



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

**Unidad Zacatenco**  
**Departamento de Computación**

**Localización y seguimiento de dispositivos móviles**

Tesis que presenta

**Hermes Francisco Montes Casiano**

para obtener el Grado de

**Maestro en Ciencias de la Computación**

Directores de la Tesis

**Dr. Pedro Mejía Alvarez**

México, D.F.

Febrero 2012

*A la memoria de mis abuelos,  
Inés Salado Acevedo y  
Marciano Casiano Factor*

*A mi mejor amiga, compañera de vida  
y esposa... Con todo mi amor para ti Ana Silvia*

*A mis padres, Francisca y Lázaro*

# Resumen

En la década de los noventa surge el campo del *Cómputo Móvil*, basado en los principios de los *Sistemas Distribuidos* y la necesidad de integrar en un tipo de arquitectura a los clientes móviles, los cuales poseen diferentes características de funcionalidad y desempeño al de las computadoras de uso convencional. El incremento en la capacidad de cómputo de los procesadores, las mejoras en los sistemas operativos modernos para dispositivos móviles y la popularidad de la banda ancha móvil, hacen de los teléfonos celulares el mejor candidato como dispositivo de cómputo móvil.

Desde el surgimiento de las redes de telefonía móvil se abrió un nicho importante para desarrollos e innovaciones tecnológicas que proveyeran a los usuarios de servicios que ayudaran a mejorar su calidad de vida o les ayudaran en la realización de sus actividades cotidianas. En los últimos años el incremento en los servicios y características, de movilidad y transferencia de información, de las redes de telefonía móvil, aunado con la proliferación de dispositivos móviles, abre un nicho importante para el desarrollo de aplicaciones que proporcionen funcionalidad a los usuarios mientras se encuentran en movimiento.

Con el fin de proporcionar una herramienta informática que permita localizar y conocer el desplazamiento geográfico de una persona (portador de un teléfono celular), en el presente trabajo de tesis se propuso diseñar e implementar un mecanismo que permita la localización y seguimiento de dispositivos móviles; motivado por la creciente popularidad del cómputo móvil y por el campo de oportunidad que generan las redes de banda ancha móvil.

El sistema desarrollado, además de proveer la funcionalidad de localización y seguimiento, proporciona mecanismos para la emisión de alertas y generación de historiales de posicionamiento. El mecanismo de emisión de alertas se basa en dos principios, la asignación de un área geográfica para desplazamiento y el establecimiento del tiempo máximo durante el cual un móvil puede estar en reposo. El mecanismo de generación de historiales de posicionamiento permite conocer la ubicación del portador de un dispositivo móvil en el pasado, con base en un intervalo de tiempo dado y, además del intervalo de tiempo, un punto de interés.

Los resultados obtenidos prueban que el sistema desarrollado es una buena herramienta para localización y seguimiento, además la precisión de los dispositivos AGPS de los teléfonos celulares utilizados permite utilizarlos en el seguimiento de personas, ya que el margen de error promedio es aceptable.



# Abstract

In the nineties appeared the Mobile Computing. It's based on the principles of Distributed Systems and the need to integrate into an architecture for mobile clients, which have different performance and characteristics of conventionally computers. The increase of computing power of processors, improvements in modern operating systems for mobile devices and the popularity of mobile broadband, make mobile phones the best candidate for mobile computing devices.

The emergence of mobile networks opened an important niche for developments and technological innovations that provided users with services. This services helped improve their quality of life and assist them in performing their daily activities. In recent years the increase in the services and features, such as: mobility and data rates of mobile networks and the proliferation of mobile devices, opens an important niche for applications development, that provide functionality for users while are in motion.

In order to provide a computational tool to locate and learn about the geographical displacement of a person (carrying a cell phone), this thesis is focused in the design and development a mechanism for location and tracking of mobile devices, motivated by the growing popularity of mobile computing and the mobile broadband networks.

The developed system provides mechanisms for alerting and generation positioning record. The mechanism of alerting is based on two principles, the setting of geofence and the setting of a maximum time that a mobile can be at rest. The generation mechanism of records positioning allows to know the past location of a mobile carrier based on a given time interval.

The results show that the developed system is a good tool for tracking and tracing, also the accuracy of AGPS devices used cell phones can be used in people tracking, since the average error is acceptable.



# Agradecimientos

- A *Dios* por llenar mi vida de bendiciones y permitirme lograr mis metas.
- A mi esposa *Ana Silvia*, quién estuvo a mi lado desde el inicio de mis estudios profesionales. Gracias por escucharme, por motivarme siempre a seguir adelante... Gracias por tu comprensión... Gracias por se la fuente de mi inspiración.
- No hay palabras para expresar el agradecimiento a quienes me dieron la vida, *Francisca* y *Lázaro*, podría escribir un libro entero y no terminaría... Gracias por haberme apoyado hasta el final, gracias por su amor y enseñanzas.
- A mis tíos, quienes siempre me brindaron su apoyo. Especialmente a *Serafín*, quien más que un tío a sido un padre para mí. Gracias por creer siempre en mi y apoyarme incondicionalmente en la realización de mis metas.
- A mi asesor de tesis el *Dr. Pedro Mejía Álvarez*, por haberme transmitido sus conocimientos y por su dirección en éste trabajo, ya que sin su apoyo no habría sido posible.
- A la Dra. Sonia y al Dr. Jorge Buenabad, por fungir como revisores de mi trabajo y por los conocimientos transmitidos en los cursos que me impartieron.
- Al *CONACyT* por la beca otorgada para la realización de mi maestría.
- Al *CINVESTAV-IPN* y a su *Departamento de Computación* por haberme aceptado en su Programa de Maestría en Ciencias de la Computación y por los medios proporcionados para la realización de la misma.
- A *José Jaime López Rabadán*, quien me ha brindado su amistad, me ha apoyado incondicionalmente y me ha enseñado con el ejemplo que en la vida hay que esforzarse para realizar lo que uno se propone.
- A *Carlos Silva Sánchez*, quien paso de ser mi maestro a ser un gran amigo.
- A la familia *Cabello Casiano* por ese apoyo y cariño que siempre me brindó cada uno de sus miembros... Les estaré eternamente agradecido.
- A Sofi la secretaria del departamento, por su apoyo.
- A Ivan, pro brindarme su amistad desde mi llegada a la hermosa Ciudad de México.

- A todos mis amigos y familiares, con quienes he compartido grandes momentos.



# Índice general

<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>v</b>
<b>Agradecimientos</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación de la tesis . . . . .	2
1.2. Objetivo de la tesis . . . . .	3
1.2.1. Objetivos específicos . . . . .	3
1.3. Organización de la tesis . . . . .	4
<b>2. Arquitecturas de software</b>	<b>5</b>
2.1. Arquitectura en capas . . . . .	5
2.2. Modelo orientado a objetos . . . . .	6
2.3. Modelo de repositorio . . . . .	7
2.4. Arquitecturas heterogéneas y de dominio específico . . . . .	9
<b>3. Aplicaciones en la Web</b>	<b>11</b>
3.1. Arquitectura Cliente Servidor . . . . .	11
3.2. Identificadores . . . . .	12
3.3. Protocolo de Transferencia de Hipertexto . . . . .	12
3.4. Protocolo de Transferencia de Hipertexto sobre SSL . . . . .	14
3.4.1. Comparación entre HTTP y HTTPS . . . . .	15
3.5. Base de las aplicaciones Web . . . . .	15
3.5.1. HTML . . . . .	15
3.5.2. Scripting y Ajax . . . . .	16
3.6. Aplicaciones Ricas de Internet . . . . .	16
<b>4. Localización y seguimiento de dispositivos móviles</b>	<b>19</b>
4.1. Cómputo Móvil . . . . .	19
4.1.1. Dispositivos móviles . . . . .	20
4.1.2. Tecnología de Comunicación Móvil . . . . .	20
4.1.3. Dimensiones del cómputo móvil . . . . .	21

4.1.3.1.	Conocimiento de la localización . . . . .	23
4.1.3.2.	Conexión de red con calidad de servicio (QoS) . . . . .	23
4.1.3.3.	Fuente de alimentación limitada . . . . .	24
4.2.	Tecnologías de localización móvil . . . . .	24
4.2.1.	Sistemas de Posicionamiento Satelital . . . . .	24
4.2.2.	Sistemas de Posicionamiento Celular . . . . .	25
4.3.	Sistemas de Información Geográfica . . . . .	25
4.4.	Trabajo relacionado . . . . .	26
4.4.1.	Sistema de seguimiento vehicular en Tiempo Real . . . . .	26
4.4.2.	Sistema de seguimiento para personas mayores basado en AGPS . . . . .	28
4.4.3.	Desarrollo de un Sistema de Tecnologías de Cómputo, Comunicaciones y Geolocalización . . . . .	30
<b>5.</b>	<b>Análisis y diseño de la propuesta</b>	<b>33</b>
5.1.	Modelo de comportamiento . . . . .	33
5.2.	Arquitectura general de comunicación . . . . .	37
5.3.	Arquitectura de subsistemas . . . . .	37
5.3.1.	Aplicación Web . . . . .	38
5.3.2.	Cliente móvil . . . . .	41
5.3.3.	Cliente Web . . . . .	42
5.3.3.1.	Componente para localización: LocationMapsLibrary . . . . .	43
<b>6.</b>	<b>Plataforma y tecnologías de desarrollo</b>	<b>47</b>
6.1.	Plataforma y tecnologías de desarrollo . . . . .	47
6.1.1.	Aplicación web . . . . .	47
6.1.1.1.	Frameworks de desarrollo . . . . .	47
6.1.1.2.	Almacenamiento y acceso a datos . . . . .	49
6.1.2.	Cliente móvil . . . . .	50
6.2.	Implementación de los principales algoritmos . . . . .	53
6.2.1.	Emisión de alertas . . . . .	53
6.2.2.	Historiales de posicionamiento . . . . .	54
<b>7.</b>	<b>Pruebas y resultados</b>	<b>57</b>
7.1.	Tiempo de respuesta y precisión del AGPS . . . . .	58
7.2.	Localización y seguimiento . . . . .	62
7.3.	Emisión de alertas . . . . .	64
<b>8.</b>	<b>Conclusión y trabajo futuro</b>	<b>69</b>
8.1.	Conclusiones . . . . .	69
8.2.	Trabajo futuro . . . . .	72

<b>A. Análisis de Requerimientos</b>	<b>75</b>
A.1. Definición de requerimientos . . . . .	75
A.1.1. Definición de requerimientos básicos del LoSeGui . . . . .	75
A.1.2. Definición de Requerimientos Funcionales del LoSeGui . . . . .	76
A.1.2.1. Control de acceso . . . . .	77
A.1.2.2. Usuarios Administrador y Supervisor . . . . .	77
A.1.2.3. Requerimientos funcionales del usuario Administrador . . . . .	77
A.1.3. Definición de requerimientos no funcionales del LoSeGui . . . . .	80
A.2. Especificación de los requerimientos del LoSeGui . . . . .	80
A.2.1. Especificación de requerimientos funcionales del LoSeGui . . . . .	80
A.2.2. Requerimientos de control de acceso . . . . .	81
A.2.2.1. Requerimientos para el usuario Administrador y Supervisor . . . . .	81
A.2.3. Requerimientos para el usuario Administrador . . . . .	88
A.3. Diccionario de Datos . . . . .	91
A.3.1. Rol Usuario . . . . .	93
A.3.2. Usuario . . . . .	93
A.3.3. Dispositivos Móviles . . . . .	93
A.3.3.1. Dispositivo en Movimiento . . . . .	94
A.3.3.2. Región . . . . .	94
A.3.3.3. Posición . . . . .	95
A.3.3.4. Alerta . . . . .	95
<b>B. Casos de Uso Principales</b>	<b>97</b>
B.1. UC1 Monitorear Dispositivo Móvil . . . . .	98
B.1.1. Descripción completa . . . . .	98
B.1.2. Atributos importantes . . . . .	98
B.1.3. Trayectorias . . . . .	98
B.1.4. Diagrama de Clases . . . . .	100
B.1.5. Diagrama de Secuencia . . . . .	101
B.2. UC2 Mostrar/Ocultar Posición . . . . .	102
B.2.1. Descripción completa . . . . .	102
B.2.2. Atributos importantes . . . . .	102
B.2.3. Trayectorias . . . . .	102
B.2.4. Diagrama de Clases . . . . .	104
B.2.5. Diagrama de Secuencia . . . . .	105
B.3. UC3 Mostrar/Ocultar Trayectoria . . . . .	106
B.3.1. Descripción completa . . . . .	106
B.3.2. Atributos importantes . . . . .	106
B.3.3. Trayectorias . . . . .	106
B.3.4. Diagrama de Clases . . . . .	108
B.3.5. Diagrama de Secuencia . . . . .	109
B.4. UC4 Generar Mapa Temático Historial de Posicionamiento . . . . .	110
B.4.1. Descripción completa . . . . .	110

B.4.2.	Atributos importantes . . . . .	110
B.4.3.	Trayectorias . . . . .	110
B.4.4.	Diagrama de Clases . . . . .	112
B.4.5.	Diagrama de Secuencia . . . . .	113
B.5.	UC5 Generar Mapa Temático Punto de Interés . . . . .	114
B.5.1.	Descripción completa . . . . .	114
B.5.2.	Atributos importantes . . . . .	114
B.5.3.	Trayectorias . . . . .	114
B.5.4.	Diagrama de Clases . . . . .	116
B.5.5.	Diagrama de Secuencia . . . . .	117
B.6.	UC6 Obtener Punto de Interés . . . . .	118
B.6.1.	Descripción completa . . . . .	118
B.6.2.	Atributos importantes . . . . .	118
B.6.3.	Trayectorias . . . . .	118
B.7.	UC7 Configurar Políticas . . . . .	119
B.7.1.	Descripción completa . . . . .	119
B.7.2.	Atributos importantes . . . . .	119
B.7.3.	Trayectorias . . . . .	119
B.7.4.	Diagrama de Clases . . . . .	120
B.7.5.	Diagrama de Secuencia . . . . .	121

# Índice de figuras

2.1. Modelo en 3 capas, ejemplo de arquitecturas en multiples capas [31]. . . . .	6
2.2. Tipos abstractos de datos y objetos. . . . .	7
2.3. Esquema de un modelo de repositorio [23] . . . . .	8
3.1. Relación entre identificador, recurso y representación . . . . .	13
3.2. Funcionamiento del protocolo HTTP . . . . .	14
3.3. Modelo tradicional de Aplicación Web vs basado en Ajax . . . . .	17
4.1. Convergencia de los sistemas de comunicación móvil . . . . .	22
4.2. Dimensiones de la movilidad . . . . .	23
4.3. Arquitectura de alto nivel del sistema . . . . .	27
4.4. Arquitectura de la Unidad en el Vehículo . . . . .	28
4.5. Arquitectura del Servidor de Seguimiento . . . . .	28
4.6. Arquitectura de alto nivel del sistema . . . . .	29
4.7. Arquitectura del sistema, basada en el modelo cliente servidor. . . . .	31
5.1. Diagrama general de los módulos de la propuesta. . . . .	35
5.2. Caso de uso del módulo de control de acceso. . . . .	35
5.3. Casos de uso del módulo de localización. . . . .	36
5.4. Casos de uso del módulo de seguimiento. . . . .	36
5.5. Casos de uso del módulo de gestión. . . . .	36
5.6. Arquitectura de comunicación del sistema de localización y seguimiento de dispositivos móviles . . . . .	38
5.7. Tecnologías utilizadas para el desarrollo de cada componente del sistema. . . . .	39
5.8. Arquitectura general de la aplicación con base en el patrón MVC. . . . .	40
5.9. Capa modelo de MVC en 3 capas . . . . .	41
5.10. Diagrama de la arquitectura general del cliente móvil. . . . .	42
5.11. Componentes del cliente web . . . . .	43
5.12. Clases del componente para localización basado en la interfaz de programación de GoogleMaps . . . . .	45
6.1. Tecnologías utilizadas para la implementación de la aplicación web . . . . .	48
6.2. Principales componentes del Sistema Operativo Android . . . . .	52
6.3. Capas de iOS . . . . .	53

7.1.	Gráfica donde se ilustra el TTFB del AGPS de los dispositivos utilizados. . .	59
7.2.	Ilustración de método adoptado para medir la precisión del AGPS. . . . .	60
7.3.	Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones favorables a la tecnología de localización. . . . .	61
7.4.	Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones desfavorables a la tecnología de localización. . . . .	63
7.5.	Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones favorables a la tecnología de localización utilizada. . . . .	63
7.6.	Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones no favorables a la tecnología de localización. . . . .	64
7.7.	Gráfica en la que se muestra el tiempo requerido para el proceso de posicionamiento. . . . .	65
7.8.	Gráfica en la que se muestra el tiempo que le toma a la aplicación visualizar la ubicación y trayectoria de los dispositivos móviles en movimiento. . . . .	65
7.9.	Gráfico en el que se muestra el tiempo requerido para emitir una alerta de región abandonada. Las regiones de prueba son polígonos regulares, de pequeña cobertura y con un número de vértices < 10. . . . .	66
7.10.	Gráfico en el que se muestra el tiempo requerido para emitir una alerta de región abandonada. Las regiones de prueba son polígonos completamente irregulares. . . . .	67
7.11.	Número de vértices de las geocercas adoptadas para la prueba con polígonos irregulares. . . . .	67
A.1.	Diagrama entidad relación de la base de datos del LoSeGui. . . . .	92
B.1.	Diagrama general de Casos de Uso . . . . .	97
B.2.	Diagrama de Clases del CU Monitorear Dispositivo Móvil . . . . .	100
B.3.	Diagrama de Secuencia de CU Monitorear Dispositivo . . . . .	101
B.4.	Diagrama de Clases Mostrar/Ocultar Posición . . . . .	104
B.5.	Diagrama de Secuencia Mostrar/Ocultar Posición . . . . .	105
B.6.	Diagrama de Clases Mostrar/Ocultar Trayectoria . . . . .	108
B.7.	Diagrama de Secuencia Mostrar/Ocultar Trayectoria . . . . .	109
B.8.	Diagrama de Clases Mapa Temático Historial Posicionamiento . . . . .	112
B.9.	Diagrama de Secuencia Mapa Temático Historial Posicionamiento . . . . .	113
B.10.	Diagrama de Clases Mapa Temático Historial Posicionamiento con Punto de Interés . . . . .	116
B.11.	Diagrama de Secuencia Mapa Temático Historial Posicionamiento con Punto de Interés . . . . .	117
B.12.	Diagrama de Clases Configurar Políticas . . . . .	120
B.13.	Diagrama de Secuencia Configurar Políticas . . . . .	121

# Índice de tablas

2.1. Ventajas y desventajas del modelo arquitectónico de repositorio [39] . . . . .	8
4.1. Comparativa de las redes de comunicación móvil . . . . .	21
4.2. Sistemas de localización celular . . . . .	26
6.1. Características principales de PostgreSQL . . . . .	49
6.2. Visión general de las funciones soportadas por cada Sistema Gestor de Base de Datos. . . . .	50
7.1. Tiempo para obtener la primera posición (TTFF). . . . .	58
7.2. Categorización del margen de error para la prueba de precisión del AGPS. . . . .	60
7.3. Resultados obtenidos en la prueba de precisión realizada bajo condiciones ideales. . . . .	60
7.4. Resultados obtenidos en la prueba de precisión realizada bajo condiciones desfavorables. . . . .	62
A.1. Definición de Requerimientos Básicos del LoSeGui. . . . .	76
A.2. RF: Tipos de usuarios del LoSeGui . . . . .	77
A.3. RF: servicios para los usuarios <i>Administrador y Supervisor</i> . . . . .	77
A.4. RF: servicios los usuarios <i>Administrador y Supervisor</i> . . . . .	78
A.5. RF: servicios para el usuario Administrador . . . . .	79
A.6. Requerimientos No Funcionales del LoSeGui . . . . .	80





# Capítulo 1

## Introducción

En la década de los noventa surge el campo del *Cómputo Móvil*, basado en los principios de los *Sistemas Distribuidos* y la necesidad de integrar en un tipo de arquitectura a los clientes móviles, los cuales poseen diferentes características de funcionalidad y desempeño al de las computadoras de uso convencional [26]. En los últimos años, el crecimiento y la popularidad de la telefonía móvil fue más significativa de lo previsto. El incremento en la capacidad de cómputo de los procesadores, las mejoras en los sistemas operativos modernos para dispositivos móviles y la popularidad de la banda ancha móvil, hacen de los teléfonos celulares el mejor candidato como dispositivo de cómputo móvil; capaces de realizar sofisticadas aplicaciones científicas [43].

Modernas plataformas para dispositivos móviles, como iPhone [38] y Android [1], incluyen una amplia gama de bibliotecas útiles e *Interfaces de Programación de Aplicaciones* (API, *Application Programming Interface*, por sus siglas en inglés) para desarrolladores. Las bibliotecas permiten acceder a los sensores internos, como acelerómetro o GPS. Estas bibliotecas y *API's* permiten a los desarrolladores implementar nuevas aplicaciones y controlar el teléfono con mayor libertad y flexibilidad [43].

Desde el surgimiento de las redes de telefonía móvil se abrió un nicho importante para desarrollos e innovaciones tecnológicas que proveyeran a los usuarios de servicios que ayudaran a mejorar su calidad de vida o les ayudaran en la realización de sus actividades cotidianas. En los últimos años, el incremento en los servicios y características de movilidad y transferencia de información, de las redes de telefonía móvil, aunado con la proliferación de dispositivos móviles, ha abierto un nicho importante para el desarrollo de aplicaciones que proporcionen funcionalidad a los usuarios mientras se encuentran en movimiento.

Por otro lado, se clasifica como Sistema de Información Geográfica (GIS, por sus siglas en inglés) a los Sistemas de Información que además incorporan información georeferenciada. Un GIS se define como un conjunto de herramientas para recolectar, almacenar, obtener, transformar y desplegar información espacial de la Tierra para propósitos específicos; son utilizados en ingeniería civil, geología, inteligencia militar, control de tráfico vial, entre otros.

En los GIS es imprescindible el uso del Sistema de Posicionamiento Global (GPS, por sus siglas en inglés), el cual es una tecnología madura para localización precisa en ambientes al aire libre y de mayor uso en la actualidad. El GPS tiene la capacidad de proveer información de posición y tiempo a los usuarios del sistema. El GPS Asistido (AGPS) es una solución híbrida que utiliza información provista por los satélites GPS y la red celular. El AGPS se auxilia de fuentes externas, como servidores de asistencia o referencias de red, para ayudar al receptor GPS a disminuir el tiempo de respuesta y el margen de error.

## 1.1. Motivación de la tesis

Existen diversas aplicaciones enfocadas en la localización de dispositivos móviles, algunas resultado de trabajos de investigación como las expuestas en la sección 4.4 y muchas otras que forman parte de una gama de servicios ofertados por empresas dedicadas a la localización vehicular, por ejemplo. Sin embargo, en la mayoría de esos casos se utilizan dispositivos sofisticados y diseñados específicamente para su localización y costosas infraestructuras de redes privadas.

En 2007 el número de usuarios de telefonía celular ascendía a 3 mil millones 300 mil usuarios en todo el mundo, lo que representaba aproximadamente la mitad de la población en el mundo [28], mientras que en el año 2009 la cifra de suscriptores había llegado a los 4 mil millones [43]. El auge de la telefonía y las redes de comunicación han traído consigo la necesidad de portar con un mayor número de aplicaciones en el teléfono celular y, además, cada vez más parecidas a las aplicaciones de escritorio. El teléfono celular se ha convertido en un excelente candidato como dispositivo de cómputo móvil debido a que, por un lado, se ha convertido en un dispositivo de uso diario y forma parte integral del estilo de vida de la sociedad moderna, al grado que su presencia en el entorno ya no es tan notoria como su ausencia y, por otro, a que los fabricantes cada vez los producen con mayores capacidades y aditamentos como: cámara digital, acelerometro, pantallas táctiles, GPS, entre otros. En los teléfonos celulares se pretende introducir todas aquellas aplicaciones que el portador del mismo pueda necesitar en el sitio en que se encuentre.

En el ámbito de la localización *Google* provee una plataforma denominada *Latitud*, misma que se enfoca en la localización de dispositivos móviles en línea. Sin embargo, dicha aplicación no tiene la funcionalidad que permita realizar el seguimiento de un dispositivo en desplazamiento y solo permite conocer la ubicación actual, lo que impide tener un historial de posicionamiento de los dispositivos de interés.

Además de lo expuesto hasta el momento, el dominio de aplicación en que se encuentra inmersa una aplicación para localización y seguimiento es extremadamente complejo; pretender realizar una aplicación de esa índole sobre dispositivos móviles impone retos y oportunidades que en la literatura se les denomina *Dimensiones de la Movilidad*:

1. Conocimiento de localización,
2. Conexión de red con calidad de servicio (QoS),
3. Capacidad limitada de los dispositivos (en particular almacenamiento y CPU),
4. Fuente de alimentación limitada,
5. Amplia variedad de interfaces de usuario,
6. Proliferación de plataformas,
7. Transacciones activas

Por ello, se considera factible el diseño e implementación de una plataforma que dote a los usuarios de la telefonía móvil de servicios de localización y seguimiento con herramientas y medios con los que ya está familiarizado y forman parte de su vida, como la internet y los teléfonos celulares.

## 1.2. Objetivo de la tesis

El objetivo del presente trabajo de tesis es diseñar e implementar una plataforma basada en web para localización y seguimiento de dispositivos móviles, que haciendo uso de la red de telefonía celular, permita localizar y monitorear el desplazamiento del portador del dispositivo.

### 1.2.1. Objetivos específicos

Para el cumplimiento del objetivo propuesto fue necesario dividirlo en los siguientes objetivos específicos:

- Diseñar e implementar la arquitectura de los dos componentes principales: cliente móvil y servidor de aplicación.
- Diseñar e implementar la aplicación del servidor de aplicación.
- Diseñar e implementar las aplicaciones para los dispositivos móviles Android y iPhone.
- Diseñar e implementar un mecanismo de comunicación que permita la transferencia de información entre el servidor de aplicación y los clientes móviles.
- Diseñar e implementar un módulo de despliegue de información geográfica utilizando el servicio de *Google Maps*.
- Diseñar e implementar un mecanismo de alertas con base en el uso de geocercas e intervalos de tiempo.

- Diseñar e implementar una base de datos geográfica para almacenar el historial de posicionamiento y las regiones de recorridos (geocercas).
- Diseñar e implementar un mecanismo para la generación de historiales de posicionamiento.

