la que se realizó.

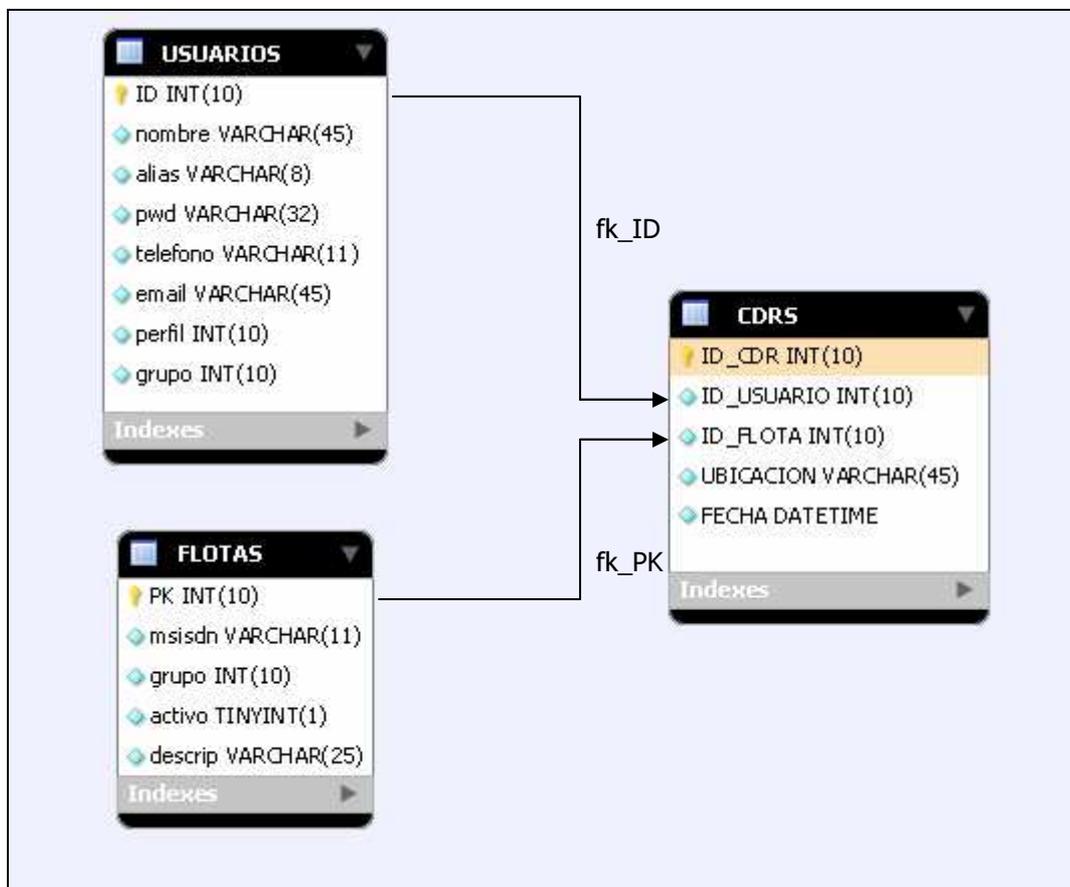


Figura 4.3. Modelo de la Base de Datos

### 4.3. Estudio de la API de localización

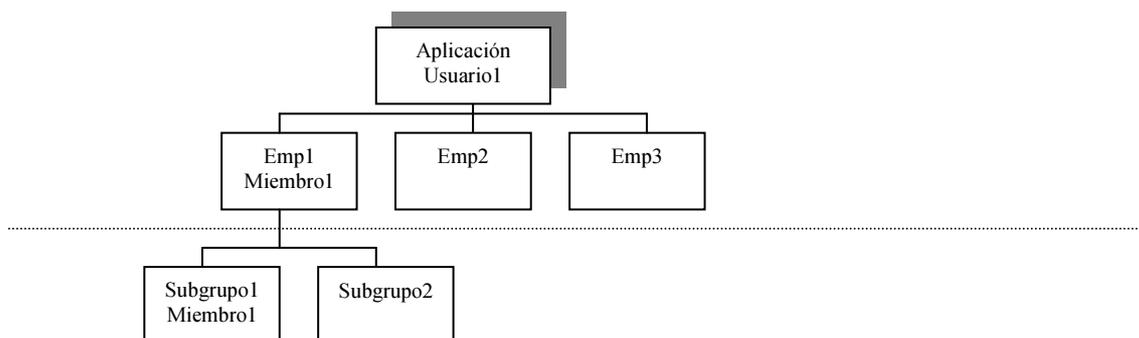
Todo el proyecto se cimienta sobre la API de localización de la plataforma del operador. A continuación se va a realizar un breve análisis de las capacidades y aspectos más importantes que se han tenido en cuenta a la hora de realizar el proyecto.

Las tres clases de objetos que se pueden manejar desde el API son:

- **USUARIO:** Es el objeto que identifica a un usuario final poseedor de un móvil en toda la aplicación.
- **GRUPO:** Es el objeto que identifica las agrupaciones de usuarios. A cada empresa que se le preste servicio le debe corresponder un grupo de primer nivel dentro de la aplicación. Existirá un grupo por defecto que se crea automáticamente con la aplicación cuyo nombre será Gdefecto.
- **MIEMBRO:** Es el objeto que identifica al usuario dentro de un grupo.

Cada objeto posee una serie de campos que lo definen, como su estado de localización, o de validación.

La estructura que ha de seguir la aplicación dentro de TME es como sigue:



**Figura 4.4. Ejemplo de estructura de la provisión en la GPP**



### 4.3.1. Alta de usuarios

El alta de usuarios se efectúa en dos pasos. Primero la aplicación debe pre-registrar al usuario utilizando como identificador su MSISDN y después éste debe validar mediante un mensaje corto este pre-registro autorizando ser localizado:

- El pre-registro por parte del proveedor a su vez se ha de efectuar en dos pasos. Primero se ha de crear un usuario en la aplicación y por último inscribir como miembro de una o varias empresas (grupos) a este usuario. Este proceso se efectuará llamando a dos funciones del API de provisioning, crear usuario y crear miembro. Por supuesto existen campos restringidos a los que el ASP no podrá acceder, como por ejemplo el campo “Validate” que siempre aparecerá inicializado a no para asegurar que el usuario no podrá ser localizado hasta que éste lo autorice.

- Una vez preinscrito el usuario, para que la inscripción se complete, el usuario deberá autorizar su localización enviando un mensaje corto con un determinado formato por cada empresa en la que se inscriba a un número de TME . Es responsabilidad de la aplicación garantizar que el usuario conoce que será localizado, las características del servicio y de cómo éste debe efectuar la validación (mensaje corto que debe enviar y a qué número).

### 4.3.2. Operaciones de provisión

El proyecto soporta las principales operación de provisión de la plataforma. A continuación se muestra un ejemplo de cómo se provisionaría un usuario.

La gestión de grupos y autenticación de usuarios mediante HTTP se realiza a través del operaciones POST. En los datos de POST se debe especificar la variable gpprequest con la petición deseada en formato XML.

En cada petición se especifica el tipo de identificación que se está utilizando para los usuarios. Esto se realiza en el atributo identype del tag PARAMETERS. Una petición viene encapsulada entre los tags <GPPR></GPPR>

La respuesta se devuelve en otro XML que cumple las mismas restricciones que la petición. El método de identificación especificado en la petición será el utilizado también en la respuesta. Status = 0 significa que la operación se ha realizado con éxito.

En primer lugar se crea un grupo para la aplicación con la operación createGroupS. A continuación se muestra un ejemplo del XML que se envía a la plataforma de localización para crear un nuevo grupo:

|  |                                      |
|--|--------------------------------------|
| <pre>- &lt;GPPR&gt;   &lt;FUNCTION type="INSERT"&gt;createGroupS&lt;/FUNCTION&gt;</pre>  | Función                              |
| <pre>- &lt;AUTENTIFICATION&gt;   &lt;CLIENT&gt;curso&lt;/CLIENT&gt;   &lt;CLI_PASSWD&gt;5curso5&lt;/CLI_PASSWD&gt;   &lt;USER_LOGIN&gt;test1&lt;/USER_LOGIN&gt;   &lt;USER_PASSWD&gt;test1&lt;/USER_PASSWD&gt; &lt;/AUTENTIFICATION&gt;</pre>  | Datos de autenticación del usuario   |
| <pre>- &lt;PARAMETERS idtype="MSISDN"&gt; - &lt;ATTRIBUTE name="GROUP"&gt;   - &lt;OBJECT&gt;     &lt;CLASS&gt;GROUP&lt;/CLASS&gt;     - &lt;CONTENTS&gt;       - &lt;ATTRIBUTE name="PARENTGROUPID"&gt;         &lt;INT val="0" /&gt;       &lt;/ATTRIBUTE&gt;       - &lt;ATTRIBUTE name="NAME"&gt;         &lt;STRING val="Curso" /&gt;       &lt;/ATTRIBUTE&gt;     &lt;/CONTENTS&gt;   &lt;/OBJECT&gt; &lt;/ATTRIBUTE&gt; &lt;/PARAMETERS&gt; &lt;/GPPR&gt;</pre> | Datos para la operación createGroupS |

#### Ejemplo 4.1. XML Operación CreateGroupS

La respuesta de la plataforma será del tipo:

```
- <GPPM_IT version="3.5" id="8c7a0122-12506cb3577-0" status="0">
- <ATTRIBUTE name="IDGROUP">
  <INT val="1543" />
</ATTRIBUTE>
</GPPM_IT>
```

#### Ejemplo 4.2. XML Resultado CreateGroupS

Donde el status="0" nos indica que la operación se ha realizado con éxito y el valor del grupo creado (1543).

A continuación se muestra la operación de provisión de un usuario:

|  |                                    |
|--|------------------------------------|
| <pre>- &lt;GPPR&gt;   &lt;FUNCTION type="INSERT"&gt;createUser&lt;/FUNCTION&gt;</pre>  | Función                            |
| <pre>- &lt;AUTH&gt;   &lt;CLIENT&gt;curso&lt;/CLIENT&gt;   &lt;CLI_PASSWD&gt;5curso5&lt;/CLI_PASSWD&gt;   &lt;USER_LOGIN&gt;abcd&lt;/USER_LOGIN&gt;   &lt;USER_PASSWD&gt;dcba&lt;/USER_PASSWD&gt; &lt;/AUTH&gt;</pre>  | Datos de autenticación del usuario |
| <pre>- &lt;PARAMETERS identype="MSISDN"&gt;   - &lt;ATTRIBUTE name="USER"&gt;     - &lt;OBJECT&gt;       &lt;CLASS&gt;USER&lt;/CLASS&gt;       - &lt;CONTENTS&gt;         - &lt;ATTRIBUTE name="MSISDN"&gt;           &lt;STRING val="34680100026" /&gt;         &lt;/ATTRIBUTE&gt;       &lt;/CONTENTS&gt;     &lt;/OBJECT&gt;   &lt;/ATTRIBUTE&gt; &lt;/PARAMETERS&gt; &lt;/GPPR&gt;</pre> | Datos para la operación createUser |

### Ejemplo 4.3. XML Operación CreateUser

El resultado satisfactorio por parte de la plataforma se muestra a continuación:

```
- <GPPM_IT version="3.5" id="8c7a0122-12080e104e3-0" status="0">
  - <ATTRIBUTE name="USERID">
    <INT val="34680100026" />
  </ATTRIBUTE>
</GPPM_IT>
```