### 1.3. Organización de la tesis

El contenido del presente documento de tesis está distribuido en 8 capítulos y 2 anexos. Del apartado 2 al 4 se abordan los conceptos generales y trabajos relacionado con el la presente tesis. Los 3 capítulos siguientes se centran en presentar el diseño de la propuesta, su implementación, pruebas y resultados obtenidos. Finalmente en el capítulo 8 se muestran las conclusiones y trabajo futuro. A continuación se detalla el contenido de cada capítulo:

- *Capítulo 2:* expone algunos de los modelos arquitectónicos más representativos, los cuales son ampliamente utilizados. El propósito es ilustrar el rico espacio de opciones de arquitecturas existentes e indicar cuáles son las ventajas y desventajas más importantes, para tomarlas en cuenta en el momento de la elección.
- *Capítulo 3:* en éste capítulo se presenta el marco contextual de las aplicaciones Web, los protocolos y especificaciones sobre los cuales operan y cuales son los servicios que proveen.
- *Capítulo 4:* en éste capítulo se estudian de manera general los aspectos más importantes de la localización de dispositivos móviles. En primera instancia se aborda el campo del computo móvil, posteriormente se realiza una reseña de las etapas de evolución de las redes de telefonía, se presentan dos vertientes de las tecnologías de localización móvil y el trabajo relacionado a éste trabajo de tesis.
- *Capítulo 5:* se presenta la arquitectura propuesta para localización y seguimiento de dispositivos móviles y el modelo general de comportamiento de todo el sistema.
- *Capítulo 6:* en éste capítulo se exponen los detalles más sobresalientes de la implementación, desde la elección de la plataforma y tecnologías de desarrollo, hasta los algoritmos en los que recae la principal funcionalidad que provee el sistema.
- *Capítulo 7:* aquí se exponen los resultados obtenidos del desarrollo, entre los que destacan: la precisión y tiempos de respuesta de los AGPS de los dispositivos móviles utilizados y tiempo de retardo para la localización de los dispositivos móviles.
- *Capítulo 8:* en éste capítulo se presenta la conclusión del trabajo de tesis, en el cual se resaltan los principales resultados obtenidos y también se presenta como trabajo futuro todo aquello que puede agregarse al trabajo o lo que es susceptible de mejoras.

# Capítulo 2

## Arquitecturas de software

En el presente capítulo se exponen algunos de los modelos arquitectónicos más representativos y ampliamente utilizados. El propósito es ilustrar el rico espacio de opciones de arquitecturas existentes e indicar cuáles son las ventajas y desventajas más importantes, para tomarlas en cuenta en el momento de la elección.

Una arquitectura se puede conceptualizar como un gráfico en el cual los nodos representan los componentes de la arquitectura y los arcos representan las conexiones que existen entre los componentes [23]. Un conector puede representar interacciones variadas, como la llamada a un procedimiento, emisión de eventos, consultas de bases de datos, una tubería, entre otras. Un estilo arquitectónico define a una familia de sistemas en términos de un patrón de organización estructural. Específicamente, un estilo arquitectónico determina el vocabulario de los componentes y conectores que se pueden utilizar, junto con un conjunto de restricciones de como se pueden combinar.

### 2.1. Arquitectura en capas

Un sistema con arquitectura en capas está organizado jerárquicamente. Cada capa proporciona servicios a la capa de encima y funge como cliente de la capa de abajo (ver Figura 2.1). En algunos sistemas de capas, las capas interiores quedan ocultas para todas las demás capas, excepto a la capa externa adyacente, a excepción de ciertas funciones cuidadosamente seleccionadas. En una arquitectura en capas, las conexiones son definidas por los protocolos que determinan cómo las capas interactúan. Las restricciones topológicas agregan limitantes a las interacciones entre capas adyacentes [33][23].

Los ejemplos más conocidos que se basan en una arquitectura en capas son los protocolos de comunicación, donde cada capa proporciona soporte para la comunicación en un cierto nivel de abstracción. Las áreas de aplicación más comunes con este estilo son los sistemas de bases de datos y los sistemas operativos.

La arquitectura en capas tiene varias propiedades. En primer lugar, provee una capacidad

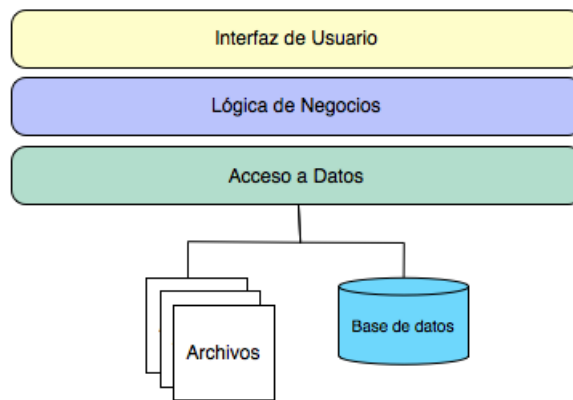


Figura 2.1: Modelo en 3 capas, ejemplo de arquitecturas en múltiples capas [31].

de diseño basada en la premisa de incrementar el nivel de abstracción con cada capa, lo que permite dividir un problema complejo en una secuencia de pasos graduales. En segundo lugar, apoya la mejora, ya que cada capa interactúa con su capa superior e inferior. Los cambios en la funcionalidad de una capa debe afectar a un máximo de dos capas. En tercer lugar, favorece la reutilización, ya que diferentes implementaciones de la misma capa se pueden utilizar indistintamente, siempre y cuando tengan las mismas interfaces de comunicación. Lo que origina la necesidad de definir una misma interfaz para diferentes implementaciones de la misma capa [23].

Por otro lado los sistemas de capas también tienen desventajas. No todos los sistemas son fáciles de estructurar en forma de capas. Al mejorar el rendimiento de un sistema se podría incrementar el nivel de acoplamiento entre las funciones de alto nivel y su correspondiente implementación en los niveles inferiores. Además, encontrar el nivel de abstracción correcto no es tarea fácil. Por ejemplo, los modelos de comunicación presentan dificultades a la hora de asignar los protocolos existentes en la norma ISO, ya que algunos protocolos forman puentes entre varias capas.

## 2.2. Modelo orientado a objetos

En este estilo la representación de la información y su asociación con operaciones primitivas está encapsulada en un tipo de dato abstracto o objeto (ver Figura 2.2). Los componentes en este tipo de arquitectura son objetos débilmente acoplados.

Los objetos son ejemplos de un tipo de componente que se denomina gestor, ya que es responsable de preservar la integridad de un recurso. Los objetos interactúan mediante llamadas a los servicios definidos en sus interfaces de comunicación (métodos). El modelo orientado a objetos tiene un principio importante al que se le denomina encapsulamiento y consiste en que: *un objeto es responsable de preservar su estado y mantener oculta la implementación*

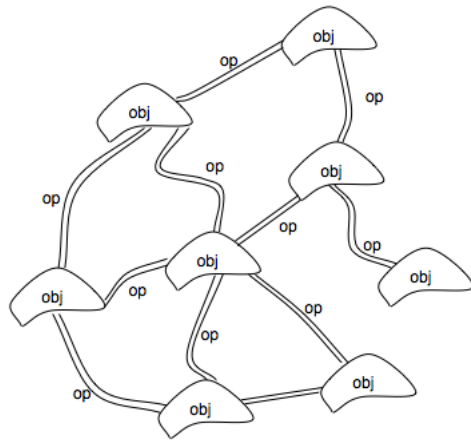


Figura 2.2: Tipos abstractos de datos y objetos.

*de sus atributos y permitir el acceso a ellos sólo a través de su interfaz de comunicación [23].*

Las ventajas de un enfoque orientado a objetos son bien conocidas. Puesto que los objetos están débilmente acoplados, la implementación de objetos se puede modificar sin afectar a otros objetos. A menudo los objetos representan entidades del mundo real, por lo que la estructura del sistema es bastante comprensible [39]. Sin embargo, el enfoque orientado a objetos tiene ciertas desventajas. Para interactuar, los objetos deben hacer referencia explícita al nombre y a la interfaz de otros objetos. Si se requiere cambiar una interfaz para satisfacer los cambios del sistema, se debe evaluar el efecto de ese cambio sobre todos los objetos que interactúan con el objeto modificado [39].

## 2.3. Modelo de repositorio

En el modelo de repositorio se distinguen dos tipos de componentes: una estructura de datos central (repositorio) que representa el estado actual y una colección de componentes independientes que operan con el almacén central (ver Figura 2.3). La interacción entre el repositorio y los componentes externos puede variar significativamente entre cada subsistema [23].

Este modelo arquitectónico es usado por sistemas que utilizan grandes cantidades de datos y en donde la información que genera un subsistema es utilizada o requerida por otro [39]. En la tabla 2.1 se muestran las ventajas y desventajas presentes en éste estilo arquitectónico.

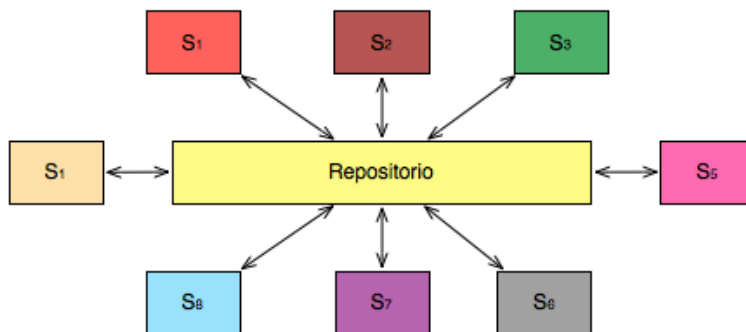


Figura 2.3: Esquema de un modelo de repositorio [23]

Ventajas	Desventajas
Es una forma eficiente de compartir grandes cantidades de datos. No existe la necesidad de transmitir datos explícitamente de un subsistema a otro.	Los subsistemas deben estar acordes al modelo de depósito de datos. De forma inevitable, esto es un compromiso entre las necesidades específicas de cada herramienta.
Los subsistemas que producen información no necesitan saber cómo ésta es utilizada por otros subsistemas.	Si se genera un gran volumen de información, será difícil evolucionar si se ha acordado un modelo de datos, ya que traducir el modelo existente a uno nuevo será muy costoso.
Las actividades de respaldo, seguridad, control de acceso y recuperación de errores están centralizadas	Varios subsistemas tienen diferentes requerimientos de políticas de seguridad, recuperación y respaldo. El modelo de repositorio fuerza a la misma política para todos los subsistemas.
El modelo de compartición es visible a lo largo del esquema. De forma directa se integran las nuevas herramientas puesto que son compatibles con el modelo de repositorio.	Es difícil distribuir el repositorio en varias máquinas. Aunque es posible distribuir un repositorio centralizado lógicamente, habrá problemas con la redundancia e inconsistencia de los datos.

Tabla 2.1: Ventajas y desventajas del modelo arquitectónico de repositorio [39]



## 2.4. Arquitecturas heterogéneas y de dominio específico

En las secciones precedentes se abordaron modelos arquitectónicos puros, sin embargo la mayoría de los sistemas suelen incluir una combinación de varios modelos. Hay diferentes formas en que los modelos arquitectónicos se pueden combinar. Una forma es a través de la jerarquía. Un componente de un sistema puede estar organizado de acuerdo a un modelo arquitectónico y tener una estructura interna con base en otro completamente diferente.

Una segunda manera de combinar modelos arquitectónicos, es permitir a un componente utilizar diferentes conectores arquitecturales. Por ejemplo, un componente puede interactuar con la interfaz de comunicación de otros a través de tuberías y por otro lado recibir sentencias de control por medio de *sockets*.

Por otro lado, también se pueden utilizar modelos arquitectónicos específicos para un dominio de aplicación particular, los cuales se clasifican en dos tipos:

1. *Modelos genéricos* que son abstracciones de varios sistemas reales, los cuales encapsulan las características principales de esos sistemas.
2. *Modelos de referencia* que son modelos abstractos y describen a una clase mayor de sistemas.

Por supuesto, no existe una distinción rígida entre estos tipos de modelos. Algunas veces, los modelos genéricos también sirven de modelos de referencia. Aquí se hace una distinción entre ellos debido a que los modelos genéricos se pueden utilizar directamente en el diseño. Los modelos de referencia se utilizan normalmente para comunicar conceptos de dominio y comparar posibles arquitecturas.



# Capítulo 3

## Aplicaciones en la Web

En este capítulo se presenta el marco contextual de las aplicaciones Web, los protocolos y especificaciones sobre los cuales operan y cuales son los servicios que proveen. La WWW (World Wide Web, por sus siglas en inglés) o simplemente Web es una red de recursos de información y se basa en tres mecanismos para que los recursos estén a disposición de la mayor audiencia posible: un esquema con nomenclatura uniforme para localizar recursos, protocolos que permitan el acceso a dichos recursos e hipertexto para facilitar la navegación entre ellos [25].

### 3.1. Arquitectura Cliente Servidor

Una arquitectura de software o de un sistema de cómputo es la estructura o estructuras del sistema, que incluye sus componentes, propiedades y las relaciones que existen entre ellos [32]. La Web esta basada en una arquitectura cliente servidor, misma que agrupa un conjunto de elementos que efectúan procesos distribuidos y cómputo cooperativo.

Las aplicaciones *cliente-servidor* se diferencian por la clara división entre el trabajo realizado por el cliente y el servidor [29]. El modelo cliente-servidor es un modelo de sistemas distribuido que muestra como los datos y el procesamiento se distribuyen a lo largo de varios procesadores [39]. Los componentes principales de este modelo son:

- Un conjunto de servidores independientes que ofrecen servicios a otros subsistemas.
- Un conjunto de clientes que llaman a los servicios ofrecidos por los servidores.
- Una red que permite a los clientes comunicarse con los servidores para acceder a estos servicios.

La ventaja más importante del modelo cliente-servidor es que es una arquitectura distribuida. Además de la fácil incorporación de un nuevo servidor e integrarlo con el resto del sistema o actualizar de forma transparente sin afectar otras partes del sistema. Los sistemas en red se pueden utilizar de forma efectiva con muchos procesadores distribuidos. [39].

## 3.2. Identificadores

Uno de los objetivos de la Web, desde su creación, ha sido la de construir una comunidad global en la que cualquier persona pueda compartir información con cualquier otra. Para lograr este objetivo, la web hace uso de un sistema de identificación global única: la URI (Identificador Uniforme de Recurso, por sus siglas en inglés). Las URI son una piedra angular de la arquitectura Web, que facilita la identificación de que es común en toda la Web. Un URI provee, a través de una secuencia de caracteres, un medio simple y extensible de identificación [42].

La *uniformidad* provee varios beneficios, ya que permite identificar recursos de diferentes tipos en un mismo contexto, incluso cuando los mecanismos de acceso a ello sean distintos. Esto permite una interpretación uniforme de la sintaxis utilizada en diferentes identificadores. Además, facilita la introducción de nuevos identificadores de recursos sin interferir con los existentes.

Lo que podría ser un *recurso* dentro de la Web no se encuentra bien delimitado, es decir, se utiliza en un sentido general para referirse a todo lo que pudiera ser identificado por una URI (ver Figura 3.1): imagen, video, hipertexto, correo electrónico, etc. Un URI identifica de forma inequívoca un recurso.

Un *identificador* incorpora la información necesaria para distinguir, dentro de los demás recursos existentes en el mismo contexto, que está siendo identificado. Un URI consta de las siguientes partes:

- Esquema: identifica el protocolo de acceso al recurso (http:, mailto:, ftp:, etc).
- Autoridad: identifica el dominio (host) en el que se ubica el recurso.
- Ruta: contiene la ubicación jerárquica del recurso.
- Cadena de Consulta: información con estructura no jerárquica (usualmente pares “clave=valor”) que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres. El comienzo de este componente se indica mediante el carácter ‘?’ y termina con el carácter ‘#’.
- Fragmento: permite la identificación de un recurso secundario de manera indirecta, en relación con el recurso especificado en la ruta. La identificación del recurso secundario puede ser alguna parte o subconjunto del recurso principal.

## 3.3. Protocolo de Transferencia de Hipertexto

En el mundo creciente de la Internet, el incremento diario de las aplicaciones que operan sobre ella y los diversos ámbitos sobre los que éstas se aplican, son los principales causantes

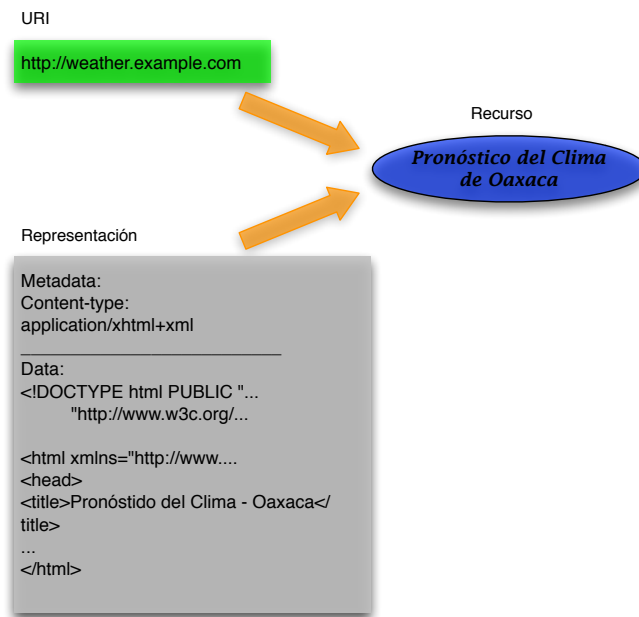


Figura 3.1: Relación entre identificador, recurso y representación

del incremento en el número de usuarios de la red de redes y con ello un incremento también de sus necesidades.

El Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés) es un protocolo a nivel de aplicación para sistemas de información distribuida, colaborativa e hipermedia [6]. HTTP es un protocolo utilizado fundamentalmente para acceder a datos en la Web. Funciona como una combinación de los protocolos FTP y SMTP. HTTP utiliza los servicios del protocolo TCP en el puerto 80 [21].

HTTP es un protocolo de petición - respuesta y a pesar de que utiliza los servicios de TCP es un protocolo sin estado. HTTP tiene el siguiente funcionamiento (ver Figura 3.2):

1. Un cliente envía una solicitud (*request*) al servidor con la siguiente información:
  - Un método
  - Un URI
  - La versión del protocolo HTTP que solicita utilizar (por ejemplo HTTP/1.0 o HTTP/1.1)
  - MIME con: modificadores de la petición, información del cliente y, posiblemente, información sobre la conexión con el servidor.
2. El servidor responde con:



Figura 3.2: Funcionamiento del protocolo HTTP

- Una línea de estado, que incluye: versión del protocolo a utilizar y un código de estado.
- MIME con meta información e información del servidor.
- El recurso solicitado.

### 3.4. Protocolo de Transferencia de Hipertexto sobre SSL

La internet y sus aplicaciones han creado una nueva forma de vida, en la cual uno de los requerimientos más importantes es la seguridad, debido a que en su esencia la Internet es una red sumamente insegura. Es por ello que fue necesaria la creación que mecanismos seguros de transferencia de información que garantizaran los requerimientos de seguridad e incluyeran la misma funcionalidad que el protocolo HTTP, universalmente utilizado. *HTTPS* es un protocolo de comunicaciones diseñado para transferir información cifrada entre equipos a través de Internet. *HTTPS* es un *HTTP* utilizando *Secure Socket Layer* y se recomienda utilizarlo cuando se transfiere información sensible de seguridad [7].

A medida que la Web se centra en la escalabilidad y el rendimiento en lugar de la seguridad, sufre mucho de diferentes ataques, incluyendo: la suplantación de servidores, mo-

dificación de mensajes, robo e inyección de *cookie's*. Contribuyendo a éstos problemas, la proliferación de redes inalámbricas facilita a cualquier atacante, ubicado en las zonas de proximidad, interceptar y modificar el tráfico de clientes Web. *HTTPS* puede poner fin a estos problemas al proporcionar *confidencialidad* e *integridad* de la información.

Si se utiliza *HTTPS* entre un cliente y un servidor Web, los recursos (paginas html, imágenes, archivos multimedia, etc...) solicitados a través de *HTTPS* no pueden ser modificadas y/o espiados durante su transferencia y muchos de los ataques, de los que se puede ser víctima en la Web, son mitigados. Incluso con estas mejoras fuertes de seguridad, *HTTPS* no ha reemplazado por completo a *HTTP*.

### 3.4.1. Comparación entre HTTP y HTTPS

Durante algunos años, *HTTPS* ha sido rechazado por muchos sitios Web, en primer lugar debido a la degradación del rendimiento. Si *HTTPS* sólo afecta al rendimiento, se adoptará por completo conforme los equipos de cómputo y las redes sean más rápidos en el futuro que en la actualidad. La principal respuesta al rechazo del protocolo se fundamenta en la arquitectura Web actual y *HTTPS* la rompe debido a que [10]:

- Por un lado, *HTTP* es un sistema de información distribuido en función de muchas entidades intermediarias entre un cliente y un servidor Web, en los cuales se almacenan las páginas en caché (*proxy's* y *gateways*, por ejemplo).
- Por otra parte, *HTTPS* proporciona seguridad en la comunicación de extremo a extremo y los recursos que viajan a través de *HTTPS* no pueden ser almacenados en caché en los dispositivos intermediarios, por lo que cualquier solicitud de algún recurso debe hacerse al servidor origen. Además, las páginas *HTTPS* no se pueden desplegar de forma incremental, ocasionando una visualización más lenta al usuario.

Sin embargo, existen estudios que muestran que se puede utilizar *HTTPS* como medio de comunicación para proveer a las aplicaciones de los servicios de seguridad que brinda el protocolo y obtener poca pérdida en el rendimiento.

## 3.5. Base de las aplicaciones Web

### 3.5.1. HTML

*HTML* (Hypertext Markup Language, por sus siglas en inglés) y *CSS* (Cascading Style Sheets) son dos tecnologías que forman parte importante en la construcción de páginas Web. *HTML* provee la estructura para las paginas, mientras que *CSS* el diseño visual y auditivo. *HTML* es el lenguaje que permite proporcionar la estructura de una pagina Web, con lo que ofrece medios para:

- Publicar documentos en línea.

- Obtener documentos en línea a través de enlaces de hipertexto.
- Diseñar formularios para realizar transacciones con servicios remotos.
- Incluir hojas de cálculo y otras aplicaciones en los documentos.

HTML ha sido desarrollado con el objetivo de que todo tipo de dispositivos puedan acceder a la información en la Web: PCs con pantallas gráficas de distintas características y resolución, teléfonos celulares, dispositivos de mano, dispositivos de entrada y salida de audio, equipos con distintos anchos de banda, etcétera.

Por otro lado, CSS es el lenguaje que permite describir la presentación de una página Web, lo que facilita la adaptabilidad de la interfaz a diferentes tipos de dispositivos (por ejemplo, pantallas o impresoras). CSS es independiente de HTML y puede utilizarse con cualquier aplicación basada en XML, a lo que se le conoce como la separación de la estructura (contenido) de la presentación.

### 3.5.2. Scripting y Ajax

Un *script* es el código de un programa que no necesita preprocesamiento (por ejemplo, la compilación), antes de su ejecución. En el contexto de un navegador Web, por lo general se refiere a las secuencias de comandos escritas en *JavaScript* ejecutadas por el navegador cuando una página es descargada o como respuesta a un evento generado por un usuario [11].

El *scripting* hace a las páginas web más dinámicas. Por ejemplo, sin necesidad de recargar una nueva versión de una página permite modificaciones en el contenido de la misma, agregar segmentos del contenido de la página o enviarlos a otra. El primero se ha denominado *DHTML* (HTML Dinámico) y el segundo AJAX (JavaScript Asíncrono con XML, por sus siglas en inglés) [11][20].

Por otro lado, el *scripting* ha permitido crear un puente entre el navegador y la plataforma que se ejecuta en el servidor, ya que es posible crear páginas Web que incorporan información del entorno del usuario: como la ubicación actual, información de la libreta de direcciones, entre otras [11]. Esta interactividad adicional hace que las páginas Web se comporten como una aplicación software tradicional. Estas páginas Web a menudo son llamadas *Aplicaciones Web* y están disponibles directamente en el navegador como una página convencional o pueden empaquetarse y distribuirse como *Widgets*.

## 3.6. Aplicaciones Ricas de Internet

Las primeras aplicaciones de Internet fueron basadas en el clásico modelo *cliente - servidor*, con varias páginas de contenido, mismas que se actualizaban con cada acción del usuario. Debido a que los usuarios comenzaron a demandar aplicaciones de web más sofisticados e



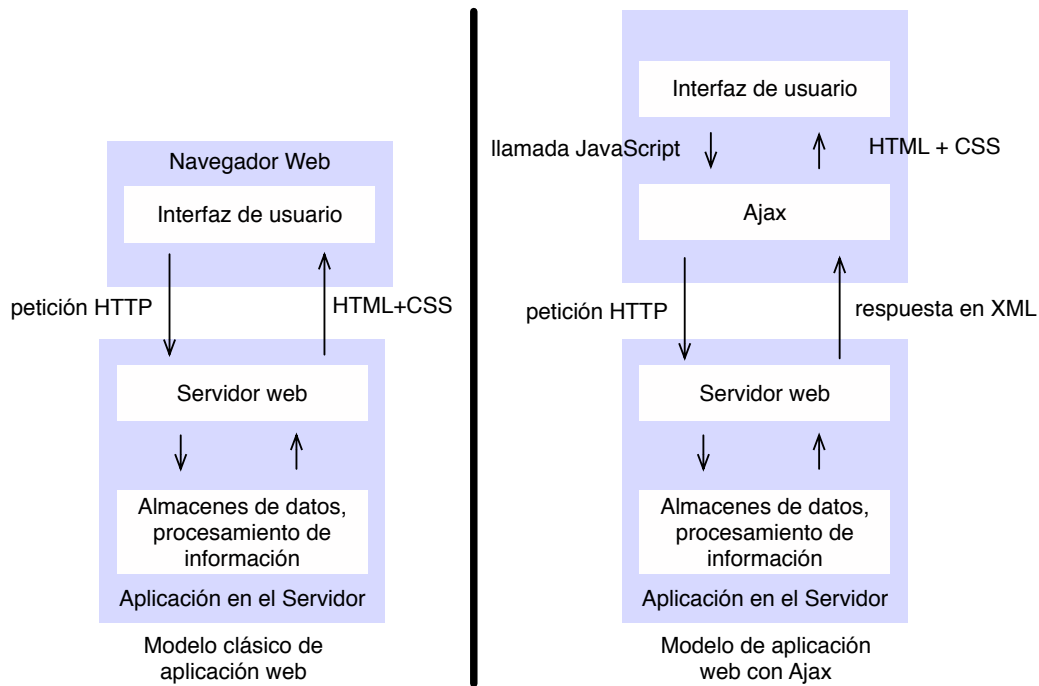


Figura 3.3: Modelo tradicional de Aplicación Web vs basado en Ajax

interactivas, el desarrollo de aplicaciones ha evolucionando para permitir una comunicación más eficiente entre clientes y servidores[18].

Las aplicaciones ricas de Internet (RIA, por sus siglas en inglés) utilizan técnicas asíncronas, como Ajax, para proveer comunicación y obtener información en tiempo real y visualizarla de forma dinámica. Esto dota a la interfaz web de la sensación de utilizar una aplicación de escritorio [18]. En la Figura 3.3 se muestra un análisis comparativo entre los modelos de aplicaciones Web tradicionales y los basadas en Ajax. Se observa que el modelo basado en Ajax consiste de una secuencia de eventos basados en peticiones asíncronas y la recuperación de la información contenida en la respuesta.

Los usuarios son sensibles a los tiempos de respuesta en percibidos. Los procesos no deben tomar lo que los usuarios podrían considerar demasiado tiempo en completarse. Una de las razones principales para el uso de RIA es crear al usuario la sensación de que está usando una aplicación de escritorio, en las que las tareas se ejecutan en el instante en que realiza una solicitud [20].