### Ejemplo 4.4. XML Resultado CreateUser

Ahora será necesario insertar el usuario anterior dentro del grupo anteriormente creado:



```

- <GPPR>
  <FUNCION type="INSERT">createMemberRec</FUNCION>
- <AUTENTIFICATION>
  <CLIENT>curso</CLIENT>
  <CLI_PASSWD>5curso5</CLI_PASSWD>
  <USER_LOGIN>test1</USER_LOGIN>
  <USER_PASSWD>test1</USER_PASSWD>
</AUTENTIFICATION>
- <PARAMETERS identype="MSISDN">
- <ATTRIBUTE name="MEMBER" param="Y">
  - <OBJECT>
    <CLASS>MEMBER</CLASS>
  - <CONTENTS>
    - <ATTRIBUTE name="GROUPID">
      <INT val="1543" />
    </ATTRIBUTE>
    - <ATTRIBUTE name="USERID">
      <STRING val="34699071728" />
    </ATTRIBUTE>
    - <ATTRIBUTE name="TIMEINIT">
      <STRING val="00:00:00" />
    </ATTRIBUTE>
    - <ATTRIBUTE name="TIMEEND">
      <STRING val="23:59:59" />
    </ATTRIBUTE>
  </CONTENTS>
  </OBJECT>
</ATTRIBUTE>
</PARAMETERS>
</GPPR>

```

Función

Datos de  
autenticación  
del usuarioDatos para la  
operación  
createMember  
Rec  
especificando  
el grupo en el  
que lo  
insertamos

#### Ejemplo 4.5. XML Operación CreateMemberRec

Y el resultado de la inserción en el grupo:

```
<GPPM_IT version="3.5" id="8c7a0122-120aaaf3379-0" status="0" />
```

#### Ejemplo 4.6. XML Resultado CreateMemberRec

Una vez pre-registrado el usuario en la aplicación sólo queda que de su consentimiento enviando un SMS con el texto ALTA nombreGRUPO al 2334. Mediante mensaje corto también se puede:

- Darse de alta/ baja



- Activar/Desactivar su localización global o en una aplicación
- Cambiar horario de localización en una aplicación

### 4.3.3. Invocando a la API de localización

La localización de usuario se hace mediante la invocación a la operación `m_findIT` de la plataforma. Se tendrán en cuenta las siguientes consideraciones:

- Como identificador de los usuarios a localizar se utilizará el MSISDN
- Es necesario que el usuario exista como miembro de un grupo para que se le pueda localizar. Por lo tanto si se quiere localizar a un usuario o a un grupo es siempre necesario incluir el `GroupId` en la petición.
- El esquema sintáctico para la invocación es mediante una operación `http GET` y queda de la siguiente forma:

**`http://<host>:<port>/<gpp_home>/WLServer?<parámetros>`**

#### Ejemplo 4.7. Formato URL Plataforma de localización

Donde

- `<host>` identifica la IP del servidor
- `<port>` es el puerto donde reside el servicio
- `<gpp_home>` la ruta en donde reside el `WLServer`
- `<parámetros>` la concatenación de los siguientes parámetros mediante ‘&’

- Autenticación y operación `M_FINDIT`

**`CLIENT=client_name&CLI_PASSWD=client_password&USER=user_name&USER_PASSWD=user_password&Method=M_FINDIT`**

- Grupo; Identificador del grupo sobre el que se desea ejecutar la búsqueda



**GROUP=group\_id**

- Usuarios a localizar

**TUSERID=MSISDN&TUSERID\_TYPE=MSISDN**

- Opciones de salida
  - SRS=EPSG:4326 ; Identificador del sistema espacial de referencia en el que se desea la respuesta. La plataforma debe tener cargado previamente el sistema de referencia. En nuestro caso se utilizará EPSG:4326 (Latitud-Longitud según sistema de referencia EPSG:4326 / WGS84) El WGS84 es el sistema actual utilizado por el sistema GPS.
  - MESH=ESP; Identificadores de los grupos de mallas en los que se desea buscar. En este caso aplicamos la malla de España que nos desvolverá la carretera, municipio, barrio, calle, provincia, comunidad autónoma y código postal correspondiente a la ubicación.

Teniendo en cuenta todos los puntos anteriores la URL final a la que realizar la operación HTTP-GET tendrá la siguiente forma:

```
http(s)://IP:puerto/GPP/WLServer?CLIENT=$client&CLI_PASSWD=$cli_passwd&USER=$user_login&USER_PASSWD=$user_passwd&Method=M_FINDIT&TUSERID=$MSISDN&TUSERID_TYPE=MSISDN&GROUP=$grupo&SRS=EPSG:4326&MESH=ESP
```

El resultado a una petición de este tipo sería un XML con la siguiente estructura:

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <GPPM_IT version="3.5" id="400c8a01-12c7dd74394-0" status="0">
- <LIST name="MOBILE_STATIONS">
  - <ITEM>
    <ORD>1</ORD>
    <M_ROUTEIT>NAVL</M_ROUTEIT>
    <DIST Unit="EPSG:9001">0</DIST>
    <FIELD name="MSISDN">34686233886</FIELD>
    <FIELD name="Nickname" />
    <FIELD name="Validado">Y</FIELD>
    <FIELD name="Localizable">Y</FIELD>
    <FIELD name="Fecha inscripcion">2010-08-27 13:16:45</FIELD>
  - <LOC>
    <TIME>20101124131639</TIME>
    - <POINT srs="EPSG:4326">
      <LAT>40 18 50.79N</LAT>
      <LON>3 46 34.06W</LON>
    </POINT>
    - <SECC>
      - <CENTER>
        - <POINT srs="EPSG:4326">
          <LAT>40 19 35.49N</LAT>
          <LON>3 44 58.75W</LON>
        </POINT>
      </CENTER>
      <SANG Unit="EPSG:9102">192.0</SANG>
      <EANG Unit="EPSG:9102">286.0</EANG>
      <IR Unit="EPSG:9001">1500.0</IR>
      <OR Unit="EPSG:9001">3777.0</OR>
    </SECC>
  - <MBR>
    - <RECT>
      - <SW>
        - <POINT srs="EPSG:4326">
          <LAT>40 17 34.61N</LAT>
          <LON>3 47 37.35W</LON>
        </POINT>
      </SW>
    </RECT>
  </MBR>
</LOC>
</ITEM>
</LIST>
</GPPM_IT>

```

Annotations in the image:

- A blue box labeled "Status=0 → OK" points to the `status="0"` attribute in the root `<GPPM_IT>` tag.
- A blue box labeled "Coordenadas" points to the `<POINT srs="EPSG:4326">` block within the `<LOC>` section.

#### Ejemplo 4.8. Resultado MFINN\_IT primera sección

En esta primera sección del resultado XML se devuelve el resultado de la operación (si ha sido o no satisfactoria), así como los datos correspondientes al MSISDN localizado, la fecha y las coordenadas del punto más probable de ubicación, así como las dimensiones del área aproximada de localización.



```

    </POINT>
  </NE>
</RECT>
</MBR>
- <MESH id="CCAA">
  - <CELL level="6400" percent="98">
    <FIELD name="literal">MADRID</FIELD>
    <FIELD name="complem" />
    <FIELD name="codigo">09</FIELD>
    <FIELD name="categ">0</FIELD>
  </CELL>
</MESH>
- <MESH id="PROV">
  - <CELL level="6400" percent="98">
    <FIELD name="literal">MADRID</FIELD>
    <FIELD name="complem" />
    <FIELD name="codigo">28</FIELD>
    <FIELD name="categ">0</FIELD>
    <FIELD name="caa">MADRID</FIELD>
  </CELL>
</MESH>
- <MESH id="MUNI">
  - <CELL level="800" percent="83">
    <FIELD name="literal">LEGANÉS</FIELD>
    <FIELD name="complem" />
    <FIELD name="codigo">28074</FIELD>
    <FIELD name="categ">0</FIELD>
    <FIELD name="provin">MADRID</FIELD>
    <FIELD name="caa">MADRID</FIELD>
  </CELL>

```

CCAA más probable

Provincia más probable

Municipio más probable

#### Ejemplo 4.9. Resultado MFIND\_IT segunda sección

En esta sección se enriquece la operación de localización con datos de las redes de información que almacena la GPP, como puede ser comunidad autónoma, provincia y municipio, identificando el porcentaje de probabilidad cuando pueden coincidir varios elementos (municipios, provincias o comunidades autónomas).

```

</CELL>
</MESH>
- <MESH id="BARRIOS">
- <CELL level="400" percent="6">
  <FIELD name="literal">POLIGONO EL PORTILLO</FIELD>
  <FIELD name="complem" />
  <FIELD name="codigo" />
  <FIELD name="categ">0</FIELD>
  <FIELD name="nucleo">LEGANES</FIELD>
  <FIELD name="munici">LEGANES</FIELD>
  <FIELD name="provin">MADRID</FIELD>
  <FIELD name="ccaa">COMUNIDAD DE MADRID</FIELD>
</CELL>
</MESH>
- <MESH id="CODPOST">
- <CELL level="400" percent="70">
  <FIELD name="literal">28914</FIELD>
  <FIELD name="complem" />
  <FIELD name="codigo" />
  <FIELD name="categ">0</FIELD>
  <FIELD name="nucleo" />
  <FIELD name="munici">LEGANÉS</FIELD>
  <FIELD name="provin">MADRID</FIELD>
  <FIELD name="ccaa">MADRID</FIELD>
</CELL>
- <CELL level="400" percent="3">
  <FIELD name="literal">28914</FIELD>
  <FIELD name="complem" />
  <FIELD name="codigo" />
  <FIELD name="categ">0</FIELD>

```

**Código Postal más probable**

#### Ejemplo 4.10. Resultado MFIND\_IT tercera sección

Se completa el resultado con datos relativos al barrio donde está ubicado el usuario, así como el código postal.



```

    <FIELD name="nucleo">PARQUE RESIDENCIAL POLVORANCA</FIELD>
    <FIELD name="munici">LEGANÉS</FIELD>
    <FIELD name="provin">MADRID</FIELD>
    <FIELD name="ccaa">MADRID</FIELD>
  </CELL>
- <CELL level="400" percent="3">
  <FIELD name="literal">28914</FIELD>
  <FIELD name="complem" />
  <FIELD name="codigo" />
  <FIELD name="categ">0</FIELD>
  <FIELD name="nucleo" />
  <FIELD name="munici">GETAFE</FIELD>
  <FIELD name="provin">MADRID</FIELD>
  <FIELD name="ccaa">MADRID</FIELD>
</CELL>
- <CELL level="400" percent="2">
  <FIELD name="literal">28905</FIELD>
  <FIELD name="complem" />
  <FIELD name="codigo" />
  <FIELD name="categ">0</FIELD>
  <FIELD name="nucleo" />
  <FIELD name="munici">GETAFE</FIELD>
  <FIELD name="provin">MADRID</FIELD>
  <FIELD name="ccaa">MADRID</FIELD>
</CELL>
</MESH>
- <MESH id="CALLES">
- <CELL level="400" percent="22">
  <FIELD name="literal">AUTOV CIRCUNVALACION M-50</FIELD>

```

#### Ejemplo 4.11. Resultado MFIND\_IT cuarta sección

Tras la información relativa a comunidad autónoma, provincia, municipio y código postal se muestran las calles o carreteras más cercanas.



```

    <FIELD name="complem" />
    <FIELD name="codigo" />
    <FIELD name="categ">0</FIELD>
    <FIELD name="nucleo" />
    <FIELD name="munici">Leganés</FIELD>
    <FIELD name="provin">MADRID</FIELD>
    <FIELD name="ccaa">MADRID</FIELD>
  </CELL>
</MESH>
- <MESH id="CARRET">
- <CELL level="6400" percent="75">
  <FIELD name="literal">M-405 (1.5)/M-406 (5.1)/M-409 (1.8)</FIELD>
  <FIELD name="complem">3/6.8/3.7/4/1.3/3/11.6</FIELD>
  <FIELD name="codigo" />
  <FIELD name="categ">4</FIELD>
</CELL>
- <CELL level="6400" percent="14">
  <FIELD name="literal">M-40 (28.7)/M-402 (1.6)/M-406 (4.8)</FIELD>
  <FIELD name="complem">7.4/3.2/9.7/0.7/7.4/4.2/2.2/6.6/3.3/2</FIELD>
  <FIELD name="codigo" />
  <FIELD name="categ">2</FIELD>
</CELL>
- <CELL level="6400" percent="9">
  <FIELD name="literal">M-402 (3.1)/M-406 (12.1)/M-50 (14)</FIELD>
  <FIELD name="complem">6.2/9.7/10/11/7.8/0.5</FIELD>
  <FIELD name="codigo" />
  <FIELD name="categ">2</FIELD>
</CELL>
</MESH>
</LOC>
</ITEM>
</LIST>
</GPPM_IT>

```

Ejemplo 4.12. Resultado MFIND\_IT última sección

En la siguiente figura se muestra el punto más probable de la localización anterior utilizando la API de Google Maps:



**Figura 4.5. Representación la localización en Google Maps**

Y la representación de los datos anteriores:



**Guillermo**  
**Latitud:** 40.318888888889  
**Longitud:** -3.7652777777778  
**PROVINCIA:** MADRID  
**MUNICIPIO:** LEGANÉS  
**CODPOST:** 28914  
**CARRET:** M-40 (28.7)/M-402  
 (1.6)/M-406 (4.8)/M-409  
 (0.3)/M-40N (28.7)/M-411  
 (2.1)/M-421 (1.1)/M-425  
 (3.3)/N-4

**Figura 4.6. Detalles de la localización**



A continuación se muestra el XML devuelto en caso que se produzca un error en la localización:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <GPPM_IT version="3.5" id="400c8a01-12c7dc628cf-0" status="0">
- <LIST name="MOBILE_STATIONS">
- <ITEM>
  <ORD>1</ORD>
  <M_ROUTEIT>NAVL</M_ROUTEIT>
  <DIST Unit="EPSG:9001">0</DIST>
  <FIELD name="MSISDN">34686233886</FIELD>
  <FIELD name="Nickname" />
  <FIELD name="Validado">Y</FIELD>
  <FIELD name="Localizable">Y</FIELD>
  <FIELD name="Fecha inscripcion">2008-11-14 15:03:09</FIELD>
- <LOC>
  <TIME>20101124130529</TIME>
  <ERROR code="799">Unable to locate the mobile station. MPC positioning error, other. 505.</ERROR>
</LOC>
</ITEM>
</LIST>
</GPPM_IT>
```

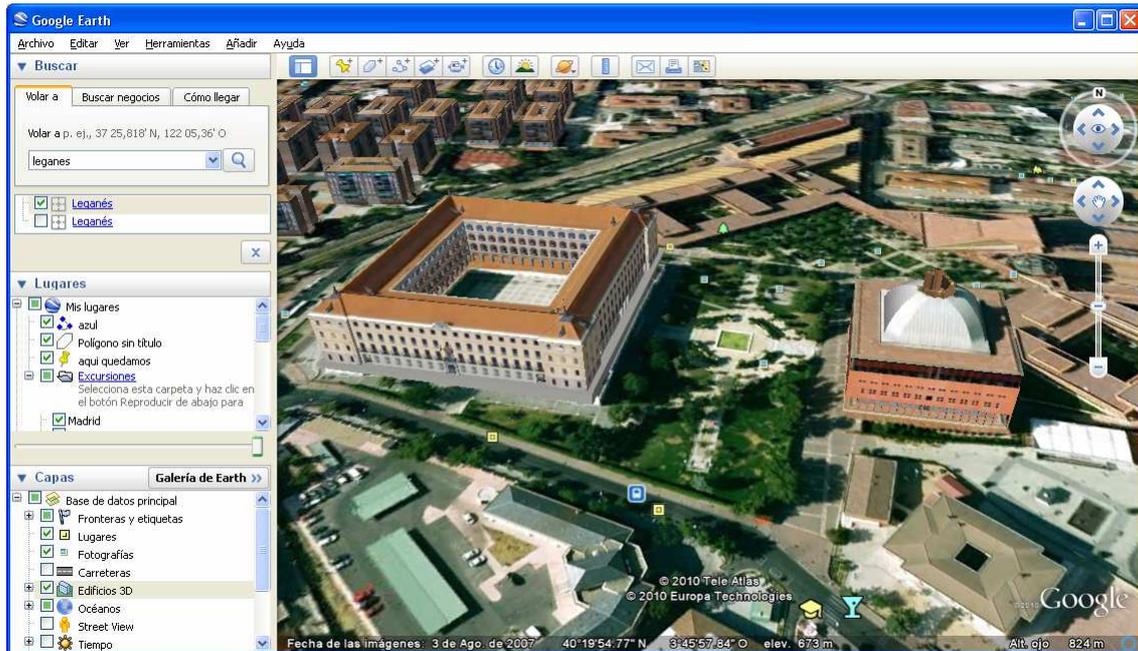
#### Ejemplo 4.13. XML Resultado erróneo en la localización

### 4.4. Estudio de la API de cartografía

En Octubre del 2004 Google compra Keyhole Corp, una compañía californiana dedicada a los mapas digitales, y fundada en 2001. El vicepresidente de Google comentaba tras la compra: *“Con Keyhole, puedes volar como un superhéroe desde tu ordenador de casa hasta la esquina de una calle de cualquier parte del mundo. O encontrar un hospital local, o trazar el mapa de un viaje por carretera, o medir la distancia entre dos puntos”*.

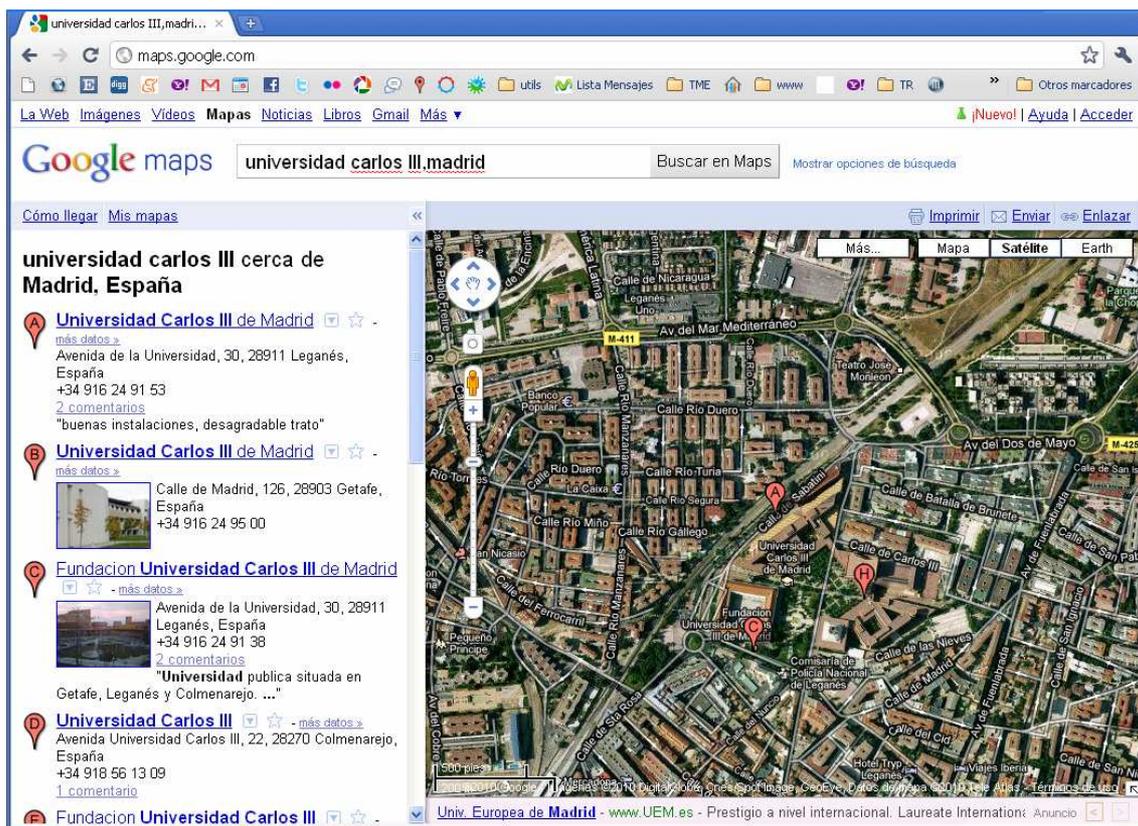
Google era consciente del potencial de la publicidad georeferenciada y hacia una apuesta por la empresa que hasta el momento había demostrado ser la más innovadora en este campo. Tras la compra de Keyhole, Google no tarda en integrar su conocimiento y tecnología en la empresa y libera al mercado los productos Google Earth y Google Maps.

Google Earth es un Sistema de Información Geográfica (SIG) que permite visualizar imágenes en 3D del planeta, combinando imágenes de satélite, mapas y el motor de búsqueda de Google que permite ver imágenes a escala de un lugar específico del planeta. La principal novedad de Google Earth es que es el único producto de su clase gratuito con cartografía actualizada de las principales ciudades del mundo.



**Figura 4.7. Captura de Google Earth**

A su vez, lanza un cliente web de la tecnología cartográfica de Keyhole, Google Maps, que se convertirá en una auténtica revolución. Se puede acceder a la mayor parte de la funcionalidad de Google Earth desde cualquier navegador web que soporte Javascript. En pocos años consigue una gran aceptación y no tardan en ofrecer una API a terceros de cartografía.



**Figura 4.8. Captura de Google Maps**

La API evoluciona rápidamente y aunque inicialmente la única API que existía era una basada en Javascript, actualmente ofrecen seis variantes:

- Maps Javascript API: JavaScript permite a los usuarios insertar un mapa de Google en sus páginas web, así como manipular el mapa y añadir contenido a través de diferentes servicios.
- Maps API for Flash: ActionScript API permite a los usuarios insertar un mapa de Google en sus aplicaciones o en sus páginas web basadas en Flash. Asimismo, les permite manipular el mapa en tres dimensiones y añadir contenido a través de diferentes servicios
- Google Earth API: Permite a los usuarios insertar un globo digital en 3D en sus páginas web. De este modo, los usuarios que visiten la página del usuario podrán viajar por cualquier lugar del mundo (incluso por debajo del océano) sin salir de la página.



- Static Maps API: Permite a los usuarios insertar una imagen rápida y sencilla de Google Maps en sus páginas web o en sus sitios para móviles sin necesidad de utilizar JavaScript ni ningún sistema de carga de páginas dinámicas.
- Servicios web: Puedes utilizar las solicitudes de URL para acceder a información de lugares, de direcciones o de codificación geográfica de las aplicaciones cliente, y manipular los resultados en JSON o en XML.
- Maps Data API: Puedes visualizar, almacenar y actualizar datos de mapas a través de los feeds de Google Data API, mediante un modelo de funciones (marcadores, líneas y formas) y conjuntos de las mismas.

La API que se ha utilizado en este proyecto es la Maps Javascript API, ya que está orientado a un entorno cliente basado en un ordenador con un navegador Web que aprovechará el potencial de esta API y es la interfaz más extendida ya que es la utilizada el servicio tradicional de Google Maps. De esta manera podremos adaptar el zoom o desplazarnos en el mapa resultante e incluir marcadores con la ubicación y descripción del individuo localizado.