El objetivo de las aplicaciones ricas de Internet es utilizar la capacidad de procesamiento de los clientes y con ello disminuir la carga del servidor de aplicación. Las aplicaciones RIA tienen las siguientes características:

- Experiencia rica del usuario
- Mejores tiempos de respuesta
- Flexibilidad
- Fácil de distribuir y actualizar
- Fácil de administrar

# Capítulo 4

## Localización y seguimiento de dispositivos móviles

En este capítulo se estudian de manera general los aspectos más importantes de la localización de dispositivos móviles. En primera instancia se aborda el campo del cómputo móvil, analizando las características de los teléfonos celulares actuales y los motivos por los cuales se consideran dispositivos de cómputo móvil. Se realiza una reseña de las etapas de evolución de las redes de telefonía, en la que resaltan las características y servicios presentes en cada una de ellas. Posteriormente, se presentan dos vertientes de las tecnologías de localización móvil: sistemas satelitales y redes celulares. También se hace una breve mención de los Sistemas de Información Geográfica, cuyo uso en este ámbito radica en la visualización de la ubicación geográfica y trayectoria de los dispositivos móviles.

En la última sección se describen las arquitecturas y funcionamiento de tres implementaciones recientes en el ámbito de la localización móvil. La primera se enfoca en la localización de vehículos en tiempo real, la segunda en la localización y seguimiento de personas mayores y la tercera hace referencia a un sistema de infracciones de tránsito.

### 4.1. Cómputo Móvil

En la década de los noventa surge el campo del *Cómputo Móvil*, basado en los principios de los *Sistemas Distribuidos* y en la necesidad de integrar en un tipo de arquitectura a los clientes móviles, los cuales poseen diferentes características de funcionalidad y desempeño al de las computadoras de uso convencional [26].

En esta sección se abordan dos puntos del cómputo móvil: dispositivos y redes de comunicación móvil. Primero se presentan las características de los teléfonos celulares actuales y por que se consideran dispositivos de cómputo móvil. También se presenta una reseña de las redes de telefonía celular.

### 4.1.1. Dispositivos móviles

Los dispositivos móviles se definen como sistemas de cómputo que pueden desplazarse con facilidad y cuya funcionalidad se puede aprovechar mientras se encuentran en movimiento. Los ejemplos más comunes de estos dispositivos son las computadoras portátiles, asistentes digitales (PDA's, por sus siglas en inglés) y los teléfonos celulares. El incremento en la capacidad de cómputo de los procesadores, las mejoras en los sistemas operativos modernos para dispositivos móviles y la popularidad de la *banda ancha móvil*, hacen de los teléfonos celulares el mejor candidato como dispositivo de cómputo móvil; capaces de realizar, incluso, sofisticadas aplicaciones científicas. Actualmente los teléfonos celulares pueden incluir: navegador Web, cámara y/o videocámara, reproductores de música y video, módem, receptor GPS, soporte para video juegos, radio y televisión [43].

Al mismo tiempo que las infraestructuras de redes de comunicación móvil evolucionaron, se incrementó considerablemente el número de usuarios de la banda ancha móvil, gracias a una mejor usabilidad, precios cada vez más bajos y a la disponibilidad de acceso a Internet prácticamente en cualquier lugar [43].

Hoy en día, los procesadores de los teléfonos celulares (teléfonos inteligentes, como se les ha catalogado) son tan completos que incluso las computadoras portátiles con bajo poder de cómputo pueden utilizarlos. Además, los procesadores para dispositivos móviles empiezan a entrar en la era multinúcleo. Anteriormente, los teléfonos inteligentes contaban con capacidad de almacenamiento de 32 a 512 Mb, actualmente ofrecen capacidades de almacenamiento de datos de 4 a 32 Gb [43]. Modernas plataformas para dispositivos móviles, como iPhone [38] y Android [1], incluyen una amplia gama de bibliotecas útiles e *Interfaces de Programación de Aplicaciones* (API, *Application Programming Interface*, por sus siglas en inglés) para desarrolladores. Las bibliotecas permiten acceder a los sensores internos, como acelerómetros o GPS. Estas *bibliotecas* y *API's* permiten a los desarrolladores implementar nuevas aplicaciones y controlar el teléfono con mayor libertad y flexibilidad [43].

El número de suscriptores de los servicios de telefonía móvil alcanzó los 4 mil millones alrededor del mundo a finales de 2009 y se espera que la cifra crezca 1 millón cada año. Los teléfonos celulares son un ejemplo de *Cómputo Móvil*, pueden realizar tareas de cómputo distribuido y ejecutar aplicaciones basadas en su ubicación [43].

### 4.1.2. Tecnología de Comunicación Móvil

La primera generación (1G) de los sistemas celulares aparecieron en la década de 1980 y estuvieron basados en tecnología analógica. Las redes de primera generación se caracterizaron por su baja capacidad de transferencia, la carencia de seguridad y la incapacidad para transmitir voz [34].

Los sistemas de segunda generación (2G) iniciaron su operación a principios de la década de 1990 ofreciendo a los operadores de red incremento en la capacidad y disminución de

1G	2G	3G	4G
Movilidad básica	Movilidad avanzada ( <i>roaming</i> )	Sin limites	Movilidad basada en IP
Servicios Básicos	Mayor número de servicios ( <i>intercambio de datos</i> )	Concepto de servicio y modelo	Altas tasas de transferencia
Incompatibilidad	En camino a una solución global	Solución global	Telecomunicaciones perfectas y convergencia.

Tabla 4.1: Comparativa de las redes de comunicación móvil

costos, mientras que a los usuarios les proporcionaron servicios de mensajes cortos y transferencia de voz a baja velocidad [35]. 2G marcó la llegada de las técnicas de modulación digital y con ello el incremento de la capacidad, mejor calidad en la transferencia de voz, mejores funciones de seguridad y terminales más eficientes [34]. En las redes de segunda generación se alcanzaron tasas de transferencia de 9 a 14 Kbps.

En 2001 los sistemas de tercera generación (3G) entran en funcionamiento con el servicio FOMA (*Freedom of Mobile Multimedia Access*) en Japón, con diversos campos de experimentación en Europa y Corea del Sur. Originalmente se pensó que 3G sería una única especificación a nivel mundial, pero en realidad existen dos tecnologías principales: WCDMA y cdma2000 [35]. Con la tecnología de tercera generación se alcanzan tasas de transferencias de 384 Kbps con EDGE (*Enhanced Data rates for GSM of Evolution*) y 2 Mbps con UMTS (*Universal Mobile Telecommunications System*). Además, HSDPA (*High-Speed Downlink Packet Access*) ofrece una tasa de 7.2 Mbps.

Posteriormente surgieron estándares denominados de cuarta generación (4G). El estándar para WiMax fue aprobado en el año 2005 para promover el cumplimiento y la interoperabilidad de la norma IEEE 802.16. La arquitectura escalable, el alto rendimiento y el bajo costo de desarrollo hacen a WiMax una solución líder para servicios inalámbricos de banda ancha. La promesa de la redes de cuarta generación es brindar tasas de transferencia de 100 Mbps a usuarios en movimiento y hasta 1 Gbps usuarios en reposo [35].

En la Figura 4.1 y la Tabla 4.1 se puede observar la evolución, desarrollo y convergencia de los sistemas de comunicación móvil, donde la mayor promesa es la movilidad.

### 4.1.3. Dimensiones del cómputo móvil

Las dimensiones de la movilidad son las herramientas que permiten clasificar un problema de creación de aplicaciones de software y sistemas de cómputo móvil. Además, se debe

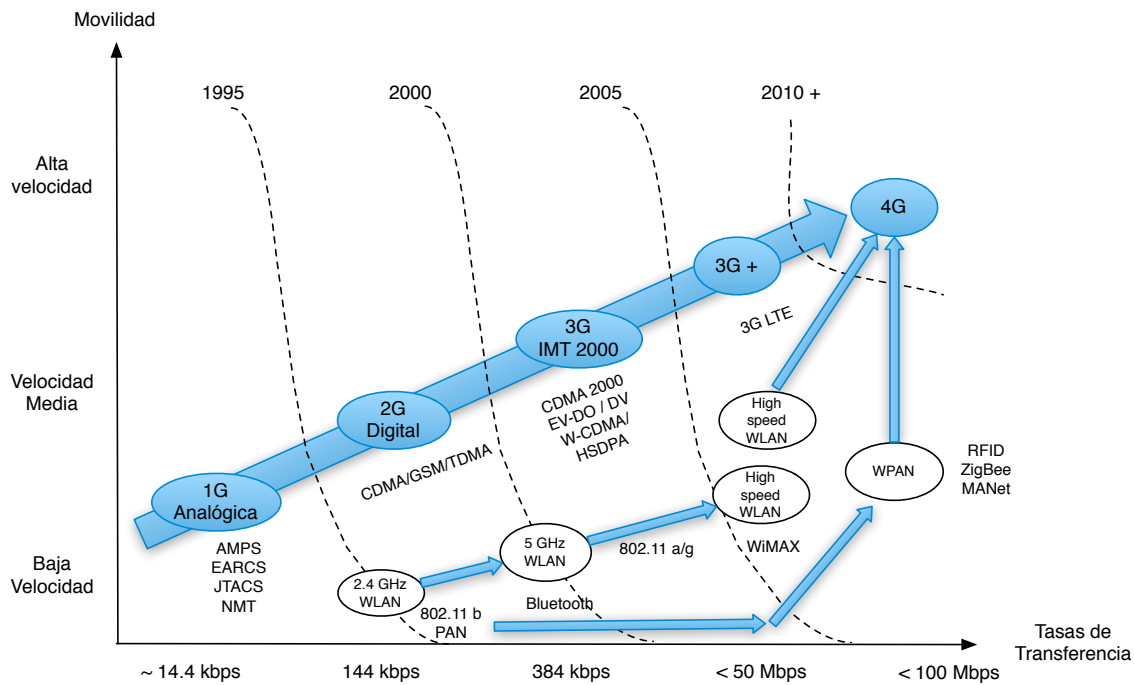


Figura 4.1: Convergencia de los sistemas de comunicación móvil

considerar que algunas dimensiones de la movilidad se refieren a limitaciones, sin embargo, éstas no se deben considerar cuando se tratan aplicaciones no móviles. Las dimensiones de la movilidad (ver Figura 4.2) son:

1. Conocimiento de localización,
2. Conexión de red con calidad de servicio (QoS),
3. Capacidad limitada de los dispositivos (en particular almacenamiento y CPU),
4. Fuente de alimentación limitada,
5. Amplia variedad de interfaces de usuario,
6. Proliferación de plataformas y
7. Comportamiento activo

En esta sección se abordan las dimensiones: *conocimiento de la localización, conexión de red con calidad de servicio (QoS) y fuente de alimentación limitada* [8].

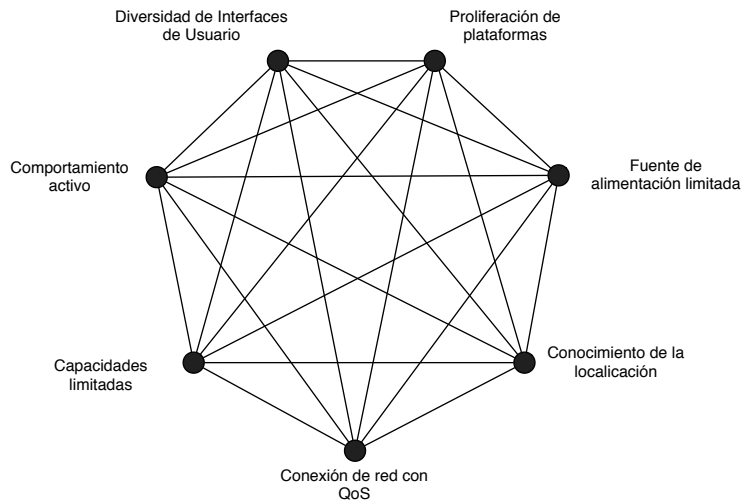


Figura 4.2: Dimensiones de la movilidad

#### 4.1.3.1. Conocimiento de la localización

El cambio en la localización de un dispositivo móvil representa para los creadores de software un gran número de dificultades y al mismo tiempo la oportunidad de utilizar el conocimiento de la ubicación para mejorar y aumentar la funcionalidad de las aplicaciones. Estos desafíos y oportunidades se clasifican en dos categorías generales: *localización* y *sensibilidad de localización*.

La **Localización** es la capacidad de la arquitectura de una aplicación móvil, basándose en la información de localización predefinida, de adaptarse para permitir la selección de diferentes lógicas de negocios, niveles de flujo de trabajo e interfaces de usuarios. Esta característica se presenta en aplicaciones móviles como en las no móviles.

La **Sensibilidad de la ubicación** es la capacidad de los dispositivos y aplicaciones móviles de, primero adquirir información sobre la localización y luego utilizarla para obtener características y funcionalidad particulares. Esta característica es propia de las aplicaciones móviles debido a la constante retroalimentación de los dispositivos y aplicaciones con su contexto y ubicación.

#### 4.1.3.2. Conexión de red con calidad de servicio (QoS)

Ya sea por medio de una red cableada o inalámbrica, la movilidad significa pérdida de una conexión de red fiable. Desplazarse de un lugar a otro crea barreras físicas que garantizan algún tiempo de desconexión de la red. En el caso de conexión por medio de una red inalámbrica factores como el tiempo, fenómenos solares y variaciones en el clima pueden afectar negativamente la calidad del servicio (QoS).

Considerar QoS en el diseño de aplicaciones móviles es crítico, ya que las aplicaciones móviles deben seguir funcionando a pesar de sufrir conexiones y desconexiones intermitentes y frecuentes de la red.

#### 4.1.3.3. Fuente de alimentación limitada

El desarrollo de las baterías mejora día con día, de la misma forma en que la energía eléctrica se encuentra disponible en casi todo el mundo. Sin embargo, la mayor parte del tiempo los usuarios se encuentran en movimiento y los dispositivos móviles consumen cada vez más energía debido a los componentes internos que contienen. El uso de una batería como fuente de alimentación crea el problema de contar con una *fuentes de alimentación limitada*. Ésta restricción se debe equilibrar con la capacidad de procesamiento, almacenamiento y las restricciones de peso y tamaño, ya que la batería suele ser la fuente más importante de peso en un dispositivo móvil.

## 4.2. Tecnologías de localización móvil

En esta sección se presentan las dos vertientes utilizadas en el ámbito de la localización móvil, abordando las características de los métodos y tecnologías utilizadas en el campo.

### 4.2.1. Sistemas de Posicionamiento Satelital

El *Sistema de Posicionamiento Global* (GPS, por sus siglas en inglés) es el sistema de posicionamiento satelital más popular y una tecnología madura para localización precisa en ambientes al aire libre [41]. El GPS consta de los siguientes tres segmentos:

- Espacio: GPS consta de una constelación de 24 satélites, distribuidos en 6 planos diferentes, orbitando a 20,200 km de la Tierra y 55° de inclinación [36][37][12].
- Control: Conjunto de centros de control y monitoreo distribuidos al rededor del mundo, cuyo objetivo es mantener a los satélites en sus orbitas correctas y ajustar sus relojes.
- Usuario: Consiste de un *receptor* GPS, que recibe las señales de los satélites GPS y utiliza la información transmitida para calcular la posición tridimensional y tiempo del usuario.

El GPS tiene la capacidad de proveer información de posición y tiempo a buques, aviones, vehículos terrestres y vehículos espaciales. Para obtener el servicio de localización, el receptor GPS debe sincronizarse con 3 satélites y un cuarto para obtener el servicio de tiempo. El potencial del GPS se ha reconocido desde la concepción del sistema [12].

En los últimos años, el GPS se ha utilizado en la agricultura, aplicaciones militares, sistemas de localización, técnicas de determinación y ubicación de órbitas, por mencionar



algunas. El GPS fue diseñado originalmente para usuarios ubicados en la Tierra o cerca de ella, sin embargo, se ha utilizado en misiones espaciales [44].

Un inconveniente para la seguimiento de personas usando GPS, además de que su uso en interiores no es viable, es el largo tiempo requerido para obtener la primera ubicación (*TTFE*, por sus siglas en inglés). En los modos de arranque en *caliente* y *frío*, el TTFE generalmente se encuentra entre 40 *s* y 2 *min*. En aplicaciones para seguimiento de personas, en las cuales el tamaño de la batería y tiempo de vida son consideraciones importantes, operar el GPS en modo caliente es una opción que generará un alto consumo de energía.

### 4.2.2. Sistemas de Posicionamiento Celular

Los sistemas de posicionamiento celular utilizados en redes de segunda generación (GSM) fueron: *Identificador de Célula* (Cell-Id, por sus siglas en inglés), Mejor Diferencia de Tiempo Observada (E-OTD, por sus siglas en inglés), Diferencia de Tiempo de Llegada del Enlace de Subida (U-TDoA, por sus siglas en inglés) y GPS Asistido (AGPS, por sus siglas en inglés).

Algunos ejemplos de sistemas de posicionamiento celular de tercera generación son: métodos *Basados en Células*, Diferencia Observada entre el Tiempo de Llegada y el Periodo de Inactividad del Enlace de Descarga (OTDoA-IPDL, por sus siglas en inglés) y AGPS.

AGPS es una solución híbrida que utiliza la información provista por los satélites GPS y la red de telefonía celular [Agrawal & Zeng, 2006]. El *GPS Asistido* se auxilia de fuentes externas, como servidores de asistencia o referencias de red, para ayudar al receptor GPS a disminuir el tiempo de respuesta y el margen de error [36][27]. Esto permite que los dispositivos móviles se comporten como receptores GPS permitiendo su ubicación rápida y con mayor precisión [3].

Actualmente, el mejor rendimiento de localización utilizando dispositivos de cómputo móvil es provisto por la técnica de localización *AGPS*. AGPS tiene mejor rendimiento que GPS, debido al menor tiempo en la adquisición de la señal de los satélites y mejor disponibilidad de servicio en áreas urbanas e interiores, además reduce significativamente la complejidad en los receptores [9][30]. En la Tabla 4.2 se muestra un análisis comparativo realizado entre los métodos mencionados en la sección [2].

## 4.3. Sistemas de Información Geográfica

En los últimos años se ha observado el creciente desarrollo de los sistemas de información que incorporan información geográfica. A estos sistemas se les ha clasificado como *Sistemas de Información Geográfica* (GIS, *Geographics Information Systems*, por sus siglas en inglés), a los cuales se les define como [17]:

	Precisión			Rendimiento
	Rural	SubUrbano	Urbano	
Cell-Id	> 10 km	2 – 10 km	20 – 1000 m	Bueno
E-OTD & OTDoA	50 – 150 m	50 – 250 m	50 – 300 m	Promedio
U-TDoA	50 – 120 m	40 – 50 m	40 – 50 m	Promedio
AGPS	10 – 40 m	20 – 100 m	30 – 150 m	Bueno

Tabla 4.2: Sistemas de localización celular

- Sistemas diseñados para almacenar y manipular información relacionada con ubicaciones de la superficie terrestre.
- Conjunto de herramientas para recolectar, almacenar, obtener, transformar y desplegar información espacial de la Tierra para propósitos específicos.

Un Sistema de Información Geográfica integra hardware, software y datos; para capturar, administrar, analizar y visualizar información geográficamente referenciada. Los GIS son utilizados en ingeniería civil, geología, inteligencia militar, control de tráfico vial, entre otros [4].

Un porcentaje considerable de uso de los GIS radica en el monitoreo vehicular, permitiendo implementar estrategias eficaces para evitar que los vehículos en tránsito tengan retrasos en los tiempos de llegada a sus destinos. Los sistemas de transportes utilizan esta capacidad para realizar seguimiento de trenes, autobuses y otros servicios [37].

## 4.4. Trabajo relacionado

En la presente sección se exponen algunas implementaciones realizadas en el ámbito de la localización móvil y cuya funcionalidad se enfoca en la localización y seguimiento de vehículos y personas.

### 4.4.1. Sistema de seguimiento vehicular en Tiempo Real

El diseño e implementación del sistema incluye la adquisición y transmisión de la información de la ubicación del vehículo, junto con el encendido y estado de las puertas, a la estación de monitoreo o servidor. El sistema también provee una interfaz Web para desplegar en un mapa la información vehicular transmitida (ver Figura 4.3). El sistema tiene dos componentes principales [16]:

- Dispositivo de *hardware* llamado *In Vehicle-Unit* (IVU), ver figura 4.4:
  - Receptor GPS: utiliza un receptor GPS para capturar la ubicación y velocidad actual del vehículo, mismas que no se encuentran en texto plano. Por lo cual, para poder transmitir la ubicación de un vehículo es necesario aplicar un preprocesamiento a dicha información.

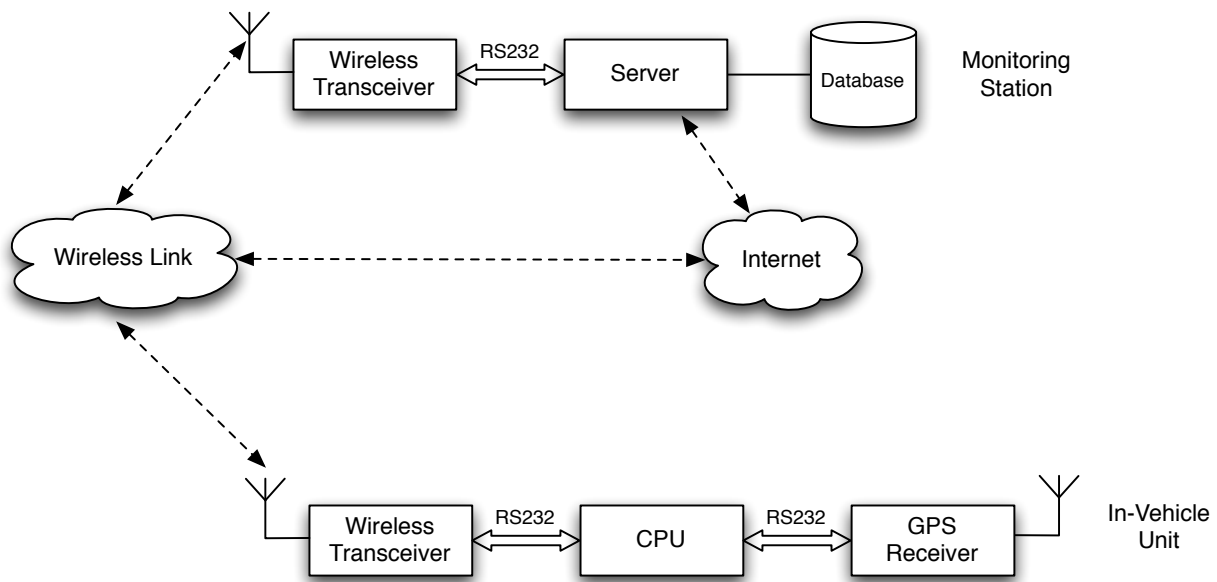


Figura 4.3: Arquitectura de alto nivel del sistema

- Unidad Central de Procesamiento (CPU): Es el encargado de procesar la información provista por el receptor GPS para extraer la información requerida de localización y velocidad. EL CPU también es el responsable de monitorear el estado de las puertas y controlar el estado de *encendido* del vehículo. El CPU es el encargado de controlar el intercambio de información con el *Servidor de Seguimiento*.
- Transmisor: Éste componente es el encargado de transmitir al *Servidor de Seguimiento* la información obtenida por el GPS y procesada por el CPU. Para transmitir la información, debido a su amplia cobertura, se utiliza la red GSM. Por lo anterior es necesario un modem GSM.
- Servidor de Seguimiento (ver Figura 4.5):
  - Módem GSM: Es el componente encargado de intercambiar mensajes SMS entre el servidor y el IVU. El módem pasa la información recibida al servidor a través del puerto serial. El servidor puede enviar comandos al IVU a través del modem y para comunicarse con él utiliza comandos *AT*.
  - Servidor de base de datos: En MySQL se implementó la base de datos que almacena la información relevante de los vehículos y de los usuarios del sistema y es necesaria para desplegar la información de posicionamiento a los usuarios finales.
  - Servidor Web: Utilizaron el servidor *Apache* para montar una pequeña interfaz

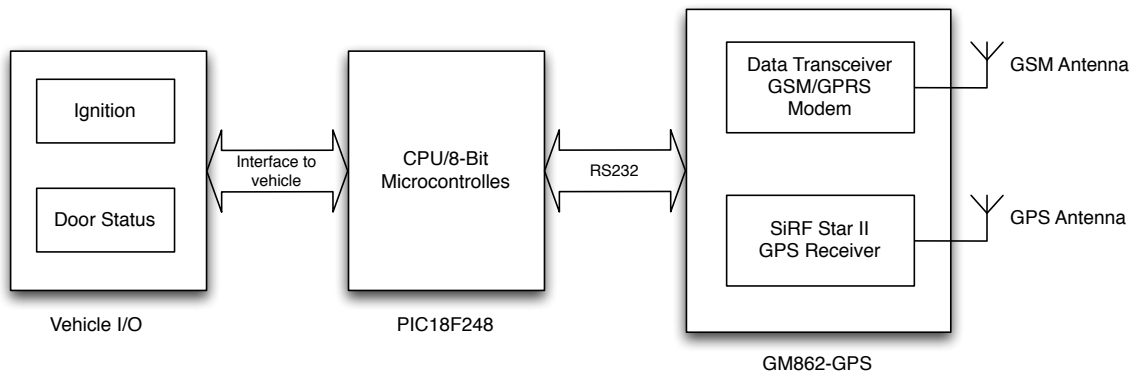


Figura 4.4: Arquitectura de la Unidad en el Vehículo

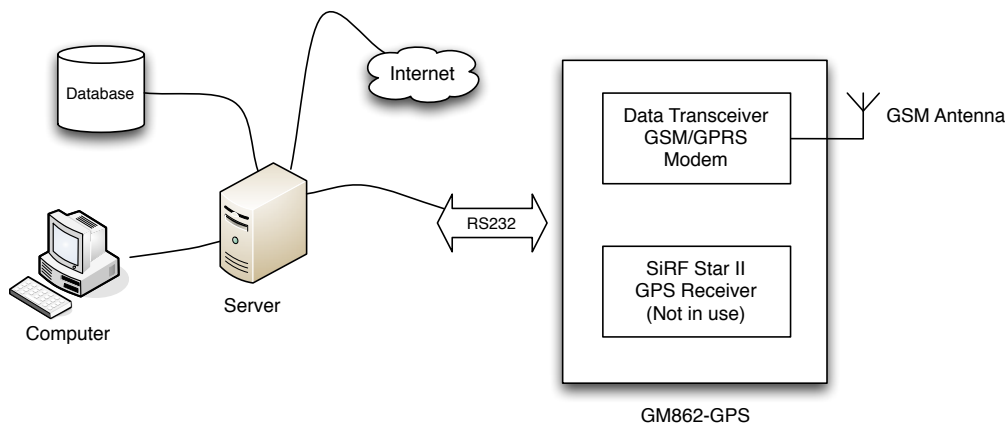


Figura 4.5: Arquitectura del Servidor de Seguimiento

Web que brindara acceso a los usuarios del sistema y a su vez realizaran el seguimiento de sus vehículos.