A la hora de invocar a la API de Google Maps, lo primero que hay que hacer es solicitar una clave para poder utilizarla. La clave se solicita por URL hasta en nivel de la carpeta en la que esté la página que la invoca, por ejemplo, si la página web que va a invocar al Javascript está en <http://www.tudominio.com/carpeta1/fichero.html> hay que solicitar una clave para la URL <http://www.tudominio.com/carpeta1> Una vez que ya se tiene la clave hay que introducirla en la cabecera de la página web:

```
<head>
<script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAF9f8
9eXg9eR20r9hZAO96xSy9MlvxEspMpa-
wZbiEwUfQKUUXxQJUmqhYDIMoI94JWgvEgpaBfHO0A" type="text/javascript">
</script>
</head>
```

#### Ejemplo 4.14. Inclusión clave API Google Maps

A continuación, dentro del cuerpo de la página hay que introducir una tabla donde se generará el mapa introduciendo el siguiente código HTML:

```
<table>
<tr><td>
<div align=center id=map style=width:640px;height:480px></div>
</td></tr>
</table>
```

#### Ejemplo 4.15. Tabla HTML con identificador del mapa

A continuación se invoca a un JavaScript que genera un mapa en el div con id=map y dibuja un punto en las coordenadas \$latitud, \$longitud y junto a una descripción del punto (Nombre, Descripción):

```
<script type="text/javascript">
  //

  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map"));
    map.setCenter(new GLatLng($latitud, $longitud), 13);
    map.enableDoubleClickZoom();
    map.addControl(new GSmallMapControl());
    map.addControl(new GOverviewMapControl());

    var baseIcon = new GIcon(G_DEFAULT_ICON);
    baseIcon.shadow =
"http://www.google.com/mapfiles/shadow50.png";
    baseIcon.iconSize = new GSize(20, 34);
    baseIcon.shadowSize = new GSize(37, 34);
    baseIcon.iconAnchor = new GPoint(9, 34);
    baseIcon.infoWindowAnchor = new GPoint(9, 2);

    function createMarker(ref, latlng, nombre, descr) {
      var marker = new GMarker(latlng);
      marker.value = nombre;
      var letter = String.fromCharCode("A".charCodeAt(0) + ref);
      var letteredIcon = new GIcon(baseIcon);
      letteredIcon.image = "http://www.google.com/mapfiles/marker"
+ letter + ".png";
      // Set up our GMarkerOptions object
      markerOptions = { icon:letteredIcon };
      var marker = new GMarker(latlng, markerOptions);
      GEvent.addListener(marker, "click", function() {
        var myHtml = "&lt;b&gt;" + nombre + "&lt;/b&gt;&lt;br/&gt;" + descr;
        map.openInfoWindowHtml(latlng, myHtml);
      });

      return marker;
    }

    var marker1=new GLatLng($latitud, $longitud);
    map.addOverlay(createMarker(0,marker1, "Nombre", "Descripcion"
));</pre>
</div>
<div data-bbox="829 936 862 954" data-label="Page-Footer">89</div>
```

```

}

//]]>
</script>

```

### Ejemplo 4.16. JavaScript para generación mapa, marcador y descriptor

Al integrar las tres partes, clave, tabla HTML y Javascript para la generación del mapa y posicionamiento de elementos el resultado es el siguiente:

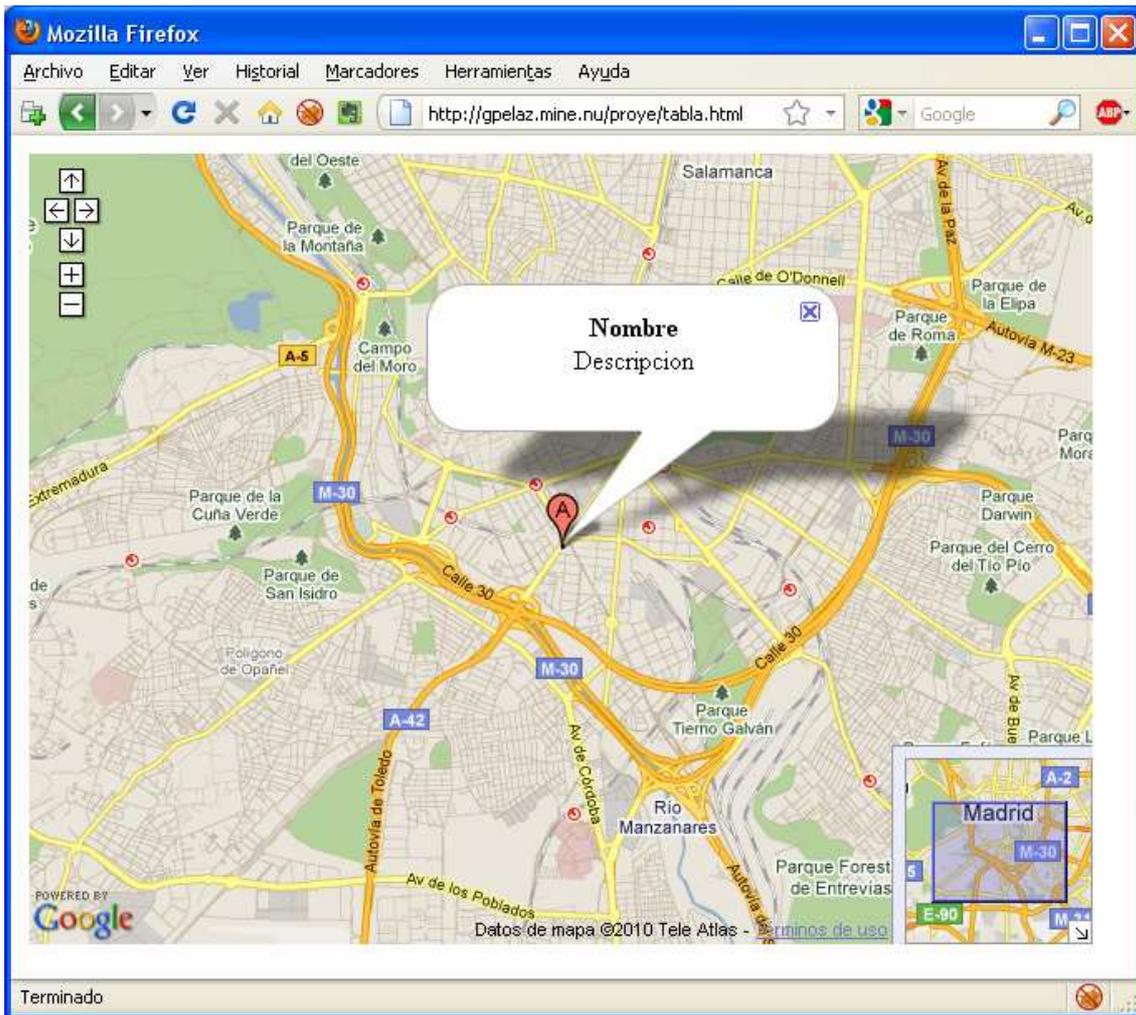


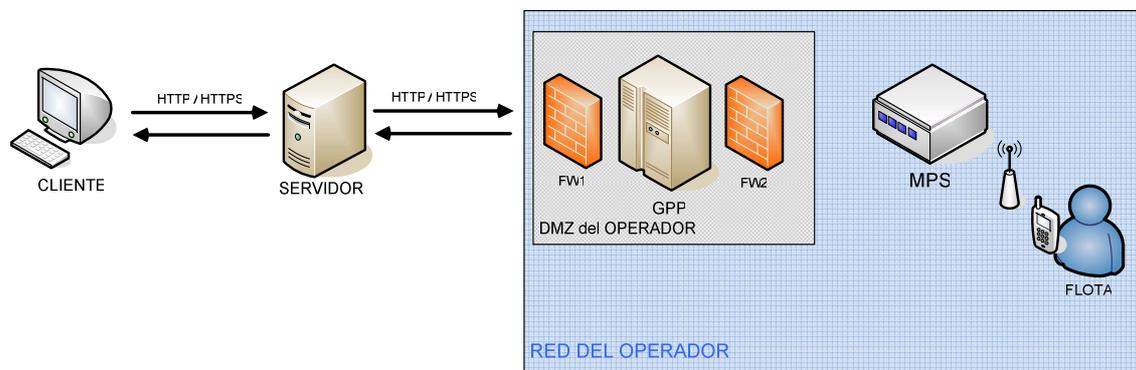
Figura 4.9. Ejemplo API de Google Maps

## 4.5. Diseño arquitectónico

Esta fase del proyecto comprende la última versión de la arquitectura definida para el soporte de la aplicación. Se pretende, en esta sección, dar una visión global de la descomposición en componentes del sistema así como del despliegue de los mismos

### 4.5.1. Infraestructura Hardware

La topología de la aplicación se muestra en la siguiente figura:



**Figura 4.10. Arquitectura del servicio**

En la figura anterior todo el servicio reside en el servidor junto a la base de datos MySQL y el servidor web Apache.

Como se observa en el gráfico, el cliente accede al servidor mediante peticiones HTTP/HTTPS. Si la petición HTTP va a salir de la intranet de la empresa se recomienda la instalación de certificados en el servidor Apache para garantizar la seguridad y privacidad del servicio.

Cuando el cliente solicita una localización, el servidor lanza una operación de localización contra la plataforma de localización del operador (GPP) que se encuentra en la DMZ del operador con IP pública.

La plataforma GPP comprueba los datos de la autenticación de la petición del servicio y si se tiene permiso para localizar a la flota, si es así, consulta al MPS que es la plataforma que se integra con el core de red del operador para identificar el punto más probable de la ubicación de la flota en base a las antenas que le dan cobertura.



El MPS devuelve estas coordenadas a la GPP y ésta devuelve al servidor, en el formato de coordenadas que ha solicitado el usuario y con la información adicional de las redes que haya solicitado (comunidad autónoma, provincia, ciudad, código postal, barrio, carreteras más cercanas, etc...), el resultado final.

El servidor se encarga de parsear y procesar la información y la representa gráficamente utilizando la API de Google Maps así como guardar un registro de la localización para futuras consultas.

#### **4.5.2. Infraestructura Software**

La aplicación queda desplegada en el servidor donde también residirá la instalación de PHP, Apache y MySQL.

La instalación del software es trivial ya que existen multitud de distribuciones de Linux con paquetes de software preconfigurado para hacer desarrollos sobre arquitecturas LAMP (**L**inux – **A**pache – **M**ySQL – **P**HP). Además últimamente han proliferado también productos para sistemas operativos Windows en los que la instalación del Apache – MySQL – PHP se realiza de manera trivial y hace que se reduzca el tiempo para conseguir la disponibilidad técnica de un servicio basado en esta arquitectura.

Para realizar el despliegue en el servidor, una vez instalado el Apache, MySQL y PHP tan solo hay que realizar la precarga de la base de datos importando el script SQL para inicializarla, copiar todos los ficheros .php en la carpeta configurada del servidor web y preconfigurar los datos de autenticación de la plataforma de localización en el servidor.

Una vez realizados todos los pasos anteriores ya podemos comenzar a hacer las pruebas iniciales del sistema.

#### **4.6. Diseño detallado**

En este apartado se hablará de cómo se ha implementado la funcionalidad requerida usando para ello capturas de pantalla de la aplicación y destacando los aspectos más relevantes de la misma.

### 4.6.1. Implementación y pruebas de la Fase Final del Sistema

En primer lugar se muestra la pantalla principal de la aplicación donde se podrá escribir el login y la clave de acceso del usuario. La validación se realiza contra los usuarios creados con anterioridad en la base de datos MySQL por el usuario administrador.

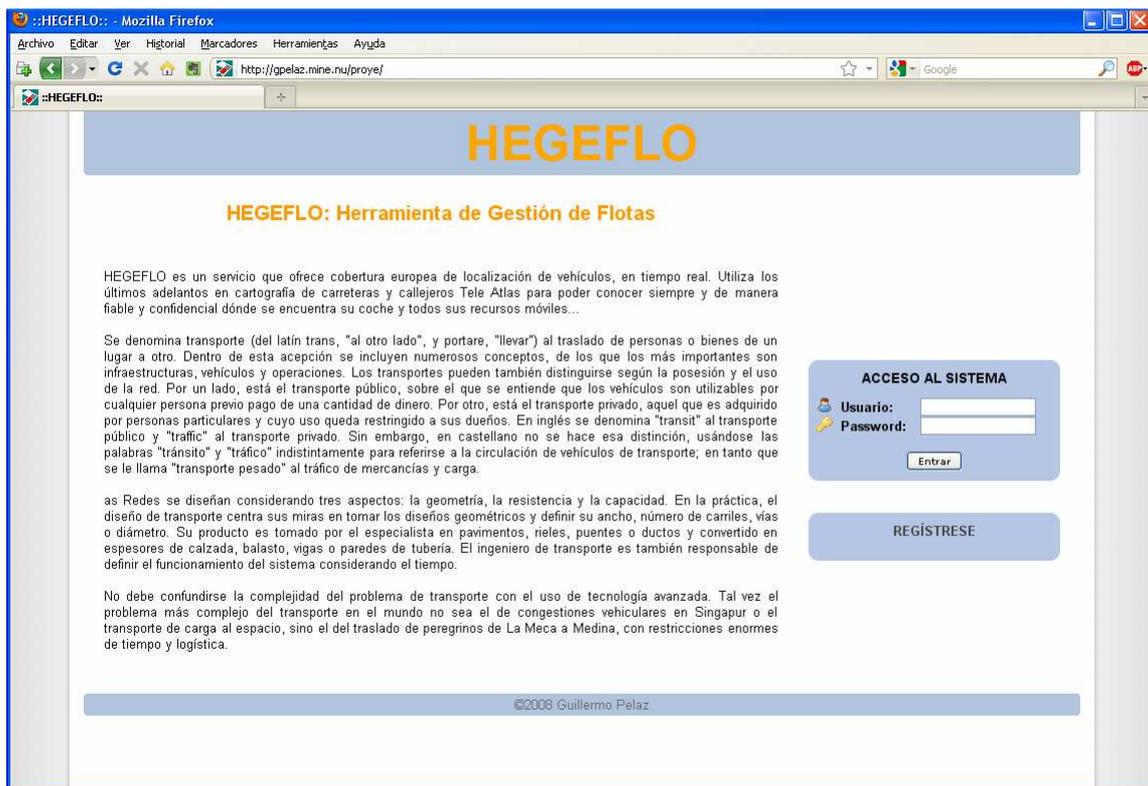


Figura 4.11. Página principal de la aplicación

Cuando el usuario accede a la aplicación, en función de su perfil, aparecen las operaciones a las que tenga acceso en la página principal. Por ejemplo, para el perfil Administrador, la pantalla de inicio es la de Administración.



**Figura 4.12. Pantalla de Inicio para un perfil de administrador**

Dado que el perfil Administrador es el perfil menos restrictivo y permite tener acceso a todas las funcionalidades de la aplicación nos centraremos en él para la descripción de cada una de ellas.

El menú principal, situado en la parte superior izquierda, permite el acceso a las siguientes funcionalidades:

- **Gestión de la GPP** Permite acceder al set completo de operaciones contra la plataforma para realizar operaciones de provisión relativas creación, consulta o borrado de usuarios, grupos y miembros de grupos:
  - createUser
  - getUser
  - deleteUser
  - createGroupS
  - getChildGroup

- deleteGroup
- createMemberRec
- getMember
- deleteMemberRec

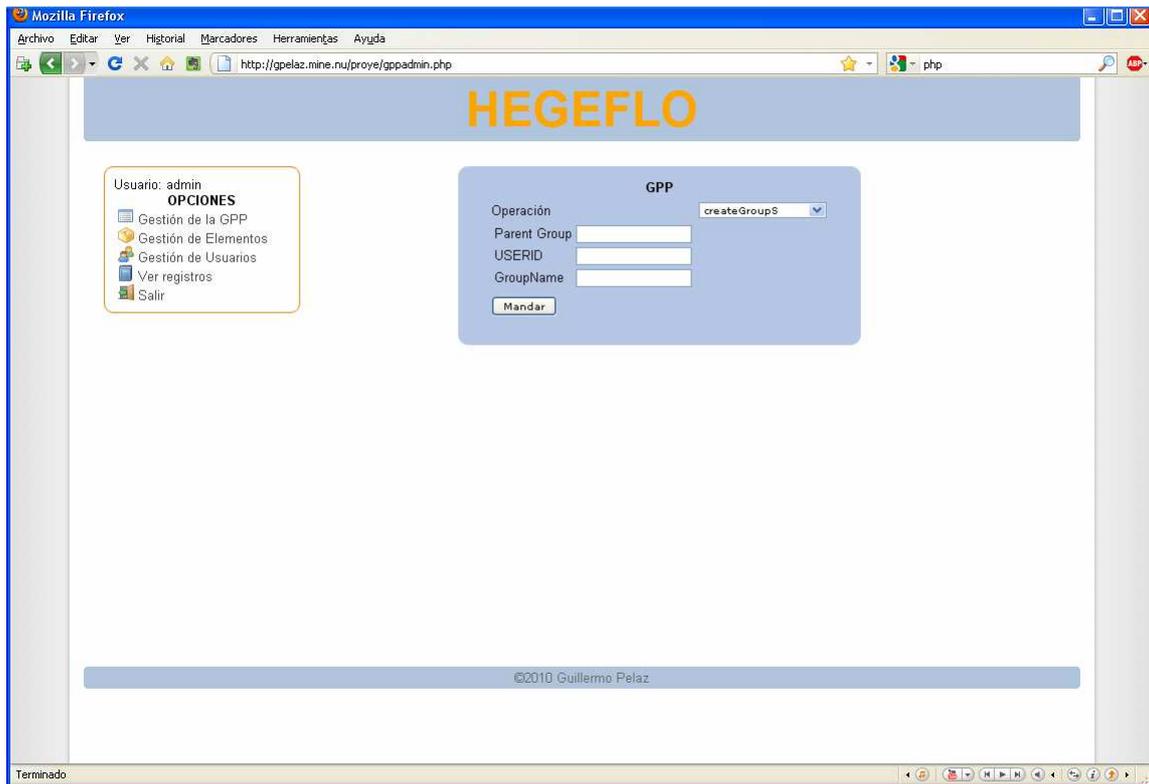


Figura 4.13. Pantalla de Gestión de la GPP

- **Gestión de Elementos** Permite consultar los elementos provisionados en las flotas y habilitar o deshabilitar su localización, así como su borrado.

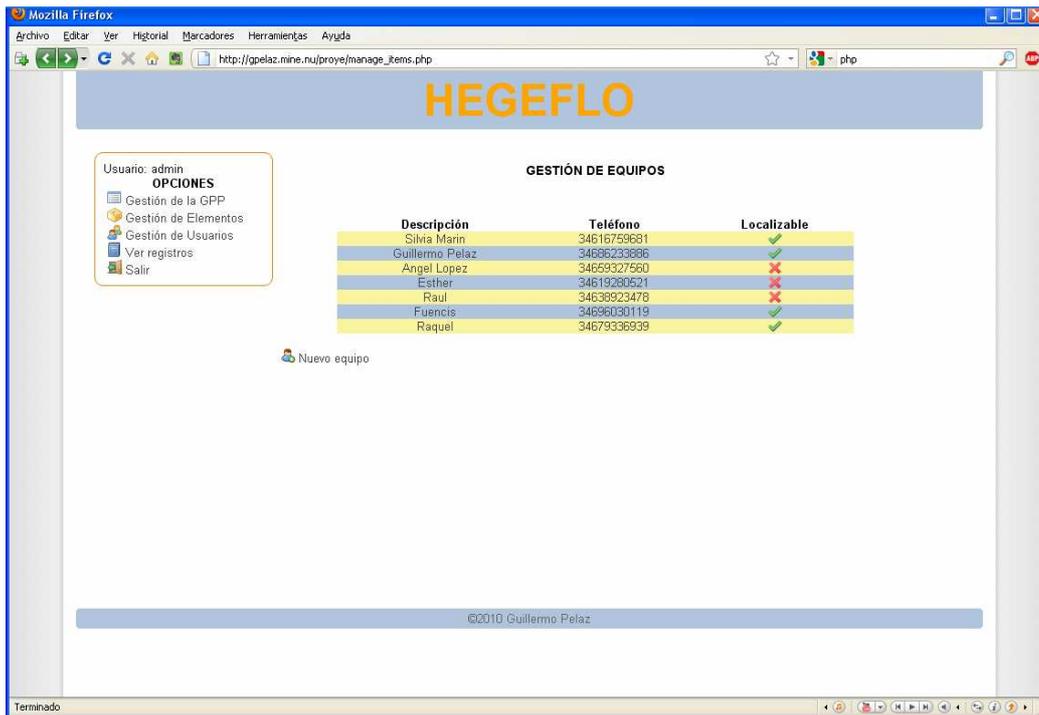


Figura 4.14. Pantalla de Gestión de Elementos

- **Gestión de Usuarios** Permite gestionar los usuario que tienen acceso a la realización de localizaciones, así como gestión de la flota.

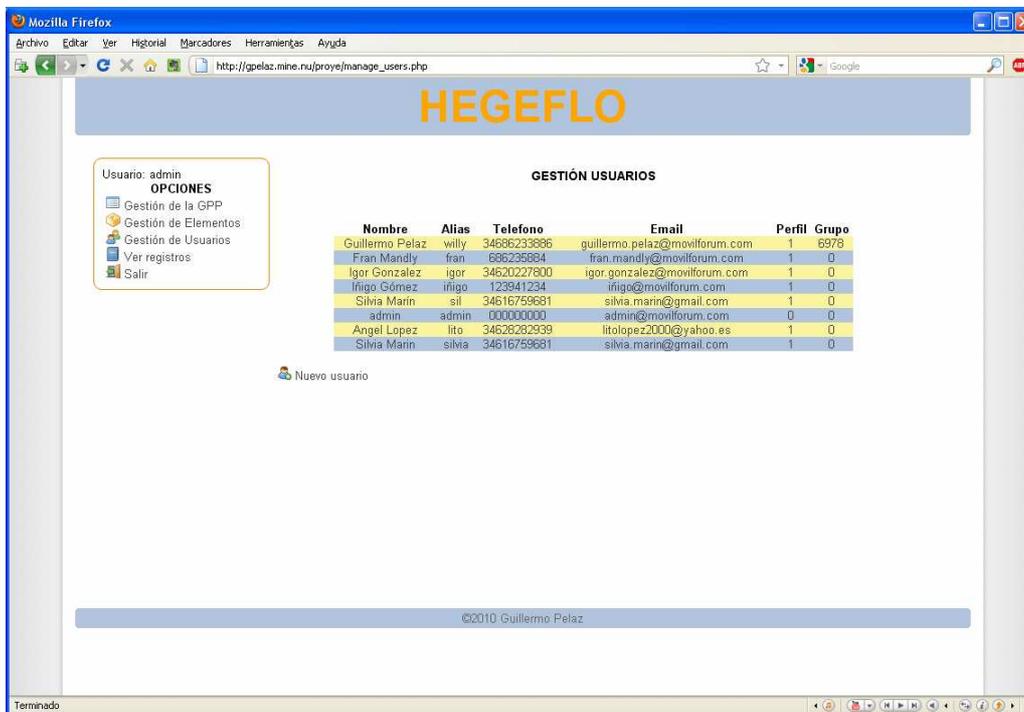


Figura 4.15. Pantalla de Gestión de usuarios

- **Ver histórico** Permite consultar el histórico de localizaciones de los usuarios.