- Servicio de mapas: Emplearon el servicio de mapas de *Google* (Google Maps) para visualizar la ubicación geográfica de los vehículos.

#### 4.4.2. Sistema de seguimiento para personas mayores basado en AGPS

Este prototipo fue desarrollado con la finalidad de mejorar la calidad de vida de las *Personas Mayores* en la ciudad de Hong Kong, debido a que éstas en su mayoría viven de manera independiente. El proyecto nace de la necesidad de proveer la ubicación de las *Personas Mayores* en seguimiento a los familiares y servicios de emergencia médica. A dichas

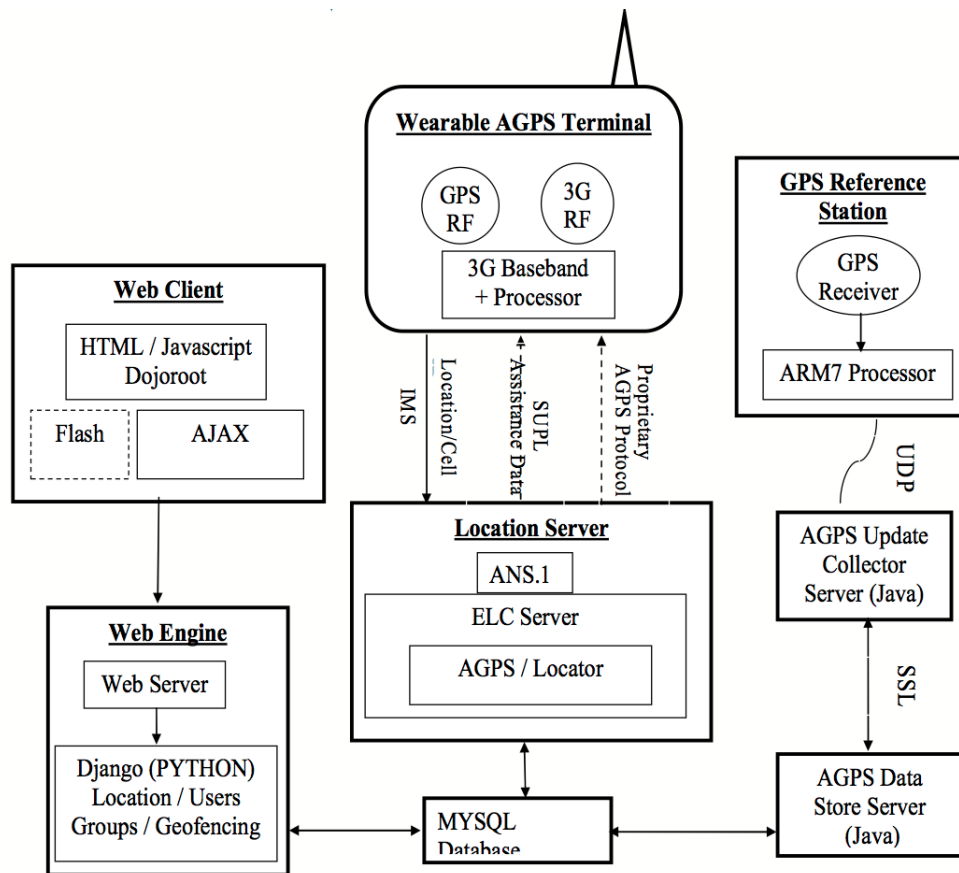


Figura 4.6: Arquitectura de alto nivel del sistema

personas se les proporcionó un dispositivo que emite una alerta cuando tienen problemas de salud. El sistema consta de los siguientes componentes [41]:

- *Terminal AGPS*: la terminal AGPS tiene un chip para comunicación por 3G y un receptor AGPS.
- *Servidor Móvil de Localización*: es un servidor *SUPL*, propietario de Broadcom, que provee un único punto de contacto a todas las terminales AGPS.
- *Estación de referencia GPS*: es un dispositivo receptor GPS y un procesador ARM7. Recibe su ubicación de los satélites GPS y la envía al servidor de localización y éste último a las terminales AGPS, con la finalidad de reducir el margen de error.
- *Servidor de base de datos*: la base de datos se montó sobre MySQL utilizando el soporte de información geográfica que ofrece.
- *Servidor Web*: tiene la funcionalidad de brindar acceso a los clientes Web, realizar control de usuarios y despliegue de información geográfica. El servidor fue implementado utilizando el *framework Django*.

- *Cliente Web*: provee a los usuarios finales acceso a la aplicación y es por medio de éste que se realiza el seguimiento de las *Personas Mayores*. Para su implementación se utilizó: AJAX, DOM, *Dojo Toolkit* y API de Google Maps.

Este prototipo provee una precisión en la localización de las personas en un rango de 10 a 20 *m* cuando las personas se encuentran en movimiento y 138 m en reposo. Además, permite la asignación de una *geocerca* a las terminales AGPS, con la finalidad de emitir una notificación cuando la terminal AGPS sale de ella. La arquitectura del sistema se ilustra en la Figura 4.6.

#### 4.4.3. Desarrollo de un Sistema de Tecnologías de Cómputo, Comunicaciones y Geolocalización

Este sistema es una aplicación cliente servidor, ver Figura 4.7, diseñada para auxiliar a un agente de tránsito en el levantamiento de una infracción [19]. Consta de los siguientes componentes principales:

1. Cliente móvil: es una aplicación desarrollada para teléfonos celulares con plataforma *Android 1.6*, con la cual es posible realizar el levantamiento de una infracción de tránsito y enviarla al servidor de aplicación por medio de una red inalámbrica 802.11 o a través de la red GPRS.
2. Servidor Web: es una aplicación diseñada para recibir las infracciones de tránsito tomadas con los clientes móviles y con base en la información recibida el servidor verifica si el vehículo involucrado en el incidente tiene asociado algún otro quebrantamiento (otra multa, denuncia de robo, etc). Por otro lado, ofrece un portal por medio del cual es posible verificar la ubicación geográfica en la que se realizó una multa auxiliándose del servicio de mapas de *Google*.

Para el manejo de la información geográfica asociada a la multa, el sistema implementó una base de datos sobre el gestor *PostgreSQL 8.4*.

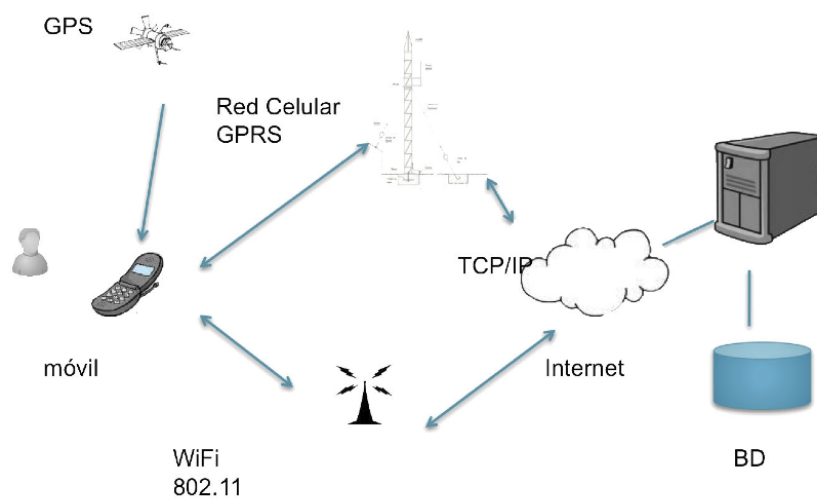


Figura 4.7: Arquitectura del sistema, basada en el modelo cliente servidor.





# Capítulo 5

## Análisis y diseño de la propuesta

En el presente capítulo se presenta el modelo de comportamiento y la arquitectura de la propuesta para localización y seguimiento de dispositivos móviles. Para lo cual, en primera instancia se presenta la funcionalidad requerida vista desde distintos puntos, es decir, desde la perspectiva de los diferentes tipos de usuarios. Posteriormente se expone la arquitectura general de comunicación, misma que se basa en el modelo *cliente servidor* para satisfacer el requerimiento de operabilidad sobre la Internet. Posteriormente se presentan las arquitecturas diseñadas para los módulos en que se ha dividido el sistema: aplicación web, cliente web y aplicación móvil. A lo largo del capítulo se utilizarán los términos *portador del dispositivo móvil*, *dispositivo móvil* y *móvil* de forma indistinta.

### 5.1. Modelo de comportamiento

El modelo de comportamiento de un sistema es el que contiene la definición de la funcionalidad requerida para un sistema. Las formas más comunes de definir el comportamiento esperado de un sistema son el lenguaje natural estructurado y modelado de casos de usos. Si se utiliza lenguaje natural estructurado los requerimientos se plasman en formas en donde se describe el requerimiento, como se presenta en la sección A.2. También se puede adoptar la estrategia de documentar con base en casos de uso del Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), la cual es muy recomendable debido a que permite modelar el comportamiento externo de un sistema, es decir, la funcionalidad que un sistema debería de proporcionarle a un determinado usuario.

La problemática general detectada en el dominio de las aplicaciones de localización y seguimiento, radica principalmente en la necesidad de conocer la ubicación geográfica de un determinado ente y la trayectoria descrita por sus recorridos. Además, existe la necesidad de poder determinar la ubicación de un elemento en un intervalo de tiempo determinado. Para solucionar ésta problemática es necesario proveer la siguiente funcionalidad:

- Mecanismos para la administración (altas, bajas y modificaciones) de las entidades presentes en el sistema.

- Proveer medios para la fácil asignación de regiones de recorridos (geocercas) a los dispositivos móviles.
- Diseñar herramientas para la actualización del tiempo en que los móviles reportan su ubicación geográfica y la distancia mínima que debe desplazarse un móvil para que no se considere desplazamiento nulo.
- Diseñar una interfaz que permita localizar y seguir la trayectoria de un móvil con base en un mapa georeferenciado.
- Proveer notificaciones que indiquen a los usuarios del sistema cuando un determinado móvil ha abandonado la región de recorrido que le fue asignada o tiene desplazamiento nulo.

La funcionalidad de la propuesta de solución se ha conceptualizado en dos subsistemas con los que interactuarán los usuarios: cliente móvil y aplicación web. Los cuales a su vez se sometieron a un análisis para su descomposición modular, el cual se muestra en la Figura 5.1. La Aplicación Web es el ente que controla toda la operación del sistema, ya que los clientes móviles y clientes web dependen de ella para su funcionamiento. Esto se debe a que el grueso del procesamiento se realiza en el servidor y los clientes.

Los usuarios del sistema que interactúan con el cliente web son el: *supervisor* y *administrador*. El supervisor es el perfil de usuario al que se le proporciona la funcionalidad para localizar a los dispositivos móviles en el sistema. El administrador, es el perfil de usuario que está facultado para realizar las altas, bajas y modificaciones de las entidades que maneja el sistema. En la Figura 5.2 se muestra la relación de asociación entre los usuarios y el caso de uso de control de acceso al sistema.

En las Figuras 5.3 y 5.4 se muestran los casos de uso de los módulos de localización y seguimiento, respectivamente. En estos casos de uso, se abstrae la funcionalidad requerida para que el usuario *supervisor* pueda visualizar, en mapa georeferenciado, la ubicación y desplazamiento de un dispositivo móvil. Además, en caso de existir alertas de región abandonada y desplazamiento nulo, el sistema notificará de forma gráfica al usuario.

Además de los casos de uso en los que recae la funcionalidad del modelo de aplicación (módulos de localización, seguimiento y políticas) todo el sistema debe proveer de los medios que le permitan la gestión de las entidades del sistema, con lo cual se dota al sistema de la flexibilidad de agregar nuevas entidades y modificar o eliminar las ya existentes. Los casos de uso de gestión del sistema los utiliza el usuario con perfil de administrador, mismos que se muestran en la Figura 5.5.

Por otro lado, existe el perfil de *usuario en movimiento*. Éste usuario es el que interactúa con el sistema a través de un dispositivo móvil y su funcionalidad proporciona una interfaz que le permite visualizar, en un mapa georeferenciado, su ubicación geográfica al mismo tiempo que la envía al servidor de aplicación.

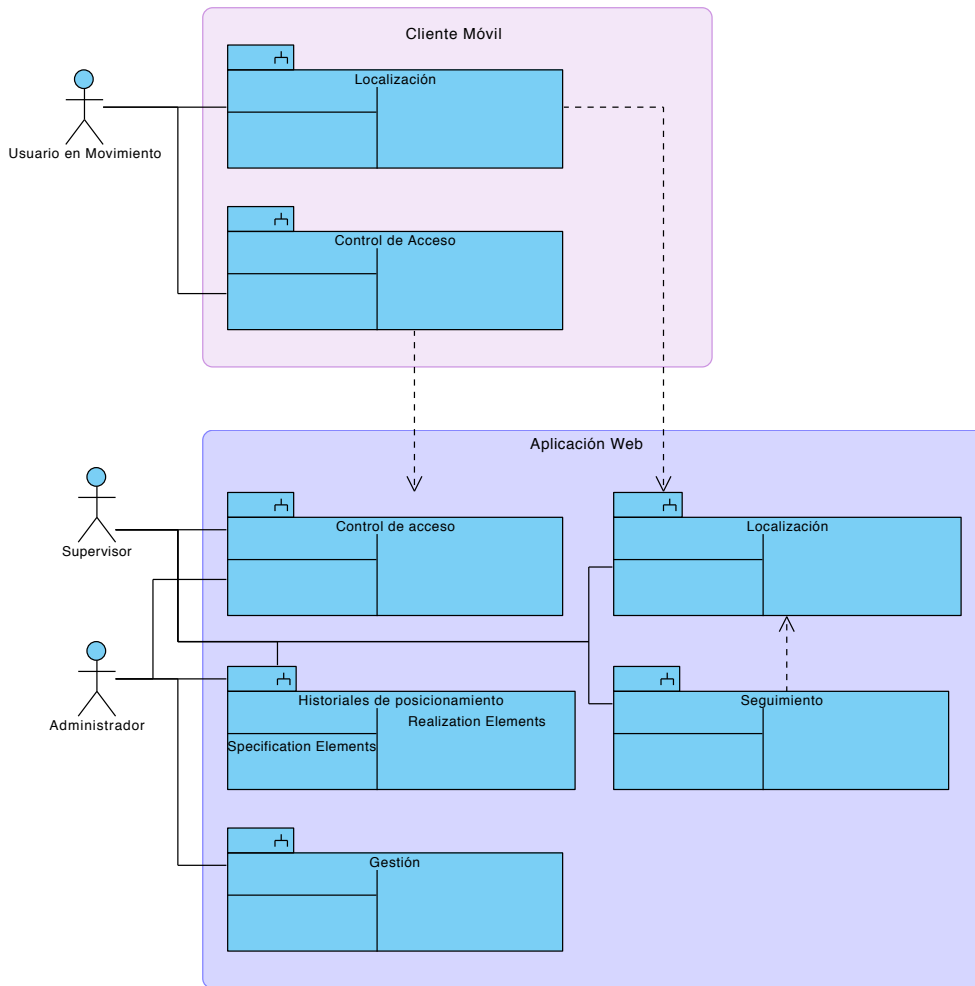


Figura 5.1: Diagrama general de los módulos de la propuesta.

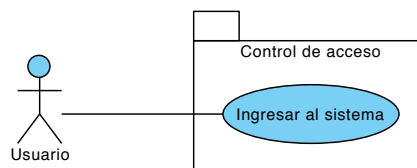


Figura 5.2: Caso de uso del módulo de control de acceso.

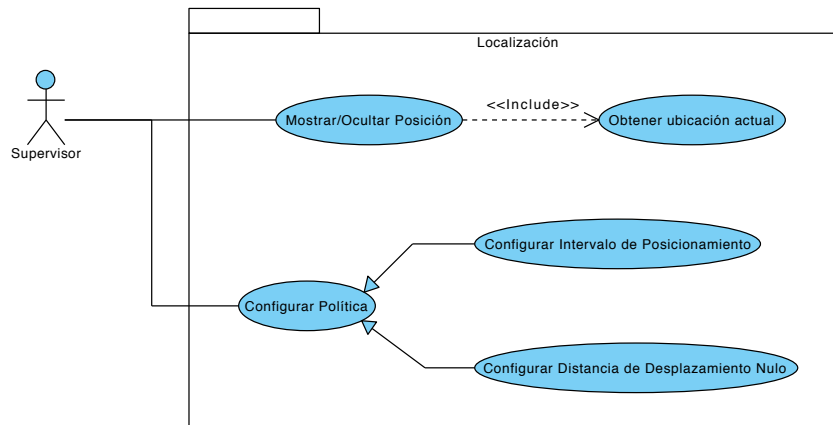


Figura 5.3: Casos de uso del módulo de localización.

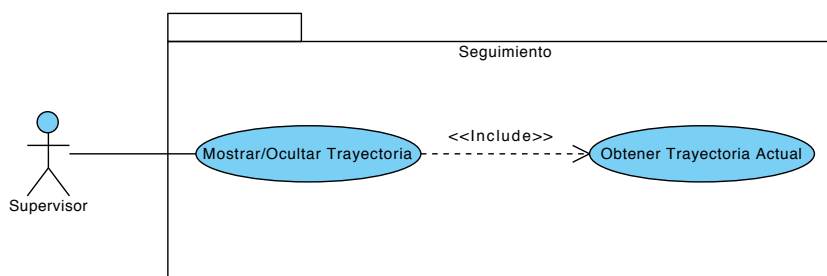


Figura 5.4: Casos de uso del módulo de seguimiento.

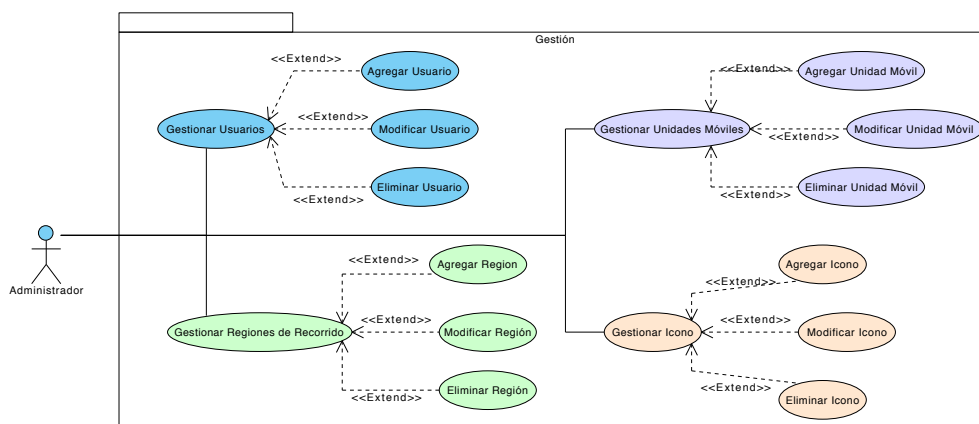


Figura 5.5: Casos de uso del módulo de gestión.

## 5.2. Arquitectura general de comunicación

La arquitectura es la organización fundamental de un sistema incorporando sus componentes, las relaciones que existen entre ellos y con el medio en el que se encuentran y los principios que guían su diseño y evolución [15]. Una arquitectura es el conjunto de decisiones importantes sobre la organización de un sistema de software; la selección de los elementos estructurales, su comportamiento y las interfaces mediante las cuales se comunican [5]. La arquitectura de un sistema también define las interacciones entre sus elementos estructurales; y son éstas interacciones las que proporcionan el comportamiento deseado del sistema [15].

El uso de una apropiada arquitectura de software puede influir enormemente en el éxito de un proyecto, ya que la arquitectura afecta a su descripción, utilización y aspectos de evaluación de software. También proporciona una descripción de los sistemas, la cual ayuda a comprender el sistema con facilidad, proporcionando un panorama más amplio del mismo [22]. La arquitectura define la estructura y el comportamiento, pero no tiene que ver con la descripción de la estructura y el comportamiento. En general, la arquitectura no tiene que ver con los detalles de grano fino de estos elementos [15].

La propuesta del sistema de localización y seguimiento de dispositivos móviles se basa en una arquitectura de comunicación basada en el modelo cliente servidor, en la que se incluyen de 3 componentes generales (ver Figura 5.6): Servidor de Aplicación, Cliente Web y Cliente Móvil. El *Servidor de Aplicación* tiene la finalidad de contener una aplicación web que atienda a los clientes móviles y clientes web, donde cada uno de ellos demanda una funcionalidad específica. Los *clientes móviles* requieren que la aplicación Web atienda las peticiones de envío de su ubicación geográfica y verifique si la ubicación es objeto de la emisión de alguna alerta de región abandonada o desplazamiento nulo.

Los *clientes web* demandan a la aplicación web la funcionalidad necesaria para realizar la tarea de localización y seguimiento de los *clientes móviles* presentes en el sistema. Aunado a lo anterior, ambos clientes requieren que la aplicación web provea un mecanismo de autenticación que permita verificar la identidad de los usuarios que ingresan al sistema.

Es importante mencionar que la comunicación y el intercambio de información entre el servidor y los clientes móviles y web se realiza bajo el protocolo HTTP/TLS, el cual utiliza una conexión TCP/IP en el puerto 443.

## 5.3. Arquitectura de subsistemas

En esta sección se presenta el diseño arquitectónico del sistema de localización y seguimiento, para lo cual el sistema en su totalidad se ha dividido en los siguientes 3 subsistemas: aplicación web, cliente móvil y cliente web. En la Figura 5.7 se muestra un diagrama de los componentes del sistema, las tecnologías de desarrollo de software y los mecanismos de comunicación utilizados entre ellos. Dicha figura tiene la finalidad de ilustrar las tecnologías

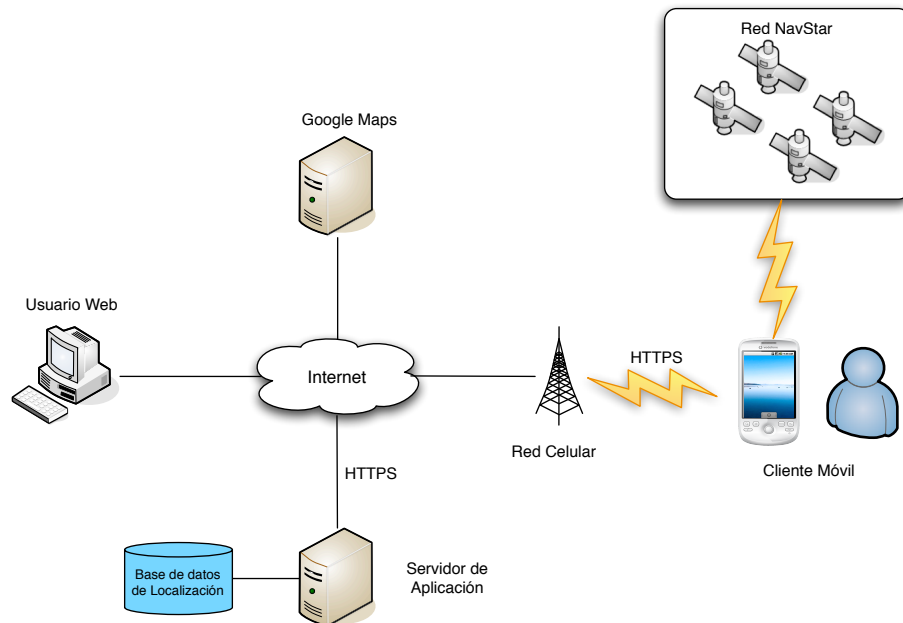


Figura 5.6: Arquitectura de comunicación del sistema de localización y seguimiento de dispositivos móviles

utilizadas en el proceso de implementación.

La arquitectura de la aplicación web, desde el punto de vista general corresponde con el patrón de diseño *Modelo-Vista-Controlador* (MVC). Además, la capa correspondiente al modelo se ha diseñado con base en una arquitectura en 3 capas, es decir, para el diseño de la aplicación web se adoptó un enfoque con arquitectura heterogénea (ver la sección 2.4).

El diseño de la arquitectura de los subsistemas correspondientes al cliente web y móvil se realizó tomando como base un modelo en 3 capas. La capa superior abstrae la funcionalidad requerida por la interfaz gráfica, en la capa intermedia se implementa la lógica de negocios presente en el contexto del subsistema y, finalmente, la tercera capa se enfoca en el acceso a datos. Es importante mencionar que en el cliente web la implementación de las capas propuestas se realiza con *JavaScript* mediante la implementación de un fragmento de software desarrollado con el *framework prototype*.

### 5.3.1. Aplicación Web

Como se mencionó al inicio de la sección anterior, la aplicación web atiende las peticiones de los clientes siguiendo el patrón de diseño MVC (ver Figura 5.8). Este patrón de diseño dota a la aplicación web de una arquitectura en capas que le permite separar las entradas, el procesamiento y las salidas en los siguientes 3 componentes: modelo, vista y el controlador.

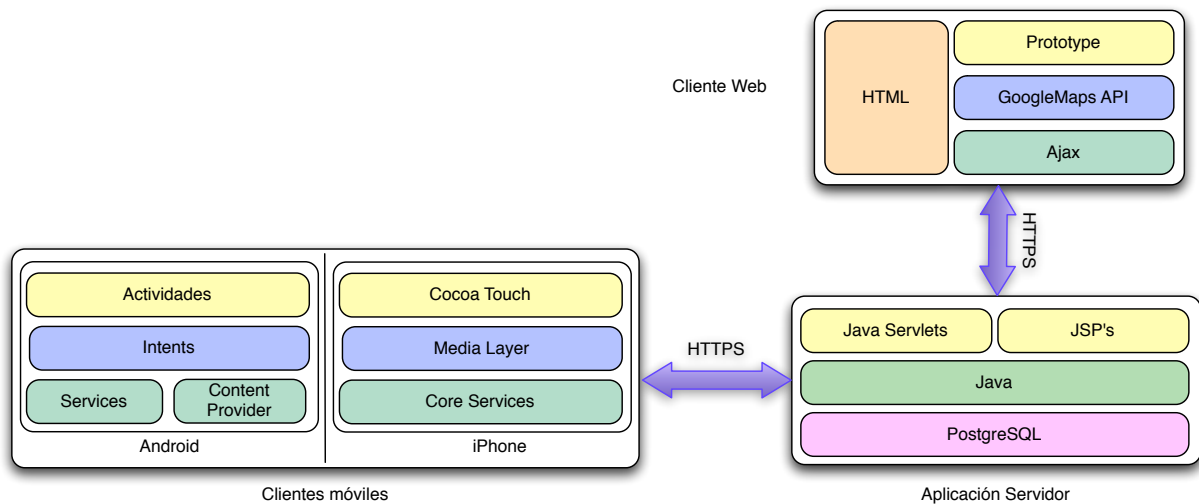


Figura 5.7: Tecnologías utilizadas para el desarrollo de cada componente del sistema.

En la capa del *modelo* se implementa principalmente la lógica de negocios. La capa de la *vista* es la encargada de generar la respuesta a las peticiones de los clientes, mismas que se hacen a través de *JSP's*. La capa de *control* es la responsable de gestionar las otras capas mediante el manejo de peticiones HTTP y la generación de la respuesta adecuada a los clientes.

La aplicación web fue diseñada para operar sobre un *framework* basado en acciones, como Strut 2, mismas que se localizan en la capa del modelo del patrón MVC ilustrado en la figura 5.8. Además, se adoptó un modelo orientado a objetos para la implementación de la funcionalidad requerida. Sin embargo, ello trajo consigo una complicación, ya que sería muy complejo concebir una forma adecuada de agrupar todas las clases del sistema (tipos abstractos de datos del modelo orientado a objetos) en una sola capa de la arquitectura del sistema.

Para resolver el inconveniente que trae consigo la capa del modelo propuesta en la Figura 5.8, se propone la división de la capa del modelo en 3 capas internas, lo que agrega al sistema de las propiedades de mantenibilidad, alta cohesión y bajo acoplamiento. La división se muestra en la Figura 5.9 y se estructura de la siguiente manera:

1. Acciones (amarillo): En la capa superior se implementan las acciones que permiten a la aplicación web atender las peticiones HTTP generadas por los clientes web y móviles. Una acción es un concepto fundamental en las aplicaciones web y representan la más básica unidad de trabajo y éstas pueden asociarse con peticiones HTTP provenientes de los usuarios.

En la capa correspondiente a las *acciones* se puede observar la división transversal

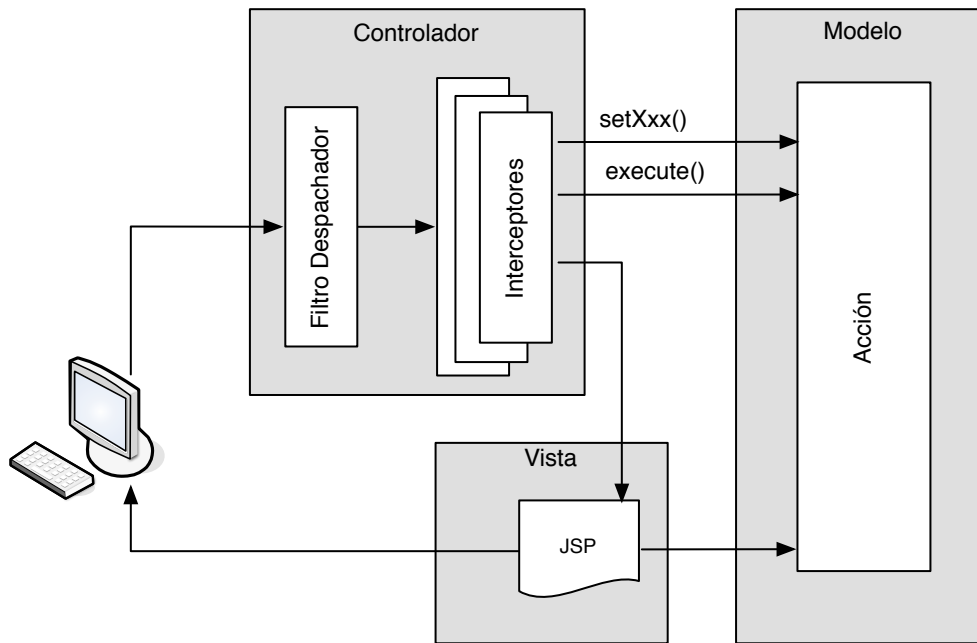


Figura 5.8: Arquitectura general de la aplicación con base en el patrón MVC.

de la capa en 4 secciones: recepción de la ubicación geográfica de los móviles, manejo de peticiones *AJAX*, implementación del mecanismo de control de acceso al sistema y obtención de información del sistema mediante catálogos. Como se puede observar ésta es la capa encargada de atender las peticiones de los clientes y el procesamiento de negocios requerido lo realizan las capas inferiores. Esta capa mantiene oculta la implementación de las capas inferiores de ésta parte del sistema.

2. Lógica de negocios (azul): La capa de lógica de negocios es la encargada de realizar el procesamiento fuerte en la aplicación web, ya que en ella es donde se encuentran las clases que implementan las funciones que permiten:
  - a) Establecer la ubicación geográfica de un dispositivo móvil y realizar el análisis geoespacial que permite determinar si la posición del teléfono celular es objeto de la emisión de una alerta.
  - b) Realizar el seguimiento de los dispositivos móviles que se encuentran en desplazamiento en un determinado momento, mediante la consulta de la ubicación geográfica actual de los mismos.
  - c) Generar mapas temáticos de localización y desplazamiento de los móviles con base en su historial de posicionamiento y un intervalo de tiempo determinado.
  - d) Realizar consultas de la información asociada a las entidades del sistema, así como modificaciones y eliminación de la misma..



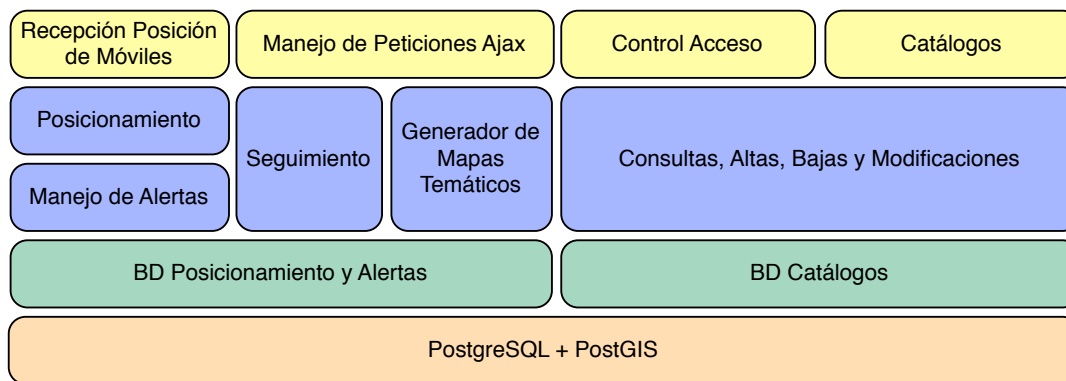


Figura 5.9: División de la capa del modelo de la arquitectura con base en MVC en una arquitectura en 3 capas.

3. Acceso a datos (verde): La capa de acceso a datos contiene las clases que abstraen la funcionalidad necesaria para el acceso a los datos almacenados en la base de datos del sistema. La concepción de una capa cuya funcionalidad exclusiva es obtener, modificar y eliminar la información contenida en la base de datos, dota al sistema de flexibilidad y capacidad de cambio, ya que no cierra la posibilidad de manejar más de un almacén de datos o la migración de uno a otro y que la afectación al sistema en su totalidad sea mínima. Esta capa, a su vez, se divide en dos bloques transversales: el primero, contiene las clases que obtienen la información de posicionamiento de los móviles por medio de consultas espaciales y el segundo, que accede a la información almacenada en los catálogos.

### 5.3.2. Cliente móvil

Para el presente trabajo de tesis se propuso la implementación de un cliente móvil que operara sobre las plataformas *Android* y *iPhone* de dispositivos móviles. Debido a que son plataformas diferentes en ésta sección se exponen la arquitectura de forma general, sin entrar en detalles propios de la implementación en cada una de las plataformas mencionadas. Esta sección contiene los detalles generales del diseño del cliente móvil.

En la Figura 5.10 se muestra la arquitectura general del cliente móvil, que al igual que las arquitecturas de los demás componentes, está basada en un modelo en capas. En la capa superior se encuentra la implementación de las interfaces de usuario, dentro de las que se encuentran la interfaz destinada al control de acceso y la pantalla que le muestra al usuario su ubicación geográfica auxiliándose de un mapa georeferenciado provisto por google.

En la capa denominada *Mecanismo de Localización* se encuentra la implementación del módulo de georeferenciación basado en la tecnología AGPS. Dicho módulo provee la ubi-

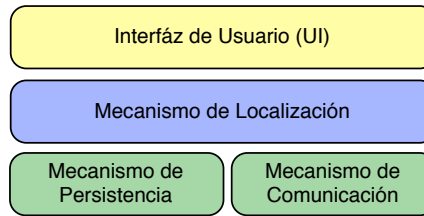


Figura 5.10: Diagrama de la arquitectura general del cliente móvil.

cación geográfica del portador dispositivo móvil mediante la sincronización con el receptor AGPS embebido en el móvil. Dicha ubicación es requerida para visualizar la ubicación en la pantalla del dispositivo móvil o para transmitirse al servidor de la aplicación.

La tercer y última capa del modelo, abstrae la funcionalidad para transmitir y almacenar la ubicación geográfica del dispositivo móvil. El componente de comunicación tiene la finalidad de proveer servicios de comunicación a la Internet mediante el protocolo TCP y transferencia de información a través del protocolo HTTPS. Sin embargo, considerando escenarios en los que no se cuente con conectividad a internet, ya sea por factores relacionados a la cobertura del servicio o incluso por fallas en la red de telefonía celular, es necesario contar con mecanismo que permita el almacenamiento temporal de las posiciones del móvil obtenidas con el GPS y que no fue posible transmitir al servidor de la aplicación.

### 5.3.3. Cliente Web

La visualización de la ubicación geográfica y seguimiento de los clientes móviles se realiza a través de la entidad denominada *Cliente Web*. Para proveer la funcionalidad requerida fue necesario conceptualizar una arquitectura cuyas características permitieran, en el contexto de un navegador web, mantener una comunicación constante con la aplicación web y realizar el procesamiento de la información que ésta le proporcione. En la Figura 5.11 se muestra la arquitectura diseñada para el cliente web, la cual está compuesta por los siguientes componentes:

1. Catálogos: éste componente hace referencia al despliegue estático de hipertexto en el navegador web. Se le denomina catálogos, por que en este componente se encuentra la funcionalidad asociada al registro, modificación y eliminación de las entidades presentes en el sistema, mismas que se detallan en el diccionario de datos de la sección A.3.
2. Visualización de información geográfica: la capa de visualización es la encargada de mostrar en un mapa la ubicación geográfica y la trayectoria de los dispositivos en movimiento, visualización de las regiones de recorrido y mapas temáticos generados con base en el historial de posicionamiento de los móviles.

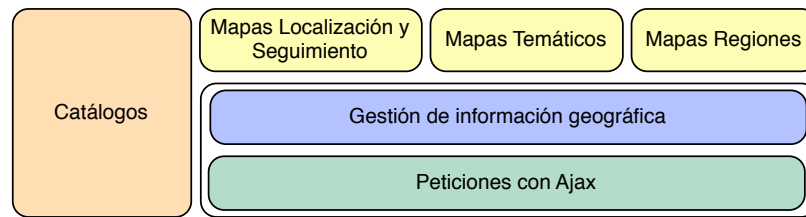


Figura 5.11: Componentes del cliente web

3. Gestión y actualización asíncrona de información geográfica: esta capa es la encargada de gestionar, mediante peticiones asíncronas al servidor de la aplicación, la información de posicionamiento de los dispositivos móviles, además de proveer una interfaz a la capa de visualización que le permite desplegar dicha información en un mapa. La funcionalidad de esta capa se abstraigo en un componente que se le denominó *LocationMapsLibrary* y que se describe a continuación.

### 5.3.3.1. Componente para localización: LocationMapsLibrary

*LocationMapsLibrary* es un componente que tiene la finalidad de gestionar información geográfica. Está basado en la interfaz de programación de *Google Maps V3* y en el paradigma de Programación Orientada a Objetos. Por otro lado, *LocationMapsLibrary* está basado en el patrón de diseño *Fachada* ya que tiene la finalidad de proporcionar una interfaz unificada de alto nivel, para las clases que constituyen la *API* de *GoogleMaps*. Este componente permite a desarrolladores el despliegue de información geográfica, en el ámbito de localización y seguimiento de móviles, sobre un mapa de manera fácil y sencilla.

*LocationMapsLibrary* fue diseñado para proveer funcionalidad que permita visualizar la ubicación y trayectoria de un móvil, dibujar polígonos en un mapa para establecer geocercas realizar peticiones Ajax y establecer geocercas. Este componente se estructuró en 6 paquetes (ver la Figura 5.12):

- *Location*: las clases de este paquete permiten visualizar sobre un mapa la ubicación de un dispositivo móvil. El despliegue de dicha información se realiza con base en un icono personalizable y puede adaptarse según los requerimiento del entorno de la aplicación. La actualización de la ubicación de un móvil se realiza con base al mecanismo de actualización por *polling*.
- *Tracking*: es el paquete diseñado para proveer la funcionalidad necesaria para el despliegue de la ruta que siguen los móviles en desplazamiento. Las clases de éste componente permiten la manipulación de las características visuales de una ruta. Para el despliegue y actualización de la ruta del móvil se toman como base las posiciones obtenidas por el componente *Location*.

- *GeoFence*: aquí se encuentra la funcionalidad necesaria para establecer, de forma visual, la región de desplazamiento permitido para un dispositivo móvil. Esta funcionalidad es proporcionada a través de métodos que permiten dibujar polígonos sobre un mapa mediante arreglos de coordenadas o *click's* sobre el mapa.
- *MapDisplay*: este paquete contiene clases que permiten incluir un mapa en una página web, para lo cual es necesario manipular el árbol del Modelo de Objetos del Documento (DOM, por sus siglas en inglés).
- *Ajax*: en este paquete se abstrae en una clase la funcionalidad necesaria para atender peticiones Ajax de una manera fácil y compatible con todos los navegadores modernos.

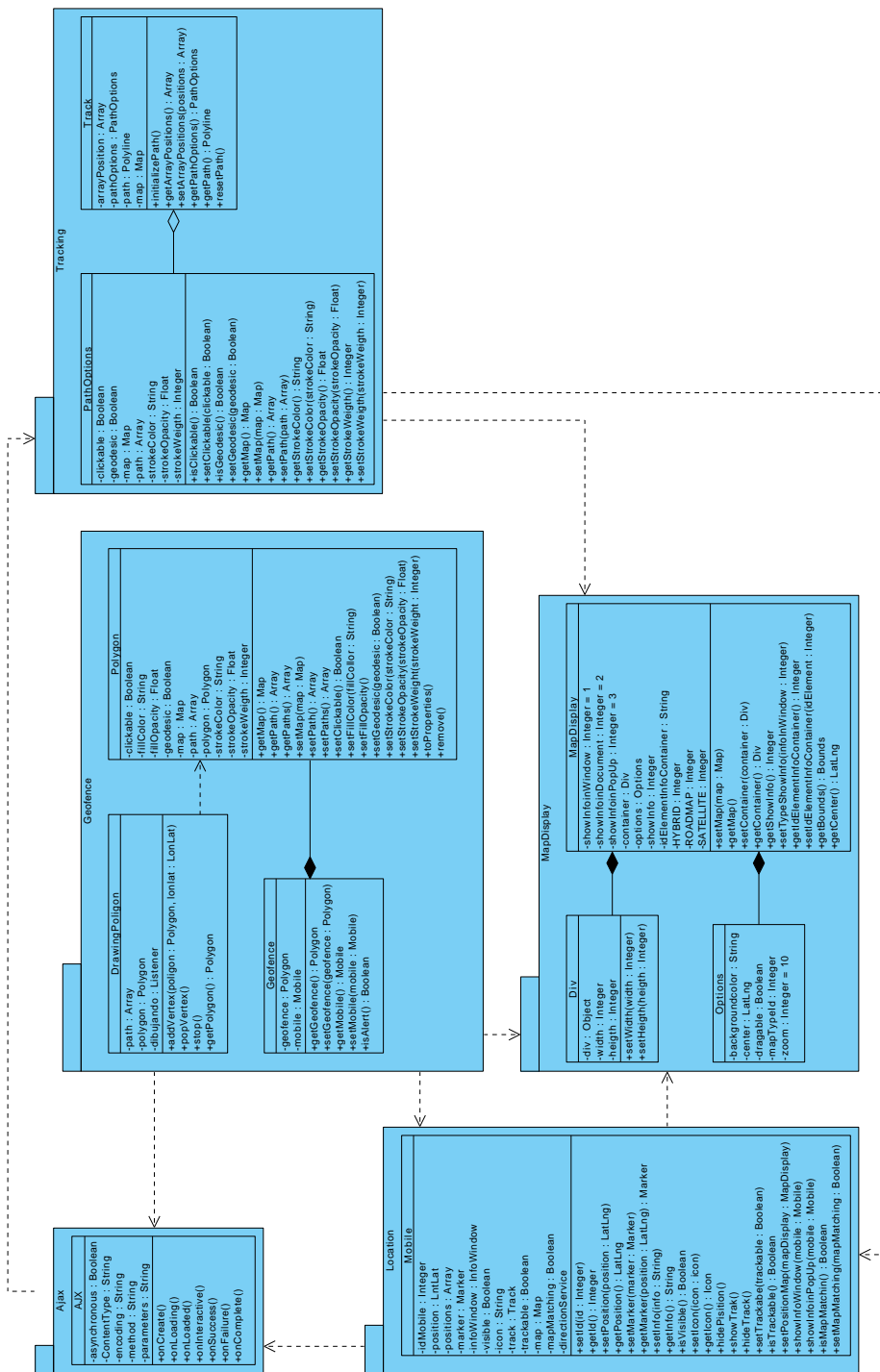


Figura 5.12: Clases del componente para localización basado en la interfaz de programación de Google Maps  
 CINVESTAV-IPN



# Capítulo 6

## Plataforma y tecnologías de desarrollo

En las secciones que contiene el presente capítulo se presentan las tecnologías de desarrollo utilizadas para la implementación de la propuesta a la problemática planteada y el diseño general de los algoritmos implementados. La primera sección contiene la descripción de la plataforma utilizada para la implementación de cada una de las aplicaciones, tales como: *frameworks* utilizados, mecanismos de persistencia y acceso a datos. En la segunda sección se exponen los algoritmos implementados sobre los que recae la funcionalidad de emisión de alertas y generación de historiales de posicionamiento.

### 6.1. Plataforma y tecnologías de desarrollo

En la presente sección se exponen las tecnologías de desarrollo utilizadas para implementar la funcionalidad de la aplicación web, cliente web y cliente móvil. Los puntos a desarrollar son: el *framework* utilizado, la implementación del mecanismo de acceso a datos y el servicio de mapas utilizado.

#### 6.1.1. Aplicación web

Para proveer una Aplicación Web flexible a los cambios, de fácil integración, con facilidad para la implementación de pruebas de unidad y con capacidad para el manejo de información geoespacial se utilizó la tecnología que se describe en ésta sección. En la Figura 6.1 se observa la forma en como se integraron las tecnologías utilizadas para la implementación de la aplicación web: Struts 2, Spring, Hibernate, Hibernate Spatial, PostgreSQL y Postgis.

##### 6.1.1.1. Frameworks de desarrollo

Apache Struts 2 es un marco de desarrollo extensible para la creación de aplicaciones web basadas en *Java*. Este *framework* está diseñado para simplificar el ciclo de desarrollo completo; desde la construcción, seguido de la implementación y, finalmente, para el mantenimiento de aplicaciones a través del tiempo. *Apache Struts 2* fue originalmente conocido

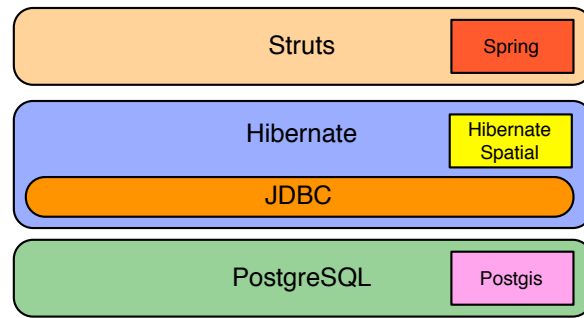


Figura 6.1: Tecnologías utilizadas para la implementación de la aplicación web

como *WebWork 2*. Después de trabajar de forma independiente durante varios años, las comunidades WebWork y Struts se unieron para crear Struts2. El marco de trabajo Struts está diseñado para ayudar a los desarrolladores a crear aplicaciones web que se basan en una arquitectura MVC. En éste marco el modelo representa el código de negocios o de base de datos, la vista la representa el diseño de páginas y el Contralor representa la navegación.

El *Framework Spring* proporciona un amplio modelo de programación y configuración de las modernas aplicaciones empresariales basadas en Java, en cualquier tipo de plataforma de implementación. Un elemento clave de Spring es el apoyo a la infraestructura a nivel de aplicación. Spring se centra en la interconexión de los componentes de las aplicaciones de manera que los equipos de desarrollo pueden centrarse en la lógica de negocio a nivel de aplicación, sin ataduras innecesarias a los entornos específicos de visualización y despliegue de información. Spring provee:

- Inyección de dependencia flexible con estilos de configuración en archivos XML y otros basados en anotaciones.
- Soporte avanzado para programación orientada a aspectos.
- Soporte para transacciones, validaciones y cache declarativo.
- Soporte para trabajar con las especificaciones más comunes de Java EE, tales como: JDBC (Java DataBase Connectivity), JPA (Java Persistence API), JTA (Java Transaction API) y JMS (Java Message Service).
- Fácil integración con otros *frameworks* de código abierto como Hibernate.
- Un marco flexible para la creación aplicaciones Web REST basadas en MVC.
- Fácil configuración de pruebas unitarias y de integración.



Límite	Valor
Tamaño máximo de BD	Sin límite
Tamaño máximo de tabla	32 TB
Tamaño máximo de tupla	1.6 TB
Tamaño máximo de campo	1GB
Número máximo de renglones	Sin límites
Número máximo de columnas por tabla	250 - 1600 (dependiendo del tipo de dato)
Número máximo de índices por tabla	Sin límite

Tabla 6.1: Características principales de PostgreSQL

### 6.1.1.2. Almacenamiento y acceso a datos

Como todo sistema de información, el sistema implementado incluye un mecanismo que le permite gestionar la gran cantidad de datos que genere el proceso de localización y seguimiento de los dispositivos móviles. Lo ideal es pensar en un utilizar una Base de Datos y en un Sistema Gestor de Base de Datos (SGBD) para manejar la información mencionada. Sin embargo, debido a que la propuesta requiere de gestión de información de localización es necesario que el SGBD soporte información espacial.

Por lo mencionado en el párrafo anterior, se adoptó como SGBD a *Postgres*; el cual es un poderoso sistema Gestor de Base de Datos Relacional Orientado a Objetos y publicado bajo licencia BSD y que en ambientes de producción maneja más de 4 terabytes de información. La arquitectura de *Postgres* garantiza fiabilidad, integridad y correctitud de la información. *Postgres* tiene la facilidad de ejecutar procedimientos almacenados desde una gran variedad de lenguajes de programación (Java, Perl, Python, C++, etc) e interfaces de comunicación para interactuar con ellos. Además, tiene versiones para operar sobre los sistemas operativos más utilizados. En la tabla 6.1 se muestra un resumen de las principales capacidades de *Postgres*.

Sin embargo, para el almacenamiento de información espacial *Postgres* no es suficiente. Por lo cual fue necesario integrar *PostGIS*, ya que agrega soporte para objetos geográficos a una base de datos objeto-relacional, lo que permite a los SIG utilizar a *Postgres* como sistema gestor de base de datos espaciales. *Postgis* ha sido desarrollado como un proyecto en la tecnología de base de datos espacial de código abierto y es liberado bajo la Licencia Pública General de GNU.

Al trabajar con software orientado a objetos y bases de datos relacionales el tiempo de desarrollo puede ser alto. Los costos de desarrollo son significativamente más altos debido a un desajuste entre el paradigma de cómo los datos son representados en los objetos frente a bases de datos relacionales. Por lo que para la implementación de la propuesta fue necesario adoptar un mecanismo eficiente de acceso a datos, que facilitara el desarrollo al mismo tiempo que redujera el tiempo necesario para ello. La herramienta adoptada para

Función	Postgres	Oracle	MySQL	SQLServer
Básicas de Geometría	✓	✓	✓	✓
Revisión de relaciones espaciales entre objetos geográficos	✓	✓	✓	✓
Análisis espacial	✓	✓	x	✓

Tabla 6.2: Visión general de las funciones soportadas por cada Sistema Gestor de Base de Datos.

el acceso a datos fue Hibernate, ya que reduce los tiempos de desarrollo permitiendo a los desarrolladores hacer mapeos de una base de datos relacional en objetos java. El término mapeo objeto/relacional se refiere a la técnica de mapeo de datos entre una representación del modelo de objetos a una representación del modelo de datos relacional.

Hibernate se encarga de la asignación de clases Java a tablas de la base de datos y del mismo modo del tipo de datos Java a tipos de datos SQL. Además, proporciona mecanismos de consulta y recuperación de datos, reduciendo el manejo manual de los datos con SQL y JDBC. La meta de diseño de Hibernate es quitarle al desarrollador el 95% de las tareas comunes de programación de acceso a datos, al eliminar la necesidad de crear consultas manuales en SQL.

Hibernate por si solo no es suficiente para la propuesta ya que no provee soporte para datos espaciales. Por lo que fue necesario agregar a la plataforma de desarrollo *Hibernate Spatial*, que es una extensión genérica de Hibernate para el manejo de datos geográficos, es de código abierto y liberada con licencia LGPL. Hibernate Spatial proporciona soporte a datos geográficos y una interfaz estandarizada para almacenamiento y funciones de consulta de información geográfica para diversos SGBD. En la Tabla 6.2 se muestra el soporte que proporciona Hibernate Spatial para los SGBD más utilizados.

### 6.1.2. Cliente móvil

Debido a que el cliente web considerado en la propuesta se pensó para dos plataformas diferentes de dispositivos móviles, fue necesario realizar dos implementaciones independientes. Una implementación desarrollada sobre la plataforma Android y otra sobre la plataforma iPhone OS. Como se mencionó en la sección 4.1.3 la amplia diversidad de plataformas para dispositivos móviles es considerada como una dimensión del cómputo móvil, es decir, un campo de oportunidad para el desarrollo de aplicaciones y a la vez un reto a resolver. Para implementar las aplicaciones en las plataformas mencionadas se requirió de la siguiente plataforma:

**Android:** Android es una pila de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones principales. Android posee un SDK por medio del cual proporciona las herramientas y API's necesarias para desarrollar aplicaciones

usando el lenguaje de programación Java. En la Figura 6.2 se muestran la arquitectura y los componentes que la integran.

- **Aplicaciones:** Android provee un conjunto de aplicaciones básicas: un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos, entre otras. Todas las aplicaciones se desarrollan usando el lenguaje de programación Java.
- **Framework para aplicaciones:** al proporcionar una plataforma de desarrollo abierta, Android ofrece a desarrolladores la capacidad de crear aplicaciones ricas e innovadoras. Los desarrolladores son libres para tomar ventaja del hardware del dispositivo, acceso a la información ubicación, ejecutar servicios en segundo plano, establecer alarmas y mucho más. También tienen acceso completo a la misma API utilizada para desarrollar las aplicaciones básicas. La arquitectura de la capa de aplicación está diseñada para simplificar la reutilización de componentes.
- **Librerías:** Android incluye un conjunto de bibliotecas C/C++ utilizadas por los diversos componentes del sistema Android. Estas capacidades están a disposición de los desarrolladores a través del marco de aplicación para Android.
- **Runtime:** Android incluye un conjunto de bibliotecas de núcleo que proporcionan la mayor parte de la funcionalidad disponible en las bibliotecas del lenguaje de programación Java. Cada aplicación Android se ejecuta en su propio proceso, con su propia instancia de la máquina virtual de Dalvik. Dalvik ha sido escrito para que un dispositivo pueda ejecutar varias máquinas virtuales de manera eficiente.