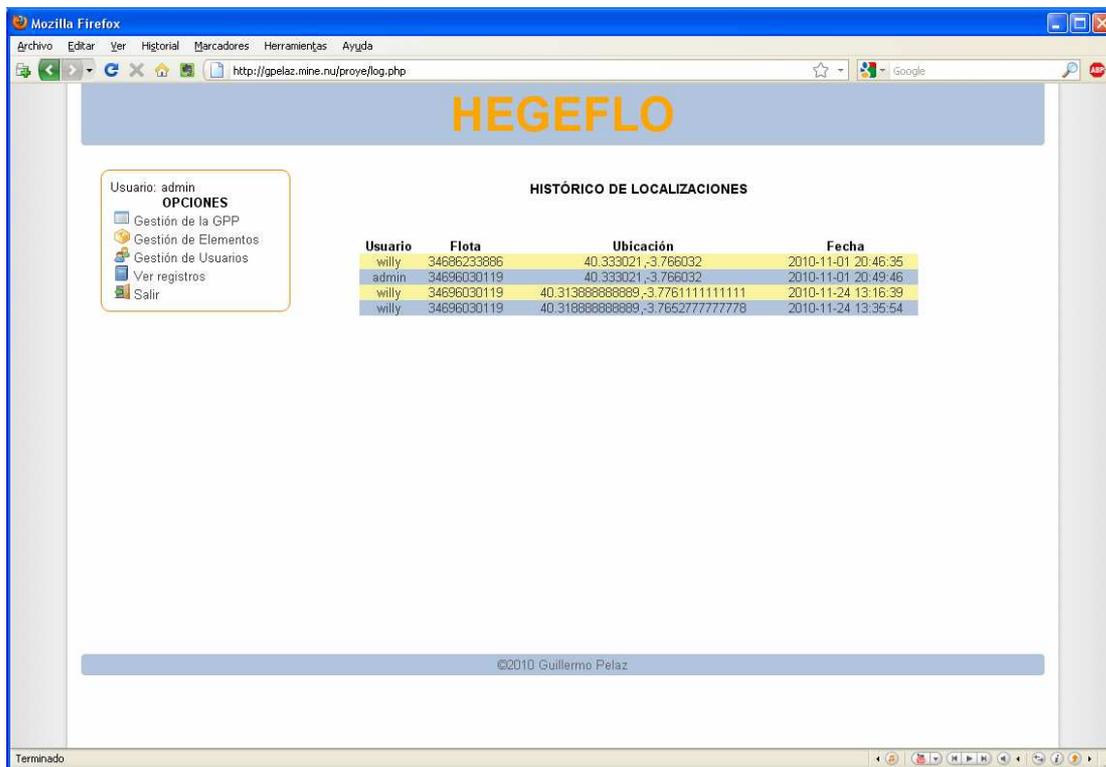


Figura 4.16. Pantalla para “Ver Registros”

- **Salir** Permite al usuario cerrar su sesión actual.

Por encima del menú principal se encuentra una zona personalizada para el usuario que contiene nombre del usuario.

Inicialmente se han definido dos roles principales para la aplicación: administrador o usuario. Ya se ha descrito las funcionalidades a las que tiene acceso el administrador y a continuación se muestra la pantalla de entrada de usuario sin perfil de administrador



**Figura 4.17. Pantalla de Inicio para un perfil de usuario**

Las funciones a las que tiene acceso un usuario normal son:

- **Gestión de mi Flota** Permite modificar elementos de la flota del usuario.



**Figura 4.18. Pantalla de Gestión de mi Flota**

- **Envío de SMS** Permite el envío de SMS a individuos de la flota

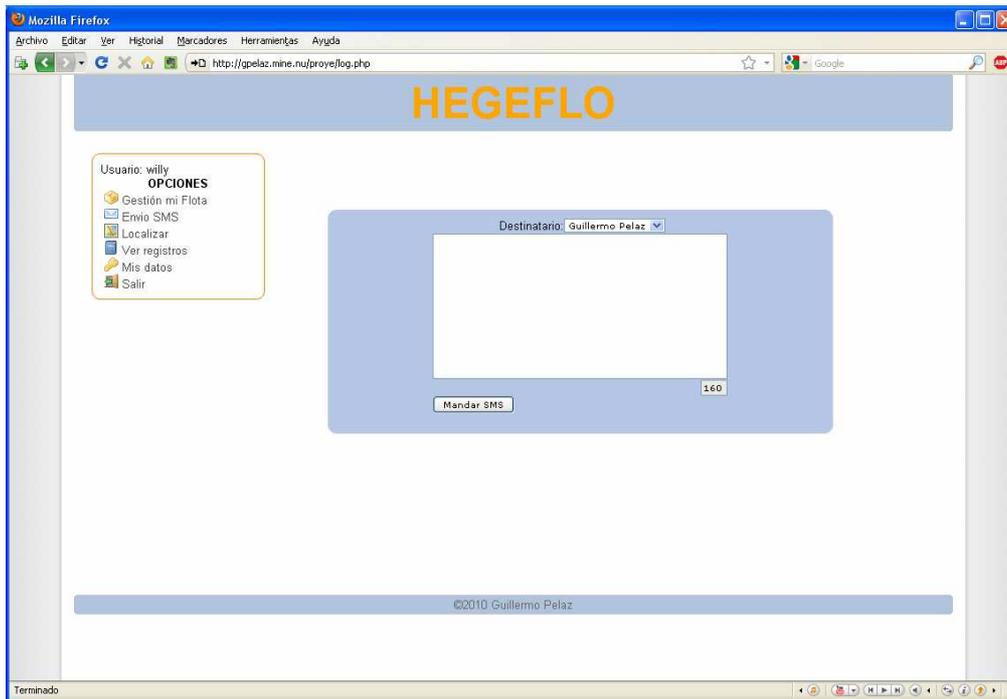


Figura 4.19. Pantalla de Envío de SMS

- **Localizar** Permite localizar a individuos de la flota

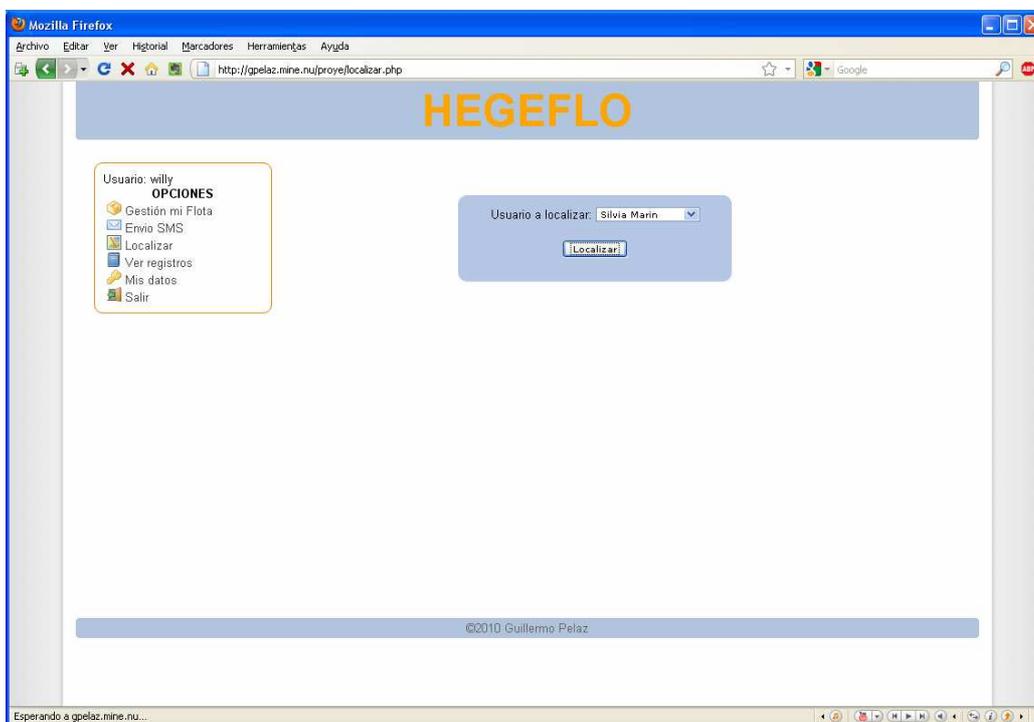


Figura 4.20. Pantalla Localizar

En el caso de una localización satisfactoria se muestra en pantalla en un primer lugar un mapa con el icono más probable de la ubicación de la flota y en una segunda sección la información devuelta por la plataforma GPP (coordenadas, provincia, municipio, código postal y carreteras más cercanas).

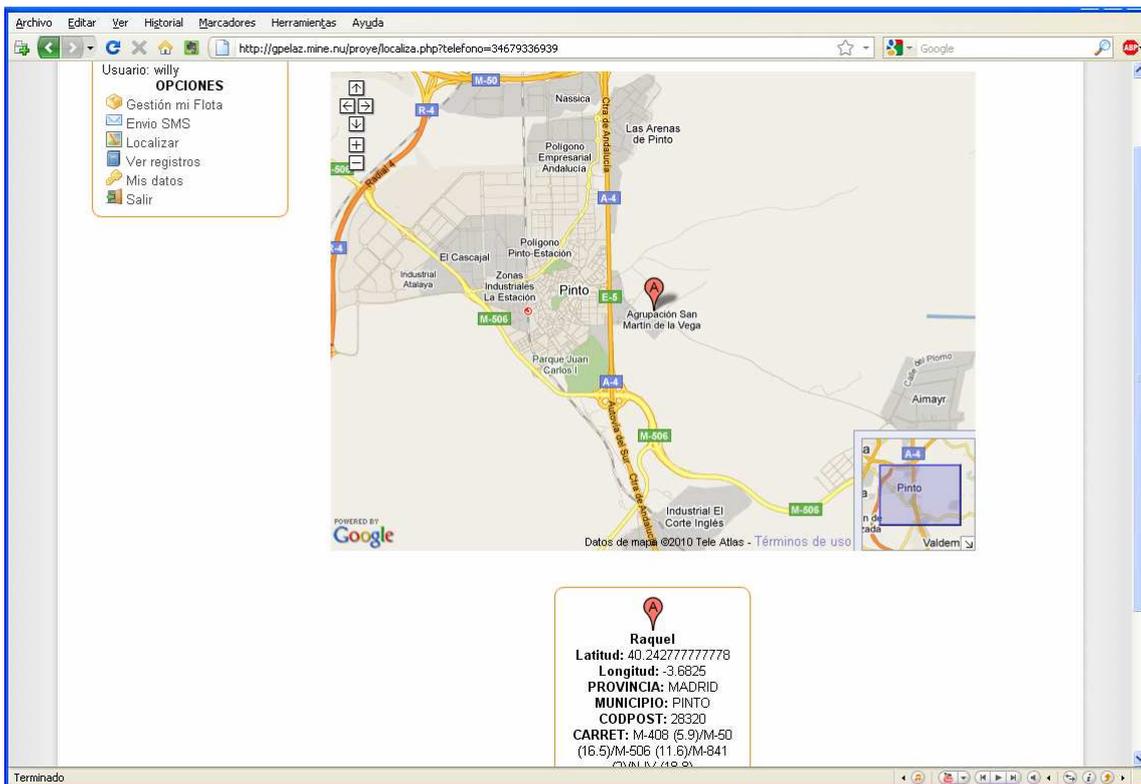


Figura 4.21. Pantalla con resultado de localización

- **Ver Registros** Permite consultar los registros de localización pero sólo de las localizaciones realizadas por el usuario actual.

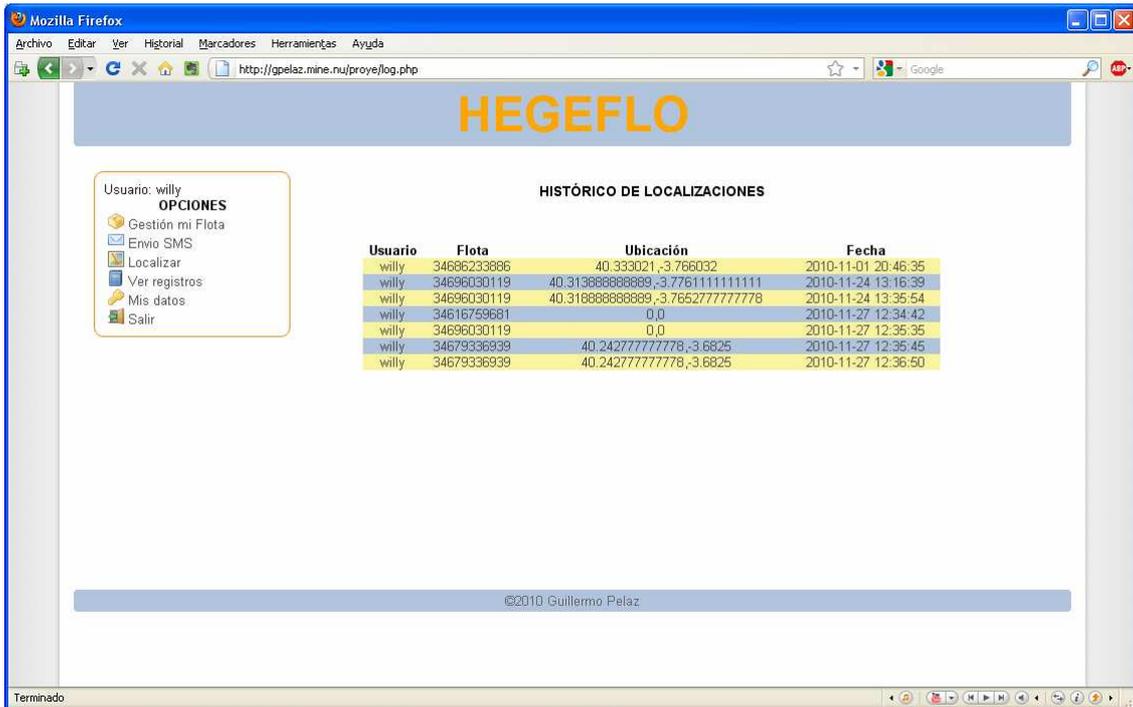


Figura 4.22. Pantalla para Ver Registros por usuario

➤ **Mis datos** Permite al usuario cambiar su password actual.



Figura 4.23. Pantalla Mis datos

### 4.6.2. Consideraciones importantes sobre el desarrollo

Toda la configuración relativa al servicio con los datos de provisión proporcionados por el operador se definen en el fichero config.php:

```
<?php
#Parámetros de configuración de la GPP
$gpp_server="195.195.195.195";
$gpp_port=8001;
$url_gpp="/GPP/WLServer?";
$client="pw001";
$cli_passwd="pw002";
$user_login="pw003";
$user_passwd="pw004";

#Parámetros de configuración del MIB (plataforma de mensajería)
$mib_server="172.16.120.27";
$mib_port="9800";
$mib_url="/mm7extadapter";
$mib_sms_sender="2116";
$mib_user="mf";
$mib_pwd="mf33";

?>
```

#### Ejemplo 4.17. Fichero config.php

A la hora de realizar las operaciones de provisión en la GPP se ha desarrollado una librería que genera los XML a enviar hacia la plataforma. A continuación se muestra la parte del código relativa a la generación del xml para crear un usuario:

```
function generate_xml($operation,$var1,$var2,$var3) {
require('config.php');

switch ($operation) {
case "createUser":
// createUser Operation $var1 is the MSISDN
$xml='<?xml version="1.0" encoding="iso-8859-1"?>';
$xml.='<GPPR>';
$xml.='<FUNCION type="INSERT">createUser</FUNCION>';
$xml.='<AUTENTIFICACION>';
$xml.='<CLIENT>$client</CLIENT>';
$xml.='<CLI_PASSWD>$cli_passwd</CLI_PASSWD>';
$xml.='<USER_LOGIN>$user_login</USER_LOGIN>';
$xml.='<USER_PASSWD>$user_passwd</USER_PASSWD>';
$xml.='</AUTENTIFICACION>';
$xml.='<PARAMETERS identype="MSISDN">';
$xml.='<ATTRIBUTE name="USER" param="Y">';
$xml.='<OBJECT>';
$xml.='<CLASS>USER</CLASS>';
$xml.='<CONTENTS>';
$xml.='<ATTRIBUTE name="MSISDN">';
$xml.='<STRING val="'.$var1.'"/>';
$xml.='</ATTRIBUTE>';
```



```

$xml.='</CONTENTS>';
$xml.='</OBJECT>';
$xml.='</ATTRIBUTE>';
$xml.='</PARAMETERS>';
$xml.='</GPPR>';
break;
case "getUser":
...
}
return $xml;
}

```

#### Ejemplo 4.18. Función generate\_xml del fichero lib\_gpp.php

De esta manera ganamos la flexibilidad de poder modificar de una manera sencilla el formato de datos que se tiene que enviar al operador en caso de que se generen nuevos formatos u operaciones.

A la hora de solicitar la localización de un usuario se utilizan las siguientes funciones:

```

function generate_url($operation,$MSISDN,$grupo) {
require('config.php');
$url=$url_gpp;
$auth="CLIENT=$client&CLI_PASSWD=$cli_passwd&USER=$user_login&USER_PAS
SWD=$user_passwd";
switch ($operation) {
case "M_FINDIT":

$url.="Method=M_FINDIT&$auth&TUSERID=$MSISDN&TUSERID_TYPE=MSISDN&GROUP
=$grupo&SRS=EPSG:4326&MESH=ESP";
break;
case "mtrackit":

break;
}
//echo "Devolviendo:",$url;
return $url;
}

function localiza($MSISDN,$grupo)
{
include_once('config.php');
require_once('lib_http.php');
$get_url=generate_url('M_FINDIT',$MSISDN,$grupo);
$result=http_get($gpp_server,$gpp_port,$get_url);
return $result;
}

```

#### Ejemplo 4.19. Función generate\_url y localiza del fichero lib\_gpp.php

A la hora de procesar la información devuelta por la GPP, se ha creado la librería `lib_parse.php` que define funciones para procesar el xml resultante de la petición de localización. A continuación se muestra la función de procesado del xml `parse_gpp` que se encarga de adaptar el sistema de coordenadas para poder mostrarlo posteriormente en google maps.

```
<?php

function parse_gpp($gpp_xml){
$xml = simplexml_load_string($gpp_xml);
//var_dump($xml);
//echo "Version: ", $xml['version'];
//echo "\nStatus: ", $xml['status'];
//echo "\nMSISDN: ", $xml->LIST->ITEM->FIELD[0];
//echo "\nTIME : ", $xml->LIST->ITEM->LOC->TIME;
//echo "\nLAT: ", $xml->LIST->ITEM->LOC->POINT->LAT;
//echo "\nLON: ", $xml->LIST->ITEM->LOC->POINT->LON;
$latitud=explode(" ", $xml->LIST->ITEM->LOC->POINT->LAT);
$longitud=explode(" ", $xml->LIST->ITEM->LOC->POINT->LON);
$lati=$latitud[0]+($latitud[1]+substr($latitud[2],0,2)/60)/60;
$longi=$longitud[0]+($longitud[1]+substr($longitud[2],0,2)/60)/60;
//Comprobamos si es al OESTE la longitud es negativa
if (substr($longitud[2],-1,1)=="W")
{
    $longi="-".$longi;
}

$xmlresult=<<<FIN_XML_RESULT
<?xml version="1.0" encoding="UTF-8" ?>
<ITEM>
FIN_XML_RESULT;

foreach ($xml->LIST->ITEM->LOC->MESH as $mesh) {
    $xmlresult.="<".$mesh['id']. ">";
    if (isset($mesh->CELL[0]->FIELD[0])) {
        $xmlresult.=$mesh->CELL[0]->FIELD[0];
    }
    $xmlresult.="</".$mesh['id']. ">";

    //echo $mesh->CELL[0]['percent']; //probabilidad
    //echo $mesh->CELL[0]->FIELD[0];
}

$xmlresult.="<MSISDN>".$xml->LIST->ITEM->FIELD[0]. "</MSISDN>". "<LAT>". $lati. "</LAT>". "<LON>". $longi. "</LON></ITEM>";
return $xmlresult;
}
```

#### Ejemplo 4.20. Función `parse_gpp` del fichero `lib_parse.php`

Una vez procesada y adaptada la información de localización se procede a invocar a la API de Google Maps para representar gráficamente la ubicación:

```

<script type="text/javascript">
  //

    if (GBrowserIsCompatible()) {
      var map = new GMap2(document.getElementById("map"));
      map.setCenter(new GLatLng($latitud, $longitud), 13);
      map.enableDoubleClickZoom();
      map.addControl(new GSmallMapControl());
      map.addControl(new GOverviewMapControl());

      //iconos
      // Create a base icon for all of our markers that
specifies the
      // shadow, icon dimensions, etc.
      var baseIcon = new GIcon(G_DEFAULT_ICON);
      baseIcon.shadow =
"http://www.google.com/mapfiles/shadow50.png";
      baseIcon.iconSize = new GSize(20, 34);
      baseIcon.shadowSize = new GSize(37, 34);
      baseIcon.iconAnchor = new GPoint(9, 34);
      baseIcon.infoWindowAnchor = new GPoint(9, 2);

      function createMarker(ref, latlng, nombre, descr) {
        var marker = new GMarker(latlng);
        marker.value = nombre;
        var letter = String.fromCharCode("A".charCodeAt(0) +
ref);

        var letteredIcon = new GIcon(baseIcon);
        letteredIcon.image =
"http://www.google.com/mapfiles/marker" + letter + ".png";
        // Set up our GMarkerOptions object
        markerOptions = { icon:letteredIcon };
        var marker = new GMarker(latlng, markerOptions);

        GEvent.addListener(marker,"click", function() {
          var myHtml = "&lt;b&gt;" + nombre + "&lt;/b&gt;&lt;br/&gt;" + descr;
          map.openInfoWindowHtml(latlng, myHtml);
        });

        return marker;
      }

      //      point=new GLatLng(37.4523, -122.1425)
      //      var prueba = new GMarker(point);
      //      map.addOverlay(prueba);
      //      window.setTimeout(function() { map.panTo(point) },
1000);
      var marker1=new GLatLng($latitud, $longitud);
      //      var willy=new GLatLng(40.309051,-3.729022);

      map.addOverlay(createMarker(0,marker1, "$name", "$xml-
&gt;CODPOST $xml-&gt;MUNI" ));
      //      map.addOverlay(createMarker(1,willy, "Willy", "El mas
guapo"));
</pre>
</div>
<div data-bbox="818 936 862 954" data-label="Page-Footer">105</div>
```



```

    }

    //]]>
</script

```

### Ejemplo 4.21. Script de invocación a Google Maps

Para el desarrollo del proyecto se ha utilizado Smarty como herramienta de generación de plantillas para la generación de las páginas web dinámicas. A continuación se muestra la invocación de la página principal tras logarse en el sistema:

```

<?php
session_start();
// load Smarty library
require('Smarty.class.php');
require_once('lib_func.php');

$smarty = new Smarty;
$smarty->template_dir = 'smarty/templates';
$smarty->config_dir = ' c:/wamp/www/proye/smarty/config';
$smarty->cache_dir = 'c:/wamp/smarty/cache';
$smarty->compile_dir = 'c:/wamp/smarty/templates_c';

$i=$_SESSION["perfil"];
$usuario=$_SESSION["user"];
$menu=generate_menu($i);
$contenido='';
$smarty->assign('usuario',$usuario);
$smarty->assign('menu',$menu);
$smarty->assign('contenido',$contenido);
// $smarty->assign('usuario',$_SESSION["user"]);
$smarty->display('main.tpl');
?>

```

### Ejemplo 4.22. Ejemplo de invocación a Smarty en el fichero main2.php

## 4.7. Resumen del proyecto

La planificación inicial que se planteó para este proyecto fue cumplida en la medida de lo posible. A grandes rasgos, dicha planificación no sufrió modificaciones drásticas en el tiempo estimado para cada tarea. Salvo pequeños reajustes, propios del desarrollo de cualquier trabajo, las líneas maestras trazadas al inicio fueron cumplidas rigurosamente.