La máquina virtual Dalvik se basa en el kernel de Linux para proveer la funcionalidad fundamental, como el manejo de hilos de ejecución y la gestión de memoria de bajo nivel.

- **Kernel:** Android se basa en la versión 2.6 del kernel de Linux para los servicios básicos del sistema como: la seguridad, la gestión de memoria, gestión de procesos, funciones de red y el modelo de controlador. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Dentro del marco de desarrollo de aplicaciones en Android la propuesta requiere consumir los *Servicios de Localización* proporcionados a través del paquete *android.location* y la librería para *SQLite*. El componente principal en el marco de ubicación es el servicio del sistema *LocationManager*, que proporciona las API's para determinar la ubicación del dispositivo. La plataforma provee distintas fuentes para obtener la ubicación: AGPS, Cell-ID y Wi-Fi. Para efectos de la propuesta, el desarrollo se realizó utilizando el servicio de localización basado en AGPS. Por su parte *SQLite* es un motor de base de datos relacional potente y ligero disponible para todas las aplicaciones en la plataforma.

**iPhone iOS:** Este sistema operativo gestiona el hardware del dispositivo y proporciona las tecnologías necesarias para implementar aplicaciones nativas. El sistema operativo



Figura 6.2: Principales componentes del Sistema Operativo Android

también viene con aplicaciones base, tales como: teléfono, correo y navegador web, que prestan servicios estándar del sistema al usuario. El SDK de iOS contiene las herramientas e interfaces necesarias para desarrollar, instalar, ejecutar y probar las aplicaciones nativas. Las aplicaciones nativas se construyen utilizando los frameworks del sistema iOS y el lenguaje Objective-C y se ejecutan directamente en iOS.

La arquitectura de iOS es similar a la arquitectura básica que se encuentra en Mac OS X. En el nivel más alto, iOS actúa como un intermediario entre el hardware y las aplicaciones de usuario, ya que rara vez éstas interactúan directamente con el hardware. La implementación de las tecnologías de iOS puede ser visto como un conjunto de capas, que se muestran en la Figura 6.3. En las capas más bajas del sistema se encuentran los servicios fundamentales y las tecnologías en las que se basan todas las aplicaciones, mientras que las capas de más alto nivel contienen servicios y tecnologías más sofisticadas.

- **Cocoa Touch:** esta capa define la infraestructura básica de las aplicaciones y soporte para las principales tecnologías como: la multitarea, entradas basadas en el contacto (touch), notificaciones push y muchos servicios de alto nivel del sistema.
- **Media:** la capa contiene los medios dirigidos al manejo de: gráficos, audio, video

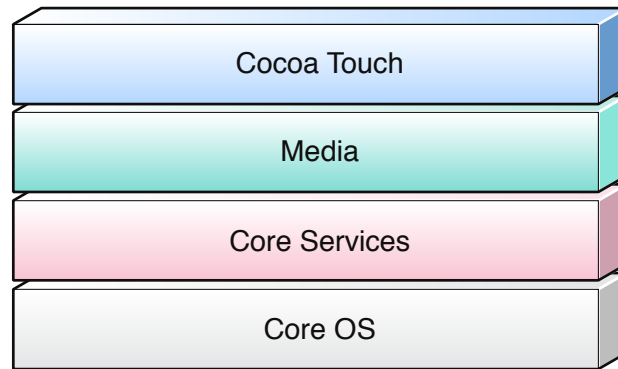


Figura 6.3: Capas de iOS

y tecnologías orientadas a crear la mejor experiencia multimedia disponible en un dispositivo móvil.

- **Core Services:** esta capa contiene los servicios fundamentales que todas las aplicaciones utilizan.
- **Core OS:** esta capa contiene todos los servicios de bajo nivel sobre los cuales están construidas todas las aplicaciones y *framework* de la plataforma. Si en algún momento se requiere interactuar directamente con el hardware del dispositivo, se recomienda se realice a través de las funciones provistas por esta capa.

Lo principal que le concierne a la propuesta se encuentra en la capa *Core Services*, específicamente en *Core Location Framework* y la librería de *SQLite*. El *Core Location Framework* (`CoreLocation.framework`) proporciona acceso a la información de ubicación por medio de: dispositivo AGPS, red celular o redes inalámbricas (WiFi) a las que se encuentre conectado.

Por su parte, la biblioteca *SQLite* permite incrustar una ligera base de datos SQL en una aplicación sin necesidad de ejecutar un proceso en el servidor remoto de base de datos. Esto permitió crear archivos de bases de datos locales desde la aplicación, gestionar las tablas y registros existentes en los archivos.

## 6.2. Implementación de los principales algoritmos

En la presente sección se presentan los algoritmos en los que recae la funcionalidad más sobresaliente de la propuesta.

### 6.2.1. Emisión de alertas

Para verificar si un dispositivo móvil ha abandonado la región geográfica que le fue asignada, cada vez que un dispositivo móvil envía su ubicación se verifica si ésta se encuentra

dentro o no de la región que tiene asociada (ver Algoritmo 1). Esta verificación se diseñó para realizarse a nivel de la base de datos utilizando un *trigger* y funciones para revisar relaciones espaciales entre objetos geográficos. La complejidad del algoritmo está en función de la complejidad de la función *contains* de *postgis*.

---

**Algoritmo 1** Emisión de una alerta de región abandonada.

---

**Entrada:** El polígono  $G$  de la geocerca y la posición actual  $p$ .

**Salida:** *true* si se debe emitir la alerta o *false* en caso contrario.

```
ALERTA_GEOCERCA(  $G$  ,  $p$  )
```

```
   $contiene \leftarrow G.contains(p)$ ;
```

```
  if  $contiene = true$  then
```

```
     $alerta \leftarrow false$ 
```

```
  else
```

```
     $alerta \leftarrow true$ 
```

```
  end if
```

```
  return  $alerta$ 
```

---

Una alerta de intervalo de tiempo se emite cuando un dispositivo móvil no ha tenido un desplazamiento significativo durante un tiempo definido. En el Algoritmo 2 se muestra la función implementada para emitir o no la alerta. Es importante mencionar se debe cumplir con la restricción  $R \geq meAGPS$  (meAGPS, máximo error de la tecnología AGPS). La complejidad del algoritmo es  $O(n)$ , ya que el algoritmo utilizado para medir la distancia entre dos puntos es constante en función de sus entradas.

---

**Algoritmo 2** Emisión de una alerta de tiempo.

---

**Entrada:** Arreglo  $P$  con las  $n$  posiciones reportadas dentro el intervalo de tiempo ( $\Delta t$ ) y la distancia  $R$  que indica un desplazamiento significativo.

**Salida:** *true* si se debe emitir la alerta o *false* en caso contrario.

```
ALERTA_TIEMPO(  $P$  ,  $R$  )
```

```
   $alerta \leftarrow true$ ;
```

```
  for  $i \leftarrow 1$  to  $i = n - 1$  or  $alerta = false$  do
```

```
     $alerta \leftarrow alerta \text{ AND } |P[0].distancia(P[i])| < R$ ;
```

```
  end for
```

```
  return  $alerta$ 
```

---

### 6.2.2. Historiales de posicionamiento

Para la implementación de la generación de historiales de posicionamiento fue necesario implementar los Algoritmos 3 y 4. Para la generación del historial de posicionamiento con base en un intervalo de tiempo (Algoritmo 3) se realiza lo siguiente:

1. Indicar el dispositivo móvil y el intervalo de tiempo para el que se quiere generar el historial.

2. Se realiza una consulta de las sesiones iniciadas desde el dispositivo móvil que se intersecten con el intervalo de tiempo dado.
3. Si existen sesiones del móvil:
  - a) Para cada sesión se obtienen las posiciones que se encuentran dentro del intervalo proporcionado y se agregan al historial.

Como se puede observar la complejidad del algoritmo implementado para la generación del historial de posicionamiento para un intervalo de tiempo dado es  $O(n \cdot m)$ ; donde  $n$  es el número de sesiones del dispositivo móvil dentro del intervalo y  $m$  es el número de posiciones dentro de cada recorrido.

---

**Algoritmo 3** Generación de un historial de posicionamiento con base en un intervalo de tiempo.

---

**Entrada:** Dispositivo móvil ( $v$ ), fecha inicial ( $f_i$ ) y fecha final ( $f_f$ ).

**Salida:** *historial* con la información de los recorridos del móvil, *null* si no hay recorridos en el intervalo indicado.

```

HISTORIAL_POSICIONAMIENTO( $v, f_i, f_f$ )
historial  $\leftarrow$  null;
if  $f_i < f_f$  then
  S  $\leftarrow$  RecorridosVehiculo( $v, f_i, f_f$ );
  if S.length > 0 then
    for  $i \leftarrow 1$  to  $i = n$  do
      trayectoria[i]  $\leftarrow$  obtenerPosiciones( $S[i], f_i, f_f$ );
      historial.agregar(trayectoria[i]);
    end for
  end if
end if
return historial;

```

---

El algoritmo implementado para la generación del historial de posicionamiento con base en un punto de interés es una especialización del algoritmo utilizado para el historial con base en un intervalo de tiempo. La diferencia entre los dos algoritmos radica en que el primero (Algoritmo 3) se enfoca en construir un historial para un dispositivo en específico, mientras que el segundo (Algoritmo 4) tiene el objetivo de construir un historial con base en un punto de interés y los dispositivos que hayan estado cercanos a él. Los pasos del algoritmo son:

1. Indicar el punto de interés y el intervalo de tiempo para el que se quiere generar el historial.
2. Se realiza una consulta de las sesiones iniciadas desde el dispositivo móvil que se intersecten con el intervalo de tiempo dado.
3. Si existen sesiones del móvil:

- a) Para cada sesión se obtienen las posiciones que se encuentran dentro del intervalo proporcionado y se agregan al historial.
- b) Se verifica cuales de las posiciones obtenidas en el punto anterior se encuentran dentro del área de interés definida.

---

**Algoritmo 4** Generación de un historial de posicionamiento con base en un intervalo de tiempo y un punto de interés.

---

**Entrada:** Fecha inicial ( $f_i$ ), Fecha final ( $f_f$ ), punto de interés ( $P$ ) y radio ( $r$ ).

**Salida:** *historial* con la información de los recorridos del móvil, *null* si no hay recorridos en el intervalo indicado.

**HISTORIAL\_POSICIONAMIENTO\_PI**( $f_i, f_f, P, r$ )

historial  $\leftarrow$  null;

**if**  $f_i < f_f$  **then**

RV  $\leftarrow$  RecorridosVehiculos( $f_i, f_f$ );

**if** RV.length > 0 **then**

**for**  $i \leftarrow 1$  **to**  $i = n$  **do**

trayectoria\_tem  $\leftarrow$  obtenerPosiciones(RV[ $i$ ],  $f_i, f_f$ );

trayectoria[ $i$ ]  $\leftarrow$  obtenerPosicionesPI(trayectoria\_tem,  $P, r$ );

historial.agregar(trayectoria[ $i$ ]);

**end for**

**end if**

**end if**

**return** *historial*;

---

Como se puede observar la complejidad del algoritmo 4 es  $O(n \cdot (m + pC))$ ; donde  $n$  el número de sesiones,  $m$  el número de posiciones en cada sesión,  $p$  es el número de posiciones que se encuentran dentro del intervalo de tiempo dado y  $C$  es la complejidad de la función *contains* de *postgis*.



# Capítulo 7

## Pruebas y resultados

El presente capítulo contiene la pruebas a las que se sometió la implementación realizada como solución al problema planteado y los resultados obtenidos de las mismas. Los aspectos objeto de prueba en el presente trabajo de tesis son: precisión y tiempos de respuesta del AGPS presente en los dispositivos *Android* y *iPhone*, así como el tiempo que consumen los algoritmos de emisión de alertas implementados. En los resultados obtenidos de las pruebas a las que se sometió el AGPS de los dispositivos mencionados se analizaron los tiempos de respuesta obtenidos contra los establecidos en la especificación de la tecnología. Las características de los dispositivos móviles sobre los cuales se realizaron las pruebas son:

### 1. Android HTC Hero

- Android<sup>TM</sup> 2.1
- Procesador Qualcomm<sup>®</sup> MSM7200A<sup>TM</sup>
- HSPA/WCDMA y conexión cuatribanda GSM/GPRS/EDGE
- Conectividad inalámbrica vía WiFi y Bluetooth
- Antena GPS interna
- > 7 hrs de batería en modo llamada
- 512 MB en ROM
- 288 MB en RAM
- 8 GB en almacenamiento

### 2. iPhone

- iOS 4
- Soporta UMTS/HSDPA/HSUPA y GSM/EDGE
- Conectividad inalámbrica via WiFi y Bluetooth
- Dispositivo AGPS interno
- > 7 hrs de batería en modo llamada con 3G y > 14 hrs 2G

- 16 GB de almacenamiento en dispositivo flash
- 512 MB en RAM

Por otro lado, el equipo utilizado para las pruebas cuenta con las siguientes características:

- MacBook Pro 7.1
  - Mac OS X 10.6.6
  - Core 2 Duo a 2.4 GHz
  - 4 GB en RAM
  - 250 GB en DD

## 7.1. Tiempo de respuesta y precisión del AGPS

En primera instancia se muestran los resultados obtenidos de realizar el posicionamiento con el dispositivo AGPS en frío, con la finalidad de medir el tiempo de respuesta de la primera posición (TTFF), ver Tabla 7.1 y Figura 7.1. De los datos obtenidos se corrobora que la precisión del AGPS de los dispositivos móviles es el esperado y descrito en la literatura. De los resultados obtenidos se puede determinar que el tiempo necesario para obtener la primera ubicación es muy similar en ambos dispositivos y éste no afecta la aplicabilidad de telefonos celulares en el ambito de localización y seguimiento.

Dispositivo móvil	Nº de posiciones	$t(s)$	Media	Desviación estándar
iPhone	357	45	51.53	9.7
	98	60		
	45	77		
hTC Hero	300	43	57.55	17.97
	115	77		
	53	80		
	32	87		

Tabla 7.1: Tiempo para obtener la primera posición (TTFF).

Por otro lado, para medir la precisión del dispositivo AGPS de los teléfonos celulares con los que se estuvo trabajando a lo largo del presente trabajo de tesis, se realizó un plan de pruebas en los que los dispositivos se encontraran dos tipos de escenarios:

1. Escenario favorable: uno que favorezca la naturaleza de la tecnología.
  - Al aire libre,
  - Condiciones climáticas favorables (cielo despejado, sin lluvia, nivel de contaminación bajo),

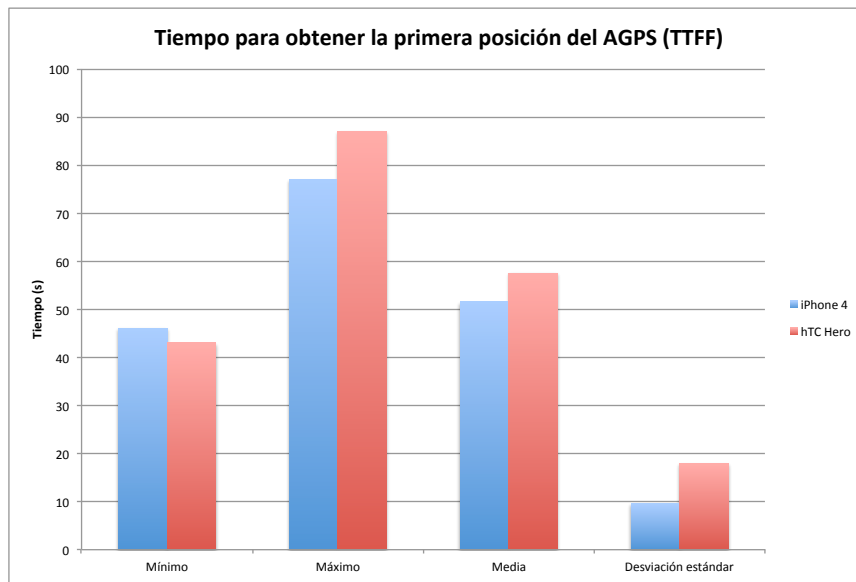


Figura 7.1: Gráfica donde se ilustra el TTFF del AGPS de los dispositivos utilizados.

- Ausencia de edificios que cubran el la visión al cielo, etc
2. Escenario desfavorable: escenario en el que haya presente fuentes que pueden afectar el correcto funcionamiento del AGPS.
- El interior de una construcción (casa, edificio, etc),
  - Condiciones climáticas desfavorables (cielo nublado, lluvia, altos niveles de contaminación, etc),
  - Visibilidad escasa al cielo por presencia de edificios, etc.

El mecanismo adoptado para medir la precisión y exactitud de posicionamiento tomó como base el servicio de mapas de *Google*, estableció un punto de referencia (coordenada geográfica) y con base en él se realizó la medición del margen de error de la ubicación proporcionada por el AGPS, ver Figura 7.2. Además, para efectos de la prueba se categorizó el margen de error como se muestra en la Tabla 7.2. Es importante mencionar que para la realización de la prueba el AGPS de los dispositivos se configuró para obtener la máxima precisión, ya que es el objetivo de la medición.

La prueba de precisión para el escenario 1 se realizó en las instalaciones deportivas del Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional obteniendo los resultados contenidos en la Tabla 7.3 y la gráfica de la Figura 7.3. Es importante mencionar que las posiciones para las que se tubo un margen de error mayor se refieren a las primeras posiciones obtenidas del AGPS y que con base en peticiones periódicas el margen de error se reduce y estabiliza.

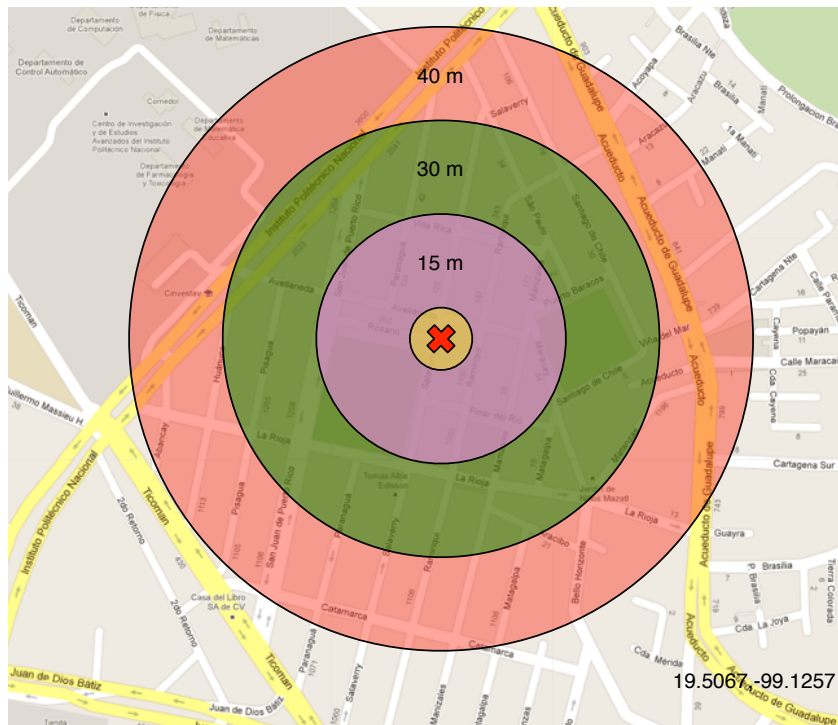


Figura 7.2: Ilustración de método adoptado para medir la precisión del AGPS.

Margen de error ( $e$ )	Categoría ( $m$ )	Margen de error ( $e$ )	Categoría ( $m$ )
$0 \leq e \leq 5$	5	$25 < e \leq 30$	30
$5 < e \leq 10$	10	$30 < e \leq 35$	35
$10 < e \leq 15$	15	$\vdots$	$\vdots$
$15 < e \leq 20$	20	$90 < e \leq 95$	95
$20 < e \leq 25$	25	$95 < e \leq 100$	100

Tabla 7.2: Categorización del margen de error para la prueba de precisión del AGPS.

iPhone				hTC Hero			
Posiciones (%)	Categoría	$m$	$\sigma$	Posiciones (%)	Categoría	$m$	$\sigma$
4	5			6	5		
51	10			47	10		
23	15	13.14	4.84	19	15	13.8	5.34
17	20			21	20		
5	25			7	25		

Tabla 7.3: Resultados obtenidos en la prueba de precisión realizada bajo condiciones ideales.

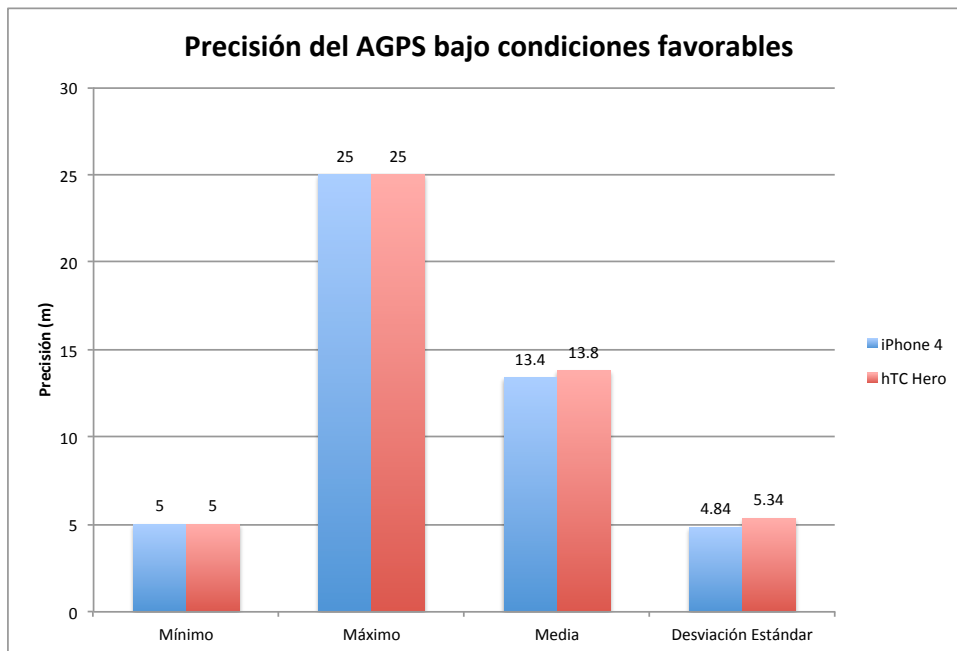


Figura 7.3: Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones favorables a la tecnología de localización.

La prueba de precisión para el escenario 2 se realizó sobre la Av. Reforma, de la Ciudad de México en el tramo que entre el *Monumento a Cuahutémoc* y la glorieta de la *Diana Cazadora*. La elección de esta avenida es por que sobre ella se encuentran edificios altos, arbolado, altas fuentes de ruido y no se cuenta con cielo totalmente abierto, lo que perjudica la tecnología GPS. Además, la prueba se realizó bajo las siguientes condiciones climáticas (información obtenida del Servicio Meteorológico Nacional de la Comisión Nacional de Agua):

- Temperatura mínima: 8 °C
- Visibilidad 6 Km
- Velocidad del viento: 5 Km/h
- Humedad: 87 %
- Precipitación: 60 %
- Radiación UV: 10 (Muy alta)

En la Tabla 7.4 y la Figura 7.4 se muestran los resultados de la prueba efectuada bajo el escenario 2. De los resultados se puede resaltar que el promedio de error obtenido, 24.86 m para el iPhone y 26.7 m para el hTC Hero, es aceptable para aspectos de localización teniendo en cuenta que las condiciones bajo las cuales se realizó la prueba se asemeja a lo que se podría denominar el peor de los casos. En las gráficas de las Figuras 7.5 y 7.6 se

iPhone				hTC Hero			
Posiciones (%)	Categoría	$m$	$\sigma$	Posiciones (%)	Categoría	$m$	$\sigma$
14	10			13	10		
21	15			18	20		
35	20	24.85	13.44	40	25	26.7	10.84
17	40			18	35		
13	50			11	50		

Tabla 7.4: Resultados obtenidos en la prueba de precisión realizada bajo condiciones desfavorables.

muestra una comparativa en la precisión de los dispositivos, observando en ambos casos que la diferencia entre la precisión obtenida por un teléfono y otro no tiene diferencias significativas.

## 7.2. Localización y seguimiento

Para medir el buen funcionamiento y rendimiento del mecanismo implementado para la localización y seguimiento de los dispositivos móviles, se implementaron pruebas de unidad; donde se construyeron objetos para simular dispositivos móviles en desplazamiento. La simulación de dispositivos móviles consistió en asignar a cada objetivo un archivo de texto donde se encontraba una ruta a seguir, del cual se extraería cada terminado intervalo de tiempo una coordenada y se enviaría al módulo de localización y seguimiento. La estructura del archivo es la siguiente:

```

1 # Ruta de simulacion 1: AV. Politecnico
2 #EPSG:4326 WGS84
3 19.472658,-99.137449
4 19.474358,-99.137105
5 19.476259,-99.136655
6 19.477958,-99.136161
7 19.479617,-99.135754
8 19.481276,-99.135410
9 19.483137,-99.134981
10 19.484594,-99.134681
11 19.486455,-99.134316
12 19.488336,-99.133887
13 19.490278,-99.133608
14 19.492139,-99.133200
15 19.494101,-99.132792
16 19.496265,-99.132363
17 19.498389,-99.131977
18 19.500392,-99.131655

```

En la gráfica de la Figura 7.7 se muestra el tiempo que requiere el mecanismo de localización, para almacenar la ubicación de un dispositivo y someterla al análisis de emisión de alertas. La prueba que se presenta en la gráfica corresponde a una simulación de un total de hasta 100 dispositivos transmitiendo su ubicación de forma concurrente a la aplicación web.

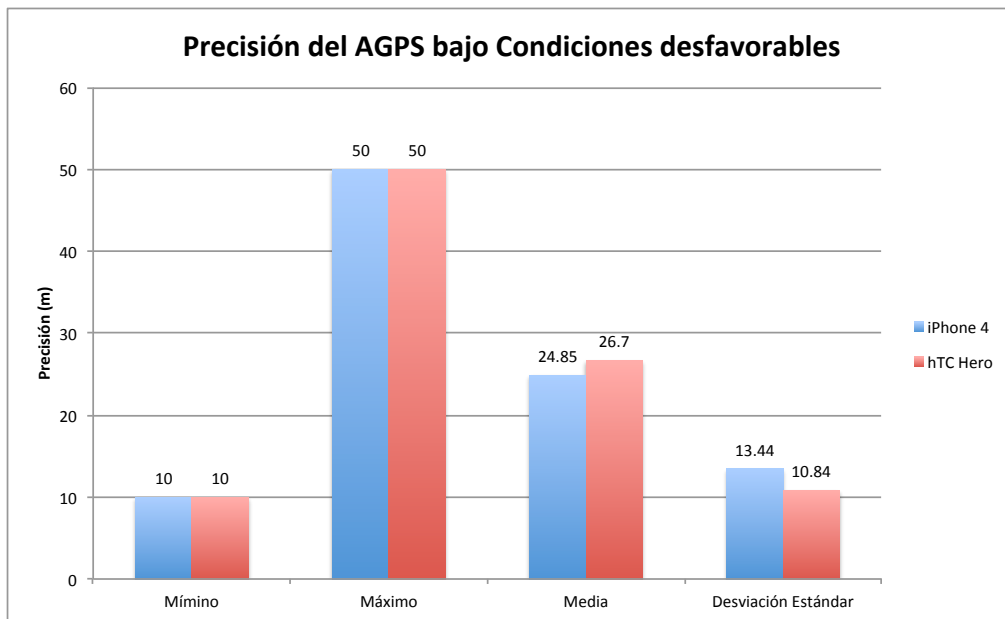


Figura 7.4: Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones desfavorables a la tecnología de localización.