Se presenta en este apartado un resumen de las tareas realizadas y su consecución en el tiempo. Para ello se ha utilizado la herramienta gráfica conocida como Diagrama de Gantt, que permite mostrar el tiempo de dedicación previsto para





| Fase                   | Horas |
|------------------------|-------|
| Análisis de requisitos | 62    |
| Diseño del sistema     | 92    |
| Diseño del programa    | 104   |
| Codificación           | 240   |
| Pruebas                | 30    |
| Implementación         | 10    |

**Tabla 4.21. Resumen de horas por fase**

Así pues se desprende que el número de horas dedicadas por el proyectando es de 538 horas. Dado que el coste por hora para un desarrollador de aplicaciones se ha computado a 12 €/ hora, el coste de personal se sitúa en 6456 €.

El coste por uso de la plataforma de localización se desglosa de la siguiente forma:

- 1000€ en un único pago en concepto de cuota de conexión
- 500€ mensuales por cuota de mantenimiento
- El coste por localización tiene un escalado de precios según la tabla siguiente:
  - Hasta 500: 0,09 €
  - De 501 a 1.000: 0,079 €
  - De 1.001 a 10.000: 0,068 €
  - De 10.001 a 100.000: 0,058 €



- De 100.001 a 1.000.000: 0,053 €
- Más de 1.000.000: 0,05 €

De esta manera si estimamos que para las fases de codificación, pruebas e implantación era necesaria la conexión a la plataforma tenemos los siguientes costes asociados:

- 1000€ de cuota de alta
- 500€ cuota mensual x 4 meses = 2.000€
- 200 localizaciones durante la fase de pruebas \* 0.09€ = 18€

Coste total asociado a la plataforma: 3.018€.

El desarrollo se ha realizado sobre un portátil Acer Aspire 5630 y no se ha concretado a fecha de hoy su despliegue en ningún servidor, por lo que no se añadirá coste de HW al proyecto.

| Concepto         | Importe   |
|------------------|-----------|
| Coste personal   | 6456 €    |
| Coste plataforma | 3018€     |
| I.V.A. (18%)     | 1705,32 € |
| Total            | 11179,32€ |

**Tabla 4.22. Resumen de costes**



### **4.7.3. Estado del proyecto**

Actualmente el proyecto se utiliza para realizar demostraciones puntuales de la capacidad de localización de las plataformas de localización del operador móvil en un site no accesible al público.



# CAPÍTULO 5: CONCLUSIONES

Partiendo de los objetivos planteados podemos concluir que la aplicación desarrollada permite realizar demostraciones de la capacidad de localización de la plataforma del operador de una manera potente, sencilla e intuitiva.

Para el desarrollo de la misma se han utilizado tecnologías, lenguajes de programación y APIs que hoy en día representan, probablemente la arquitectura de servicios más extendida, dada su potencia, flexibilidad para adaptarse a cualquier entorno y reducido coste relativo a licencias por parte de los grandes vendedores de software.

Para una consecución con éxito del proyecto se puso especial atención en la captura de los requisitos iniciales, donde se estableció la necesidad de una herramienta que fomentase, aparte de la usabilidad, su extensibilidad, para permitir el rápido despliegue e integración con los sistemas de la empresa donde se implementase.

La mayor parte de los problemas en el desarrollo del proyecto se han encontrado en la interacción con la plataforma de localización. Al tener una API muy extensa, tanto en funciones como en información, la selección de las funciones necesarias, así como los mecanismos para procesar toda la información y representar sólo la requerida, ha sido la parte más complicada, aunque no haya supuesto un gran retraso en el plan de trabajo.

Durante la consecución de este proyecto he podido poner en práctica algunos de los conocimientos adquiridos en la Universidad y he ido adquiriendo otros según iba avanzando en el proceso de desarrollo.

Este proceso de desarrollo me ha permitido seguir toda la cadena de despliegue de un nuevo servicio de valor añadido en base a los requisitos de un grupo, su análisis, desarrollo, despliegue y testeo.



## 5.1. Futuras líneas de desarrollo

El proyecto está desarrollado fomentando la extensibilidad para permitir añadir nuevas funcionalidades al desarrollo base, como por ejemplo:

- Modalidad Application Service Provider (ASP): en la que un tercero se integra con la plataforma de localización y cree grupos para distintas empresas, con administradores locales de cada una de estas empresas a las que factura en base al volumen de localizaciones realizadas mensualmente.
- Personalización de la interfaz del usuario en base a la empresa: en la que se permita que el administrador suba un logo o descripción de la empresa que gestiona.
- Localización simultánea: Permitir localizar múltiples individuos de la flota en un único mapa.
- Implementación de otras funciones de la plataforma de localización para poder realizar trackings automáticos de usuarios.



## CAPÍTULO 6: REFERENCIAS

3GPP. 2000. Functional stage 2 description of Location Services (LCS) Disponible en: <http://www.3gpp.org/ftp/Specs/html-info/23271.htm>. Consultado en Noviembre 2010.

ALARCÓN, R. 2007. Diseño orientado a objetos con UML. Editorial: Grupo Edidos.

BOOCH, G. 2006. UML: el lenguaje unificado de modelado: guía del usuario. Editorial: Addison Wesley.

CASTEJÓN, JUAN SALVADOR. 2004. Arquitectura y diseño de sistemas web modernos. Revista de Ingeniería Informática del CIIRM.

CUEVAS AGUSTÍN, G. 2003. Gestión del proceso software. Editorial: Centro de Estudios Ramón Areces, S.A.

ECLIPSE FOUNDATION. 2010. PHP Development Tools. Disponible en: <http://www.eclipse.org/pdt/> . Consultado en Noviembre 2010.

EGUÍLUZ PÉREZ, J. 2008. Introducción a JavaScript. Disponible en <http://www.librosweb.es/javascript/>, consultado en Noviembre 2010.

ESA, 2010. Estándar ESA. Disponible en [http://www.fabricadesoftware.cl/fabrica\\_documentos.php](http://www.fabricadesoftware.cl/fabrica_documentos.php), consultado en Noviembre 2010.

GENUITEC. 2010. MyEclipse Web Project Tutorial. Disponible en: <http://www.myeclipseide.com/documentation/quickstarts/webprojects/> . Consultado en Noviembre 2010.

HERNANDEZ, E. 2005. El lenguaje unificado de modelado (UML).

MySQL, 2008. MySQL 5.1 Reference Manual. Disponible en <http://dev.mysql.com/doc/>, consultado en Noviembre 2010.

NETCRAFT. 2010 November 2010 Web Server Survey. Disponible en: <http://news.netcraft.com/archives/category/web-server-survey/> . Consultado en Noviembre 2010.



OMA. 2010. Secure User Plane Location. Disponible en: [http://www.openmobilealliance.org/Technical/release\\_program/supl\\_v2\\_0.aspx](http://www.openmobilealliance.org/Technical/release_program/supl_v2_0.aspx) . Consultado en Noviembre 2010.

PORTER, MICHAEL E. 1998. Competitive Advantage: Creating and Sustaining Superior Performance. Ed. Free Press.

PHP GROUP,THE . 2010. Site español PHP. Disponible en: <http://es.php.net> . Consultado en Noviembre 2010.

PHP GROUP,THE . 2010. Manual PHP Español. Disponible en: <http://es.php.net/manual/es/>. Consultado en Noviembre 2010.

UML 2.0, 2008. The Current Official UML Specification. Disponible en <http://www.uml.org/#UML2.0>, consultado en Noviembre 2010.

MYECLIPSE. 2008. MyEclipse Web Project Tutorial. Disponible en <http://www.myeclipseide.com/documentation/quickstarts/webprojects/>, consultado en Noviembre 2010.

NEW DIGITAL GROUP INC. 2010. Smarty web site. Disponible en: <http://www.smarty.net/> . Consultado en Noviembre 2010

NIELSEN, J. 1994. Usability Inspection Methods. Editorial Wiley.

W3CSCHOOLS. 2010. AJAX Manual. Disponible en: <http://www.w3schools.com/ajax/default.asp>. Consultado en Noviembre 2010.

W3CSCHOOLS. 2010. CSS Reference. Disponible en: [http://www.w3schools.com/css/css\\_reference.asp](http://www.w3schools.com/css/css_reference.asp). Consultado en Noviembre 2010.

W3CSCHOOLS. 2010. HTML Manual. Disponible en: <http://www.w3schools.com/html/default.asp> . Consultado en Noviembre 2010.

WIKIPEDIA. 2010. Definición de Servidor Web. Disponible en: [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web) . Consultado en Noviembre 2010.



# CAPÍTULO 7: ACRÓNIMOS

|        |                                     |
|--------|-------------------------------------|
| 3GPP   | 3rd Generation Partnership Project  |
| AGPS   | Assisted GPS                        |
| API    | Application Programming Interface   |
| ASP    | Application Service Provider        |
| BTS    | Base Transceiver Station            |
| CGI    | Cell Global Identity                |
| CGI-TA | Cell Global Identity-Timing Advance |
| E-OTD  | Enhanced Observed Time Difference   |
| GPP    | Global Positioning Platform         |
| GPS    | Global Positioning System           |
| GSM    | Global System for Mobile            |
| HLR    | Home Location Register              |
| HTML   | HyperText Markup Language           |
| HTTP   | HyperText Transfer Protocol         |
| LAN    | Local Area Network                  |
| LBS    | Location Based Service              |
| LES    | Location Enabling Server            |
| LMU    | Location Measurement Units          |
| LS     | Location Server                     |
| MS     | Mobile Station                      |
| NLBS   | Near Location Based Service         |
| OMA    | Open Mobile Alliance                |
| PFC    | Proyecto Fin de Carrera             |
| SUPL   | Secure User Plane Location          |



## CAPÍTULO 7: Acrónimos

---

|      |                               |
|------|-------------------------------|
| TA   | Timing Advance                |
| TDOA | Time Difference Of Arrival    |
| TOA  | Time Of Arrival               |
| VLR  | Visitor Location Register     |
| WAP  | Wireless Application Protocol |
| WWW  | World Wide Web                |
| XML  | eXtensible Markup Language    |