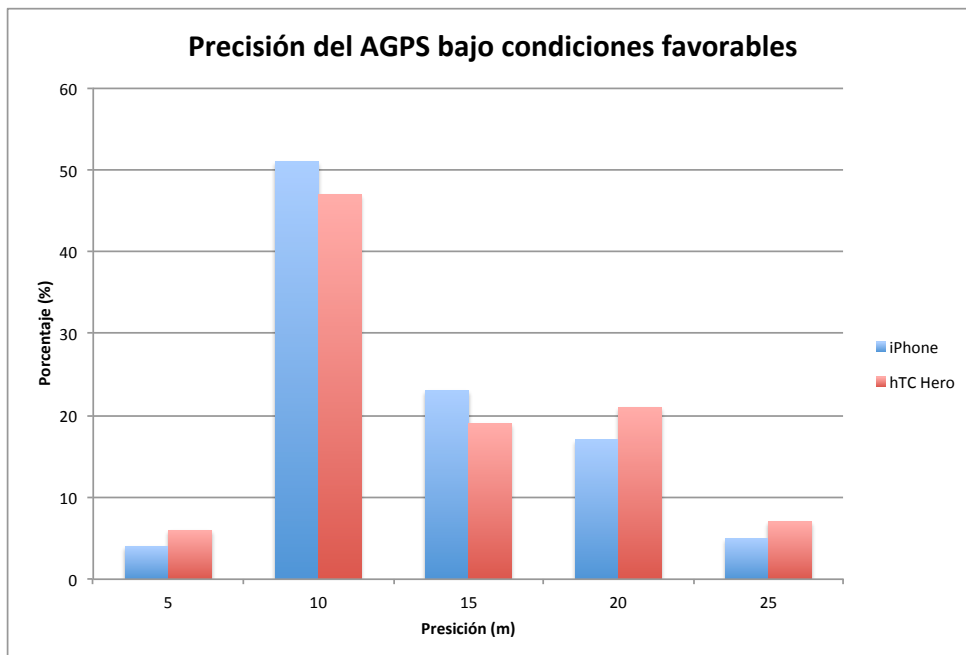


Figura 7.5: Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones favorables a la tecnología de localización utilizada.

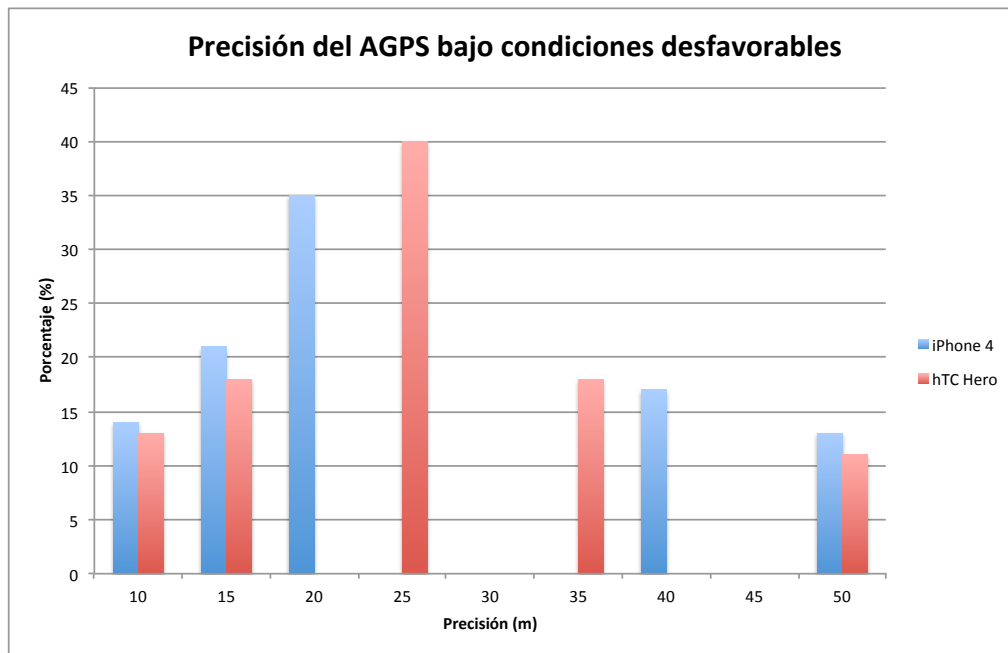


Figura 7.6: Gráfico de la precisión del AGPS del iPhone y hTC Hero bajo condiciones no favorables a la tecnología de localización.

Por otro lado, en la gráfica de la figura 7.8 se muestra el tiempo necesario para la visualización de la ubicación geográfica de los dispositivos simulados. Es importante mencionar que en las gráficas no se ve reflejada la latencia producto de la transferencia de información ya que las pruebas se realizaron en un ambiente de red local con poca demanda.

### 7.3. Emisión de alertas

Para probar el funcionamiento del proceso de emisión de alertas de región abandonada se tomaron dos especificaciones: regionales de forma regular y con un número de vértices menor a diez y regionales de mayor cobertura con número de vértices entre 100 y 1300. Para éste último caso se utilizaron las delegaciones políticas del distrito federal como regiones de recorrido. Para realizar la prueba se utilizó solo un móvil, esto con la finalidad de medir el tiempo que consume la funcionalidad en el caso en que solo hay un móvil en desplazamiento.

En las Figuras 7.9 y 7.10 se puede observar el tiempo necesario para realizar el análisis de emisión de alertas de region abandonada. En ellas se puede observar que para regiones pequeñas y con pocos vértices el tiempo necesario es más o menos constante. Sin embargo, para regionales de mayor cobertura geográfica y de forma completamente irregular el tiempo requerido se incrementa.



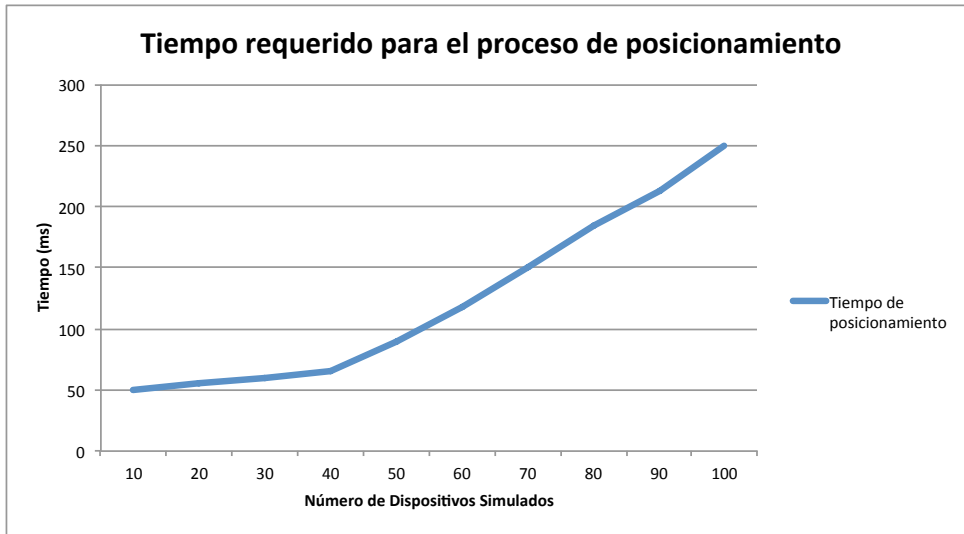


Figura 7.7: Gráfica en la que se muestra el tiempo requerido para el proceso de posicionamiento.

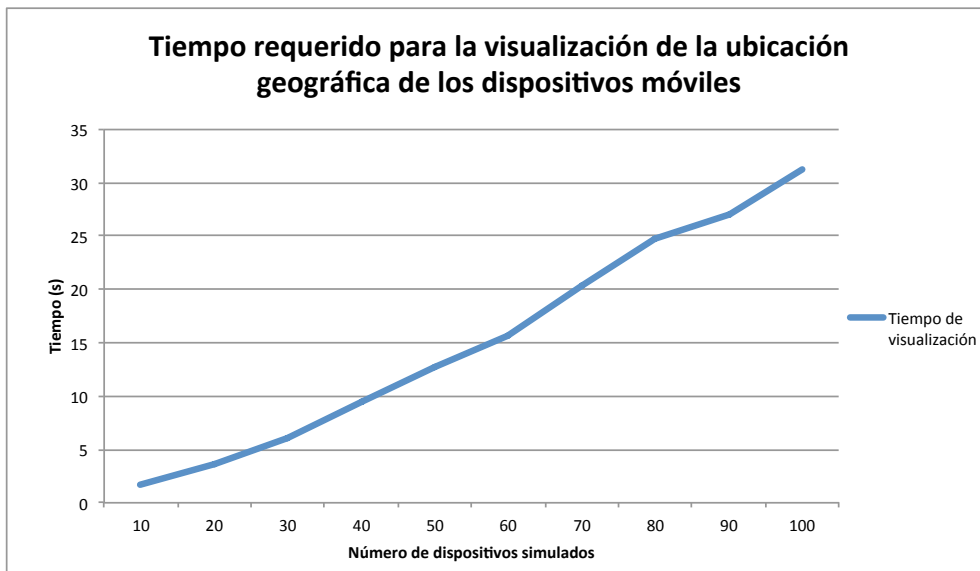


Figura 7.8: Gráfica en la que se muestra el tiempo que le toma a la aplicación visualizar la ubicación y trayectoria de los dispositivos móviles en movimiento.

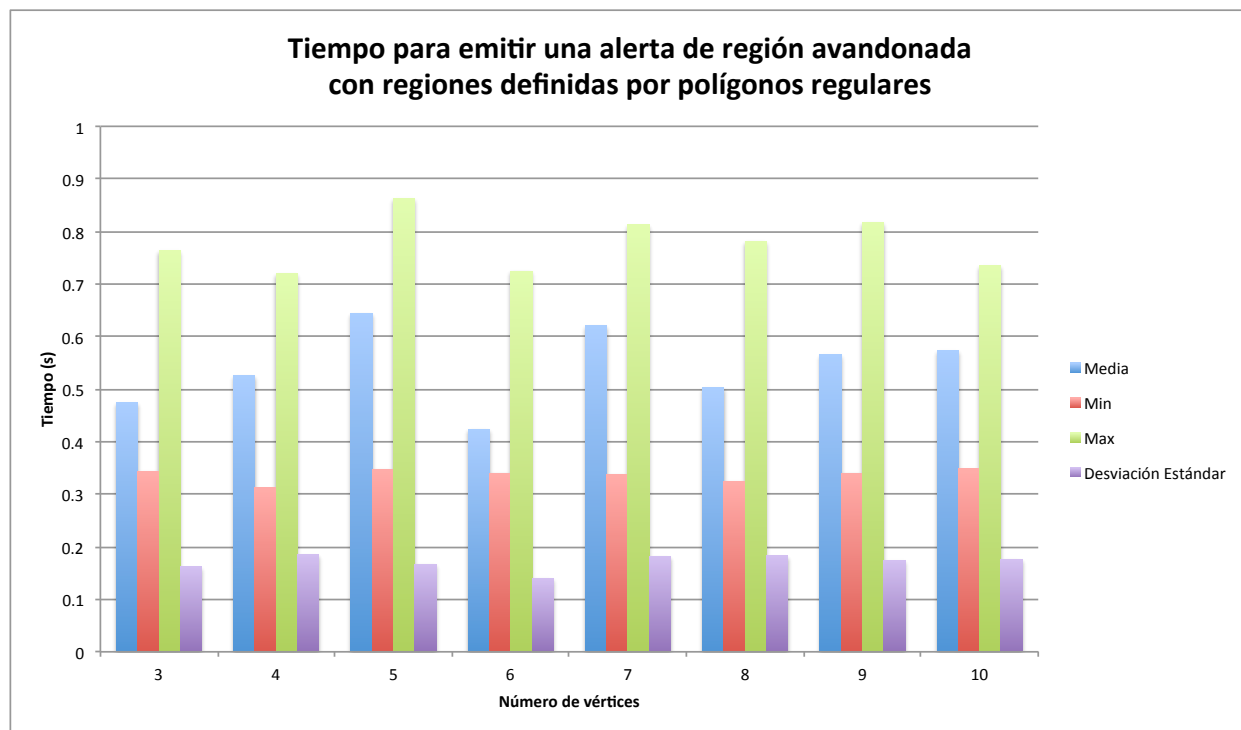


Figura 7.9: Gráfico en el que se muestra el tiempo requerido para emitir una alerta de región abandonada. Las regiones de prueba son polígonos regulares, de pequeña cobertura y con un número de vértices  $< 10$ .

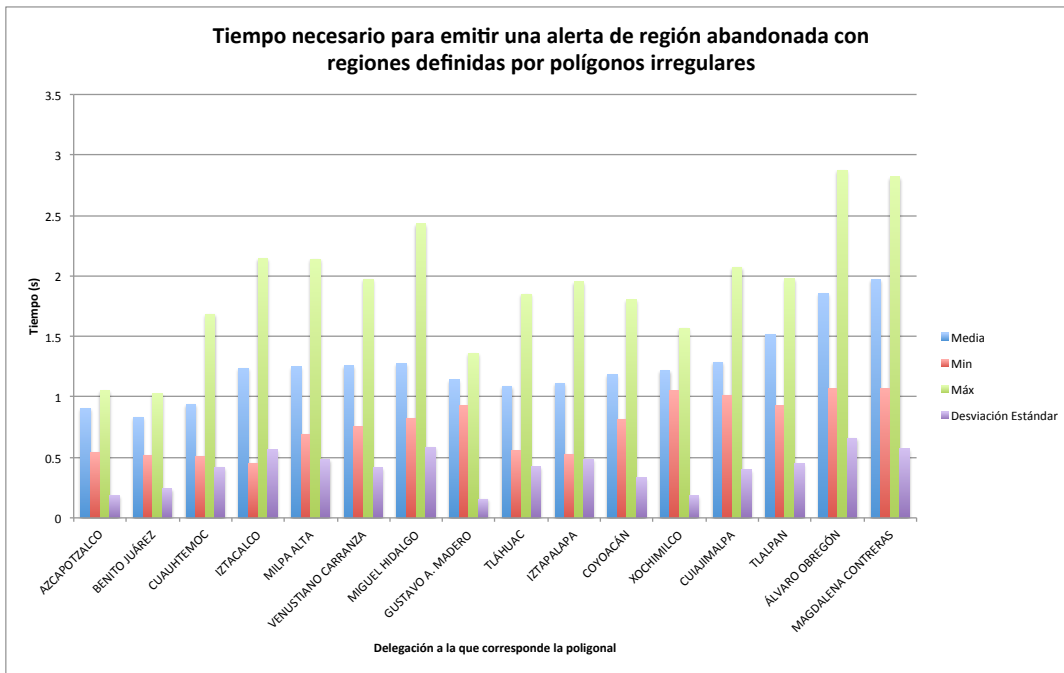


Figura 7.10: Gráfico en el que se muestra el tiempo requerido para emitir una alerta de región abandonada. Las regiones de prueba son polígonos completamente irregulares.

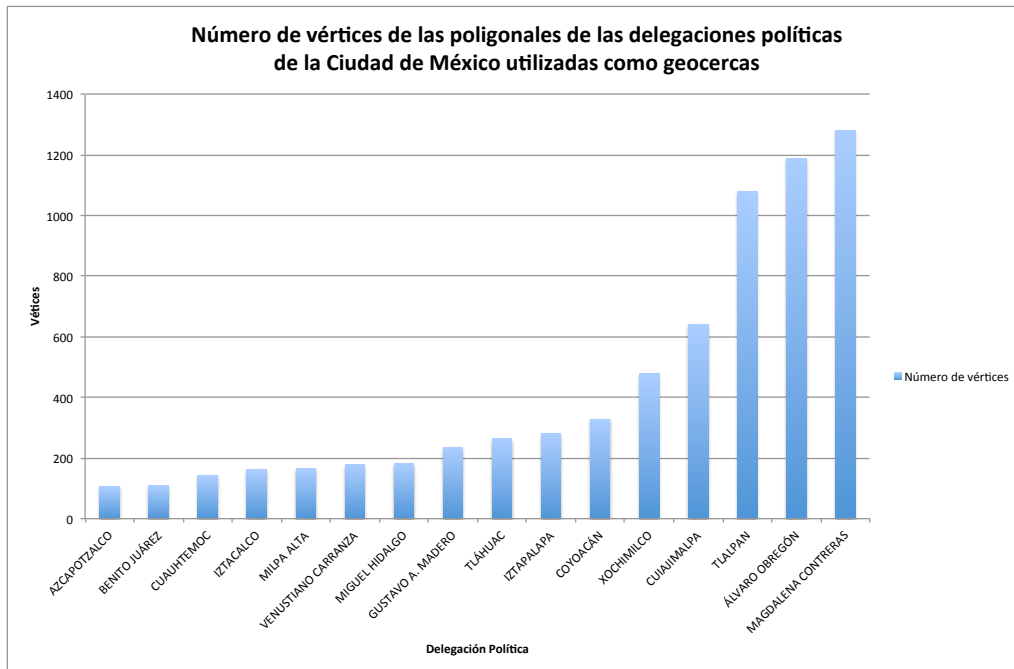


Figura 7.11: Número de vértices de las geocercas adoptadas para la prueba con polígonos irregulares.



# Capítulo 8

## Conclusión y trabajo futuro

En 2007 el número de usuarios de telefonía celular ascendía en ese momento los 3 mil millones 300 mil usuarios en todo el mundo, lo que representaba aproximadamente la mitad de la población en el mundo [28], mientras que el año 2009 la cifra de suscriptores había llegado a los 4 mil millones. El auge de la telefonía y las redes de comunicación han traído consigo la necesidad de portar con un mayor número de aplicaciones en el teléfono celular y, además, cada vez más parecidas a las aplicaciones de escritorio. Debido a que el teléfono celular, por un lado, se ha convertido en un dispositivo de uso diario y forma parte integral del estilo de vida de la sociedad moderna, al grado que su presencia en el entorno ya no es tan notoria como su ausencia y, por otro, a que los fabricantes cada vez los producen con mayores capacidades y aditamentos.

Hoy en día, los procesadores de los teléfonos celulares (teléfonos inteligentes, como se les ha catalogado) son tan completos que incluso las computadoras portátiles con poder de cómputo reducido pueden utilizarlos. Modernas plataformas para dispositivos móviles, como iPhone y Android, incluyen una amplia gama de librerías útiles e *Interfaces de Programación de Aplicaciones* (API, *Application Programming Interface*, por sus siglas en inglés) para desarrolladores. Las bibliotecas permiten acceder a los sensores internos, como acelerómetro o GPS. Los teléfonos celulares son un ejemplo de *Cómputo Móvil*, pueden realizar tareas de cómputo distribuido y ejecutar aplicaciones basadas en su ubicación.

Con el fin de proporcionar una herramienta informática que permita localizar y conocer el desplazamiento geográfico de una persona (portador de un teléfono celular), en el presente trabajo de tesis se propuso diseñar e implementar un mecanismo que permita la localización y seguimiento de dispositivos móviles; motivado por la creciente popularidad del cómputo móvil y por el campo de oportunidad que generan las redes de banda ancha móvil.

### 8.1. Conclusiones

Para concluir con el trabajo de tesis desarrollado, en esta sección se exponen de manera resumida los resultados obtenidos a partir de los cuales es posible determinar los objetivos

cumplidos, las ventajas del sistema desarrollado, así como sus desventajas y limitantes. De los objetivos propuestos al inicio del trabajo, se puede concluir que se cumplió con todos ellos debido a que:

- Se diseñó y adoptó una arquitectura de comunicación para el sistema basada en el modelo cliente servidor, para operar sobre la internet y las redes de telefonía celular. Lo anterior con la finalidad de proveer un área amplia de cobertura para localización y seguimiento con el sistema.
- Se diseñó la arquitectura para los clientes móviles, con base en la cual posteriormente se realizaría una implementación sobre las plataformas Android y iPhone de dispositivos móviles.
- Para cumplir con el requerimiento no funcional de independencia de la plataforma, se diseñó un servidor de aplicación basado en web. Además, el diseño de la misma se realizó con base en patrones de diseño para proveer de buenas características de software al sistema que se desarrollaría.
- Se diseñó un mecanismo de comunicación que permite el intercambio de información entre el servidor de la aplicación y los clientes móviles con base en el mecanismo *polling* sobre *HTTPS*, con las limitantes que esto genera.
- Se diseñó un mecanismo de despliegue de información geográfica basado en *Google Maps* y *Google Directions*, con la finalidad de abstraer en él la funcionalidad común requerida en las aplicaciones de localización y seguimiento.
- Una vez finalizado el diseño de las arquitecturas y módulos mencionados en los puntos anteriores, se realizó la implementación correspondiente de cada módulo del sistema basado en buenas prácticas de desarrollo, tomando como base una plataforma que provea al sistema de flexibilidad a cambios.

Una vez que se finalizó con las actividades de diseño e implementación, fue necesario diseñar una etapa de pruebas a las que se sometería el sistema desarrollado. Dichas pruebas tuvieron la finalidad de evaluar el rendimiento de la funcionalidad implementada. Los principales resultados obtenidos son:

- Se midió la capacidad del sistema para proveer la funcionalidad de manera concurrente hasta 100 dispositivos móviles simulados. De la prueba realizada se concluyó que el tiempo necesario para atender a los 100 dispositivos móviles, 250 *ms*, es un tiempo bastante bueno considerando el equipo de cómputo sobre el cual se instaló la aplicación. Es importante mencionar que el tiempo de localización también incluye el tiempo necesario para someter la ubicación geográfica al análisis de emisión de alerta de desplazamiento nulo.
- Se realizó una prueba para medir el tiempo de respuesta del servidor hacia los clientes móviles. La finalidad de la prueba fue medir el tiempo requerido para visualizar la

ubicación geográfica de un conjunto de unidades móviles en localización. De la prueba se obtuvo que el tiempo necesario para visualizar la ubicación geográfica de las 100 unidades en desplazamiento es de 33 s en promedio.

Una vez que el sistema fue evaluado y el análisis de los resultados fueron aceptables, se diseñó un plan de pruebas para medir los tiempos de respuesta del AGPS del Android HTC Hero y iPhone para los que se implementó el cliente móvil:

- Los teléfonos celulares se sometieron a pruebas para medir el rendimiento del AGPS en su modo de arranque en frío. La finalidad de medir el tiempo de respuesta en ese modo de arranque se realizó por que es el modo en el que se obtiene el mayor tiempo de respuesta del AGPS, que en los dispositivos HTC Hero y iPhone el promedio es de 57.55 y 51.53 s respectivamente. Dicho tiempo de respuesta es bastante aceptable ya que en lo sucesivo el tiempo de respuesta mejora disminuyendo el retardo.
- Por otro lado, los dispositivos fueron sometidos a pruebas para medir la precisión del AGPS. Se sometieron a dos escenarios distintos: un escenario en el que las condiciones fuesen favorables a la tecnología AGPS y otro en el que contexto creara un escenario desfavorable. En el primer escenario la precisión promedio del HTC Hero y el iPhone es de 13.8 y 13.14 m respectivamente. De igual manera en el segundo escenario el promedio es de 26.7 y 24.85 m.

De la ejecución de las pruebas a las que se sometieron los dispositivos se concluye que son una excelente herramienta para localización y seguimiento ya que son dispositivos que día a día están más presentes en las sociedades modernas. Además, los tiempos de respuesta y precisión son bastante aceptables, considerando que tales dispositivos no fueron construidos con la finalidad de proveer un mecanismo de posicionamiento de alta precisión.

Para finalizar, después de analizar los resultados obtenidos, los requerimientos funcionales y no funcionales que satisface el sistema, el sistema implementado tiene las siguientes:

- Ventajas:
  1. Provee un mecanismo preciso y eficiente para localización utilizando dispositivos de uso cotidianos, como son los teléfonos celulares.
  2. La cobertura para localización y seguimiento es amplia ya que, por un lado el sistema opera sobre la Internet y un usuario podría verificar la ubicación del portador de un dispositivo móvil solo con tener una computadora con acceso a la red y un navegador de internet. Por otro lado, las redes de banda ancha móvil han cubierto prácticamente todo el globo terrestre, de tal forma que se podría localizar un móvil ubicado en cualquier parte del mundo desde cualquier sitio.
  3. Permite conocer, de forma rápida, ubicaciones del pasado de los dispositivos móviles a través del mecanismo de generación de historiales de posicionamiento.
  4. La metodología y plataformas de desarrollo adoptadas favorecen el crecimiento del sistema y además posibilita futuros cambios, lo que permite continuar con el trabajo desarrollado.

- Desventajas:
  1. La dependencia a una conectividad a Internet es alta y si por algún motivo no se tiene acceso a ella (una desastre natural que corte las telecomunicaciones, por ejemplo) el sistema no podría utilizarse.
  2. Se tiene total dependencia del servicio de mapas de *Google*, debido a que el despliegue de la información de ubicación de los dispositivos móviles y se realiza sobre un mapa de Google y además se utiliza el servicio de *Google Directions* para obtener la dirección de la coordenada en la que se encuentra el móvil. Por lo anterior, algún cambio en los servicios mencionados podría afectar de manera negativa la funcionalidad implementada.
  3. Aunque hoy día la cobertura de las redes de telefonía celular es casi global, aun existen lugares sin cobertura, sobre todo en los países en vías de desarrollo y los llamados países de tercer mundo. Aunado a ello, en ese tipo de países las tarifas de la telefonía celular son muy altos, los que incrementa los costos que el uso del sistema generaría.
  4. Debido a que el desarrollo para los clientes móviles son aplicaciones nativas, para los dispositivos Android y iPhone, éstas aplicaciones no se podría utilizar en alguna otra plataforma de teléfono móvil, por ejemplo: *Blak Berry*, *Symbian*, por mencionar algunas.
  5. Debido a que el sistema opera sobre el protocolo HTTPS y a que se implementó un mecanismo de *polling* del cliente móvil hacia la aplicación web, no es posible establecer comunicación del servidor de la aplicación hacia los dispositivos móviles. Lo anterior ocasiona un retardo cuando es necesario realizar una notificación del servidor al móvil. Dicho retardo, en el peor de los casos es tan grande como el intervalo de tiempo con base en el cual se realiza el polling.

## 8.2. Trabajo futuro

En la presente sección se expone el trabajo futuro que se propone como complemento o mejora a lo realizado en el presente trabajo de tesis. El trabajo futuro propuesto es el siguiente:

- Implementar la funcionalidad del cliente móvil en otras plataformas de dispositivos, por ejemplo: *Blak Berry*, *Symbian*, *Windows Mobile*, *MeeGo*, etc.
- Implementar el mecanismo de localización utilizando no solo AGPS, sino también redes WiFi o por medio de *Cell-Id*, en los dispositivos móviles que lo soporten, por ejemplo los de la plataforma Android.
- Implementar un servicio web de mapas que contenga las calles, avenidas y carreteras de, al menos, todo el país. Con ello se dotaría al sistema de independencia hacia los servicios de *Google* utilizados en el presente trabajo.



- 
- Diseñar un mecanismo de comunicación entre el servidor y los clientes móviles basados en *push*, lo que permitiría enviar información en el momento en que se requiera del servidor de la aplicación hacia los clientes móviles, sin el retraso que ocasiona el mecanismo de *polling* implementado sobre HTTPS, del cliente móvil hacia el servidor, en este trabajo de tesis.
  - Implementar funcionalidad en algún dominio en específico: localización vehicular, transporte público, empresas de envíos, reparto de algún tipo de mercancía, seguimiento de personas con algún interés en particular, etc. Por ejemplo, aplicar el sistema al seguimiento vehicular con un mecanismo de planeación y seguimiento de rutas basándose en información de tráfico vial, costos de las carreteras, condiciones climáticas del ambiente, etc.
  - Diseñar e implementar un mecanismo, que además de la transferencia sobre el protocolo HTTPS, dote al sistema de mayor seguridad para la transferencia de información sobre la Internet.



# Anexo A

## Análisis de Requerimientos

El presente capítulo tiene como finalidad, presentar los requerimientos del LoSeGui, en distintos niveles de especificación: básicos, funcionales y no funcionales, hasta finalmente llegar a la especificación de los requerimientos funcionales del sistema. Está estructurado en las siguientes tres secciones:

- A.1: En esta sección encontrará los requerimientos básicos, extraídos de un análisis general del problema. Posteriormente se presentan los requerimientos funcionales.
- A.2: Aquí encontrará la especificación de los requerimientos funcionales presentados en el la sección A.1.
- A.3: Ésta sección contiene el diccionario de datos, el cual fue construido con la finalidad de guiar el diseño e implementación de la base de datos que satisfacer los requerimientos del LoSeGui.

### A.1. Definición de requerimientos

En la presente sección se abordan los requerimientos del LoSeGui desde distintos niveles de especificación y clasificación, aquí los presentamos como:

- **Requerimientos Básicos o de Usuario:** Son declaraciones en lenguaje natural y en diagramas de: las necesidades del usuario, de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.
- **Requerimientos del Sistema:** Establecen con detalle los servicios y restricciones del sistema, los cuales se clasifican en *funcionales y no funcionales*.

#### A.1.1. Definición de requerimientos básicos del LoSeGui

La definición de los requerimientos básicos muestra las necesidades identificadas en los usuarios del LoSeGui. Durante la etapa de obtención de requerimientos, éstos se recabaron a partir de la descripción inicial del problema proporcionada por el cliente y las entrevistas

realizadas a los usuarios potenciales, en ocasiones denominados *stakeholders*.

En la tabla A.1 se citan los requerimientos básicos del LoSeGui, en la cual se usa un identificador para cada requerimiento, el cual está formado por dos letras (RB, el cual es un acrónimo de Requerimiento Básico) y un número para identificar cada requerimiento.

<b>Id</b>	<b>Descripción</b>	<b>Origen</b>
RB1	El sistema permitirá realizar la administración de los dispositivos móviles para localización y seguimiento	Definición del sistema
RB2	El sistema permitirá elegir los dispositivos móviles que se desee monitorear.	Definición del sistema
RB3	El sistema permitirá conocer la ubicación y trayectoria de los dispositivos móviles	Definición del sistema
RB4	El sistema permitirá el acceso desde una interfaz gráfica disponible desde la Internet.	Definición del sistema
RB5	El sistema deberá manejar distintos perfiles de usuario.	Definición del sistema
RB6	El sistema permitirá la administración de usuarios.	Definición del sistema
RB7	El sistema permitirá el acceso únicamente a usuarios debidamente identificados.	Definición del sistema
RB8	El sistema permitirá a los usuarios generar mapas temáticos configurables.	Definición del sistema
RB9	El sistema emitirá alertas deberá emitir alertas de región abandonada y desplazamiento nulo.	Definición del sistema
RB10	El sistema permitirá a los usuarios configurar el intervalo de tiempo de localización de los dispositivos móviles.	Definición del sistema
RB11	La información respecto a dispositivos móviles, portadores de dispositivos móviles y demás usuarios del sistema se manejará sólo a través del sistema, garantizando el así la consistencia de los datos.	Definición del sistema

Tabla A.1: Definición de Requerimientos Básicos del LoSeGui.

### A.1.2. Definición de Requerimientos Funcionales del LoSeGui

En la definición de los requerimientos funcionales del LoSeGui se muestran a detalle los servicios que éste debe proporcionar a los distintos tipos de usuarios con los que interactúa.

Los requerimientos funcionales encontrados para el LoSeGui se obtuvieron a partir de la revisión, análisis y depuración de los requerimientos básicos encontrados en la definición del problema. El identificador de cada *Requerimiento Funcional* está compuesto por dos letras

(*RF*, el cual es un acrónimo de Requerimiento Funcional), seguido de un guión y un número que diferenciará cada requerimiento.

#### A.1.2.1. Control de acceso

En la tabla A.2 se muestran los requerimientos de control de acceso y los tipos de usuarios que interactuarán con el LoSeGui.

ID	Control de acceso del LoSeGui	Origen
RF-1	El sistema debe permitir el ingreso de los usuarios mediante un usuario y una contraseña.	RB7
RF-1.1	El sistema debe identificar el tipo de usuario que ingresa y brindarle servicios de acuerdo a su perfil. Un usuario puede ser un empleado adjunto al centro de mando ( <i>Supervisor</i> ) y jefe del centro de mando ( <i>Administrador</i> ).	RB5

Tabla A.2: RF: Tipos de usuarios del LoSeGui

#### A.1.2.2. Usuarios Administrador y Supervisor

En las tablas A.3 y A.4 se describen los servicios que el LoSeGui ofrece, en forma compartida, a los usuarios *Administrador* y *Supervisor*.

ID	Servicios para los usuarios Administrador y Supervisor	Origen
RF-2.1	El sistema debe permitir al usuario seleccionar, de los dispositivos móviles activos registrados en el sistema, los que desee mostrar su posición haciendo uso de un mapa georeferenciado.	RB2
RF-2.2	El sistema deberá permitir al usuario proporcionar los parámetros para la generación de los siguientes mapas temáticos:	RB8, RB2, RB3
RF-2.2.1	El sistema deberá generar mapas temáticos con base en información histórica de localización y desplazamiento en un intervalo de tiempo dado por el usuario.	RB2, RB3

Tabla A.3: RF: servicios para los usuarios *Administrador* y *Supervisor*

#### A.1.2.3. Requerimientos funcionales del usuario Administrador

En la tabla A.5 se describen los servicios que el LoSeGui ofrece, exclusivamente, al usuario *Administrador*.

ID	Servicios para los usuarios Administrador y Supervisor	Origen
RF-2.2.2	El sistema deberá generar mapas temáticos con base en información histórica de posicionamiento de los dispositivos móviles y su relación con puntos de interés. Los dispositivos y el punto de interés serán proporcionados por el usuario.	RB2, RB3
RF-2.3	El sistema deberá emitir las siguientes alertas observables por el usuario:	RB9, RB3
RF-2.3.1	El sistema deberá emitir una alerta cuando algún dispositivo móvil manifieste desplazamiento nulo durante un intervalo de tiempo, el cual fue configurado previamente en el sistema.	RB3
RF-2.3.2	El sistema deberá emitir una alerta cuando algún dispositivo móvil abandone la geocerca que le fue asociada previamente en el sistema	RB3
RF-2.4	El sistema permitirá al usuario asociar a cada dispositivo móvil una geocerca, con la finalidad de lanzar una alerta cuando éstas no se encuentren dentro de ella.	RB9
RF-2.5	El sistema deberá permitir al usuario realizar las siguientes tareas de administración de portadores de dispositivos móviles:	
RF-2.5.1	El sistema deberá permitir al usuario agregar portadores de dispositivos móviles.	RB3
RF-2.5.2	El sistema deberá permitir al usuario modificar la información de los portadores de los dispositivos móviles. Los portadores debieron ser registrados previamente en el sistema.	RB3
RF-2.5.3	El sistema deberá permitir al usuario eliminar portadores de dispositivos móviles. Los portadores debieron ser registrados previamente en el sistema.	RB3
RF-2.6	El sistema deberá permitir al usuario realizar las siguientes tareas de administración de geocercas:	
RF-2.6.1	El sistema permitirá al usuario agregar geocercas trazándolas sobre un mapa georeferenciado.	Propuesto
RF-2.6.2	El sistema permitirá al usuario modificar la información de las geocercas. Las geocercas debieron ser registradas previamente en el sistema	Propuesto
RF-2.6.3	El sistema permitirá al usuario eliminar regiones de recorrido. Las regiones debieron ser registradas previamente en el sistema	Propuesto

Tabla A.4: RF: servicios los usuarios *Administrador y Supervisor*

<b>ID</b>	<b>Servicios para el usuario Administrador</b>	<b>Origen</b>
RF-3.1	El sistema deberá permitir al usuario configurar las siguientes alarmas que emitirá el sistema:	RB9
RF-3.1.1	El sistema deberá permitir al usuario asignar o modificar el intervalo de tiempo que un dispositivo móvil puede permanecer estático para que se emita la alerta de movimiento nulo.	RB9
RF-3.2	El sistema deberá permitir al usuario configurar el intervalo de tiempo, por defecto, en el que el dispositivo móvil envía su ubicación geográfica.	RB10
RF-3.3	El sistema deberá permitir al usuario realizar las siguientes tareas de administración de usuarios:	
RF-3.3.1	El sistema deberá permitir al usuario agregar usuarios: administradores y supervisores.	RB3
RF-3.3.2	El sistema deberá permitir al usuario modificar la información de usuarios: administradores y supervisores. Los usuarios debieron ser registrados previamente en el sistema.	RB3
RF-3.3.3	El sistema deberá permitir al usuario eliminar usuarios del sistema: administradores y supervisores. Los usuarios debieron ser registrados previamente en el sistema.	RB3
RF-3.4	El sistema deberá permitir al usuario realizar las siguientes tareas de administración de dispositivos móviles:	
RF-3.4.1	El sistema deberá permitir al usuario agregar dispositivos móviles.	RB3
RF-3.4.2	El sistema deberá permitir al usuario modificar la información de los dispositivos móviles. Los dispositivos debieron ser registradas previamente en el sistema.	RB3
RF-3.4.3	El sistema deberá permitir al usuario eliminar dispositivos móviles. los dispositivos móviles debieron ser registradas previamente en el sistema.	RB3

Tabla A.5: RF: servicios para el usuario Administrador

### A.1.3. Definición de requerimientos no funcionales del LoSeGui

En la tabla A.6 está contenida la definición de los requerimientos no funcionales del LoSeGui.

ID	Requerimientos no Funcionales	Origen
RNF1	El LoSeGui proporcionará a los distintos tipos de usuario una interfaz gráfica con alto grado de usabilidad.	RB3
RNF2	Garantizará la integridad y consistencia de la información de las bases de datos	Propuesto.
RNF3	Usará una base de datos para la administración de la información de los dispositivos móviles, localización y portadores de dispositivos móviles, además los usuarios del sistema.	RB11
RNF4	El tiempo que debe transcurrir para notificar de eventos al móvil debe ser el mínimo posible.	Propuesto

Tabla A.6: Requerimientos No Funcionales del LoSeGui

## A.2. Especificación de los requerimientos del LoSeGui

En esta sección se aborda la especificación de los requerimientos funcionales del LoSeGui, tratados en la sección A.1, con la finalidad de proporcionar una especificación funcional de los requerimientos y sea el punto de partida en el proceso de diseño del LoSeGui.

### A.2.1. Especificación de requerimientos funcionales del LoSeGui

Para especificar los requerimientos funcionales del LoSeGui se utiliza lenguaje natural estructurado, incluyendo la siguiente información:

1. Una descripción de la función o la entidad a especificar.
2. Descripción de las entradas, donde se originan y hacia donde van.
3. Descripción de las salidas.
4. Una precondición y una postcondición.



## A.2.2. Requerimientos de control de acceso

**LoSeGui:** RF1.1 Acceso al LoSeGui

**Descripción:** El sistema debe permitir el ingreso de los usuarios mediante un identificador de usuario y una contraseña. El identificador del usuario y la contraseña serán cadenas de texto formadas por cualquier caracter. Si el acceso al LoSeGui es satisfactorio brindará la funcionalidad de acuerdo al perfil del usuario.

**Entradas:** Identificador y contraseña del usuario.

**Origen:** El usuario que desea ingresar al sistema proporcionará los datos de entrada por medio del teclado de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Ventana principal del LoSeGui con la funcionalidad de acuerdo a su perfil o un mensaje de error con el texto: *El usuario y/o contraseña son incorrectos.*

**Precondiciones:** Para que el usuario pueda tener acceso debe estar registrado en la base de datos de usuarios del LoSeGui.

**Postcondiciones:** El sistema brindará acceso al usuario.

**Efectos laterales:** Ninguno.

### A.2.2.1. Requerimientos para el usuario Administrador y Supervisor

**LoSeGui:** RF2.1 Localizar y seguir dispositivos móviles deseados.

**Descripción:** El LoSeGui permitirá al usuario seleccionar, de los dispositivos activos de cada región, los que desee localizar y seguir. El usuario podrá localizar más de un dispositivos simultáneamente.

**Entradas:** Identificador de los dispositivos móviles a localizar.

**Origen:** El usuario elegirá, mediante el mouse de la computadora, los identificadores de los dispositivos móviles de cada región, que requiera localizar.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mapa con una representación gráfica de la ubicación de los dispositivos móviles.

**Precondiciones:** los dispositivos móviles deberán estar registradas y activas en el sistema.

**Postcondiciones:** El sistema deberá proporcionar la ubicación de los dispositivos móviles en el mapa georeferenciado.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF2.2.1 Generar mapas temáticos de localización.

**Descripción:** El sistema deberá generar mapas temáticos con base en información histórica de localización y trayectoria de los dispositivos móviles deseados para un determinado intervalo de tiempo. El mapa temático mostrará la información de posicionamiento y trayectoria de los dispositivos seleccionados.

**Entradas:** Identificador de los dispositivos móviles, fecha, hora de inicio y hora final para los cuales se generará el mapa temático.

**Origen:** El usuario proporcionará los datos de entrada mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mapa temático correspondiente a los parámetros de entrada o mensaje de error con el texto: "No existe información que cumpla con las restricciones proporcionadas en los parámetros de entrada".

**Precondiciones:** Debe existir información correspondiente a los datos de entrada en la base de datos del LoSeGui:

- La fecha para la que se generará el mapa temático debe ser menor o igual a la fecha actual del sistema ( $f_{mt} \leq f_a$ ).
- La hora inicial debe ser menor a la hora final ( $h_i < h_f$ ).

**Postcondiciones:** El sistema deberá mostrar el mapa temático generado.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF2.2.2 Generar mapas temáticos de dispositivos móviles con base a su cercanía con puntos de interés.

**Descripción:** El sistema deberá generar mapas temáticos con base en información histórica de posicionamiento de los dispositivos móviles y su relación con puntos de interés para un determinado intervalo de tiempo.

**Entradas:** Fecha, hora de inicio, hora final, radio de la región (circunferencia) y punto de interés.

**Origen:** El usuario proporcionará los datos de entrada mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mapa temático correspondiente a los parámetros de entrada o mensaje de error con el texto: *No existe información que cumpla con las restricciones proporcionadas en los parámetros de entrada.*

**Precondiciones:** Debe existir información correspondiente a los datos de entrada en la base de datos del LoSeGui.

**Postcondiciones:** El sistema deberá mostrar el mapa temático generado.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF2.3.1 Emitir alertas de dispositivos con desplazamiento nulo

**Descripción:** El sistema deberá emitir una alerta cuando algún dispositivo móvil manifieste movimiento nulo durante un intervalo de tiempo, el cual fue configurado previamente en el sistema. El dispositivo móvil que emita la alerta se marcará en el mapa con un símbolo especial y se agregará al panel de alertas.

**Entradas:** Identificador del dispositivo móvil que emite la alerta.

**Origen:** Dispositivo móvil

**Destino:** Servidor del LoSeGui y pantalla de localización de dispositivos móviles.

**Salidas:** El sistema deberá mostrar en el mapa base, con un símbolo especial, los dispositivos que emiten la alerta y la agregará en el panel de alertas.

**Precondiciones:** Algún dispositivo móvil debió manifestar movimiento nulo durante el intervalo de tiempo definido en la configuración de la alerta.

**Postcondiciones:** El LoSeGui deberá emitir la alerta.

**Efectos laterales:** Registrar la alerta en la base de datos del LoSeGui.

**LoSeGui:** RF2.3.2 Emitir alertas de dispositivos móviles fuera de su geocerca.

**Descripción:** El sistema deberá emitir una alerta cuando algún dispositivo móvil abandone la geocerca que le fue asignada, la cual fue configurada previamente en el sistema. El dispositivo móvil que dispare la alerta se marcará en el mapa con un símbolo especial y se agregará al panel de alertas.

**Entradas:** Identificador del dispositivo móvil que dispara la alerta.

**Origen:** Dispositivo móvil.

**Destino:** Pantalla de localización de dispositivos móviles.

**Salidas:** El sistema deberá mostrar en el mapa base, con un símbolo especial, la ubicación del dispositivo que dispara la alerta y la agregará al panel de alertas.

**Precondiciones:** El dispositivo móvil abandonó la geocerca que le fue asignada.

**Postcondiciones:** El LoSeGui deberá emitir la alerta indicada.

**Efectos laterales:** Registrar la alerta en la base de datos del LoSeGui.

**LoSeGui:** RF2.4 Asignar geocerca a los dispositivos móviles.

**Descripción:** El sistema deberá permitir al usuario asignar o modificar la geocerca a cada dispositivo móvil, con la finalidad de establecer regiones de recorrido y que el LoSeGui emita una alerta cuando los dispositivos las abandonen.

**Entradas:** Identificador del dispositivo y el nombre región que se le asignará.

**Origen:** El usuario asignará la región de recorrido de un dispositivo móvil seleccionando una región previamente registrada en el LoSeGui.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mensaje de asignación satisfactoria.

**Precondiciones:** La región proporcionada por el usuario debe ser un polígono.

**Postcondiciones:** El sistema asociará la región proporcionada al dispositivo móvil deseado.

**Efectos laterales:** El LoSeGui deberá emitir una alerta cuando un dispositivo móvil abandone la región que le fue asociada.

**LoSeGui:** RF2.5.1 Registrar Portador

**Descripción:** El LoSeGui deberá permitir al usuario registrar a los portadores de los dispositivos móviles.

**Entradas:** Información del portador a registrar.

**Origen:** El usuario proporcionará la información necesaria para el registro de los portadores mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El usuario deberá proporcionar toda la información solicitada por el LoSeGui para registrar al portador requerido.

El portador no debe estar registrado en el sistema.

**Postcondiciones:** El sistema deberá registrar al portador almacenando su información en la base de datos.

**Efectos laterales:** El sistema deberá permitir asociar un dispositivo móvil y un portador.

**LoSeGui:** RF2.5.2 Modificar Portador

**Descripción:** El LoSeGui permitirá al usuario modificar la información de los portadores registrados en el sistema. El usuario no podrá modificar el identificador del portador.

**Entradas:** Información del portador a modificar.

**Origen:** El usuario proporcionará la información necesaria para modificar los datos del portador mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El portador a modificar debe estar registrado en el sistema.

**Postcondiciones:** El LoSeGui debe modificar la información referente al portador seleccionado, es decir, debe actualizar la información del portador a la proporcionada por el usuario.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF2.5.3 Eliminar Portador.

**Descripción:** El LoSeGui permitirá al usuario eliminar portadores de dispositivos móviles.

**Entradas:** Identificador del portador a eliminar.

**Origen:** Elegirá el portador a eliminar mediante el mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mensaje de eliminación satisfactoria.

**Precondiciones:** El portador a eliminar debe estar registrado en el sistema y no debe estar asociado a un dispositivo móvil.

**Postcondiciones:** El LoSeGui debe eliminar al portador seleccionado, es decir, cambiar el estado del portador a *eliminado* en la base de datos.

**Efectos laterales:** El portador eliminado no podrá asociarse a un dispositivo móvil en el futuro.

**LoSeGui:** RF2.6.1 Registrar dispositivos móviles.

**Descripción:** El LoSeGui deberá permitir al usuario registrar los dispositivos móviles, las cuales son objeto de interés.

**Entradas:** Información del dispositivo móvil a registrar.

**Origen:** El usuario ingresará la información desde el dispositivo móvil.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El dispositivo móvil no debe estar registrado.

**Postcondiciones:** El sistema deberá registrar el dispositivo móvil almacenando su información en la base de datos del LoSeGui.

**Efectos laterales:** El dispositivo móvil registrado puede ser localizado y monitoreado en sus recorridos.

**LoSeGui:** RF2.6.2 Modificar dispositivos móviles.

**Descripción:** El LoSeGui deberá permitir al usuario modificar la información de los dispositivos móviles que estén registradas en el sistema. El identificador del dispositivo móvil no podrá ser modificado.

**Entradas:** Identificador del dispositivo e información a modificar.

**Origen:** El usuario proporcionará la información necesaria para la modificación de la información del dispositivo móvil mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El dispositivo móvil a modificar debe estar registrada en el sistema y no debe estar en desplazamiento.

**Postcondiciones:** El LoSeGui debe modificar la información del dispositivo móvil deseada, es decir, debe actualizar la información proporcionada por el usuario en la base de datos.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF2.6.3 Eliminar dispositivos móviles.

**Descripción:** El LoSeGui permitirá al usuario eliminar dispositivos móviles.

**Entradas:** Identificador del dispositivo móvil a eliminar.

**Origen:** El usuario seleccionará el identificador del dispositivo móvil mediante el mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mensaje de notificación de eliminación satisfactoria.

**Precondiciones:** El dispositivo móvil a eliminar debe estar registrado en el sistema.

**Postcondiciones:** El LoSeGui debe eliminar el dispositivo proporcionado, es decir, cambiar el estado del dispositivo a eliminado en la base de datos. Con lo anterior será posible generar historiales de posicionamiento del dispositivo aun después de ser eliminada,

**Efectos laterales:** El dispositivo móvil ya no podrá ser objeto de localización y seguimiento.

### A.2.3. Requerimientos para el usuario Administrador

**LoSeGui:** RF3.1.1 Asignar intervalo de tiempo de la alerta de desplazamiento nulo.

**Descripción:** El sistema permitirá al usuario asignar o modificar el intervalo de tiempo que debe transcurrir para que el LoSeGui emita una alerta cuando un dispositivo móvil manifieste desplazamiento nulo.

**Entradas:** Intervalo de tiempo que un dispositivo puede estar estática.

**Origen:** El usuario proporcionará el intervalo de tiempo a través del teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El intervalo de tiempo debe ser positivo ( $t_a > 0$ ).

**Postcondiciones:** El sistema configurará la alerta de desplazamiento nulo al intervalo de tiempo dado y lo almacenará en la base de datos.

**Efectos laterales:** El sistema emitirá una alerta de desplazamiento nulo si un dispositivo permanece estático ( $t_e$ ) un tiempo mayor o igual al establecido ( $t_e \geq t_a$ )

**LoSeGui:** RF3.2 Configurar intervalo de tiempo ( $t_p$ ) en que un dispositivo reporta su ubicación.

**Descripción:** El sistema permitirá al usuario configurar periodo de tiempo en que un dispositivo envía su posición.

**Entradas:** Intervalo de tiempo.

**Origen:** El usuario proporciona el intervalo de tiempo haciendo uso del teclado y mouse de la computadora.

**Destino:** Servidor del LoSeGui y dispositivo móvil.

**Salidas:** En caso de ocurrir algún error, mostrará el mensaje correspondiente.

**Precondiciones:** El intervalo de tiempo proporcionado por el usuario debe ser positivo ( $t_p > 0$ ).

**Postcondiciones:** El sistema notificará a cada dispositivo la reconfiguración del tiempo de envío de la posición.

**Efectos laterales:** El dispositivo móvil enviará su posición al servidor del LoSeGui de acuerdo al intervalo dado por el usuario.



**LoSeGui:** RF3.3.1 Registrar usuarios.

**Descripción:** El LoSeGui permitirá al usuario registrar otros usuarios, los cuales harán uso del mismo.

**Entradas:** Información del usuario a registrar.

**Origen:** El usuario proporcionará la información necesaria para el registro de los usuarios mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, mostrará el mensaje correspondiente.

**Precondiciones:** El usuario deberá proporcionar toda la información solicitada por el LoSeGui para el registro.

**Postcondiciones:** El sistema deberá registrar al usuario almacenando su información en la base de datos de usuarios.

**Efectos laterales:** El LoSeGui dará acceso al usuario registrado.

**LoSeGui:** RF3.3.2 Modificar usuarios.

**Descripción:** El LoSeGui permitirá al usuario modificar la información de usuarios que estén registrados en el sistema. El usuario no podrá modificar los identificadores de los usuarios registrados.

**Entradas:** Información del usuario a modificar.

**Origen:** El usuario proporcionará la información necesaria para la modificación del usuario mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El usuario a modificar debe estar registrado en el sistema.

**Postcondiciones:** El LoSeGui debe modificar al usuario proporcionado, es decir, debe actualizar la información referente al usuario en la base de datos.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF3.3.3 Eliminar usuarios.

**Descripción:** El LoSeGui permitirá al usuario eliminar usuarios previamente registrados en el sistema.

**Entradas:** Identificador del usuario a eliminar.

**Origen:** El usuario proporcionará el identificador de usuario para su eliminación mediante el mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mensaje de eliminación satisfactoria.

**Precondiciones:** El usuario a eliminar debe estar registrado en el sistema.

**Postcondiciones:** El LoSeGui debe cambiar el estado del usuario a eliminado.

**Efectos laterales:** El sistema no dará acceso al usuario eliminado.

**LoSeGui:** RF3.4.1 Registrar Regiones.

**Descripción:** El LoSeGui permitirá al usuario registrar las regiones de recorrido (geocercas), a las cuales se le van asociar dispositivos móviles.

**Entradas:** Información de la región a registrar.

**Origen:** El usuario proporcionará la información referente a la región y la trazará en un mapa.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** El usuario proporcionará toda la información para registrar la región.

**Postcondiciones:** El sistema deberá registrar la región almacenando su información en la base de datos del LoSeGui.

**Efectos laterales:** A la región se le podrán asociar dispositivos móviles

**LoSeGui:** RF3.4.2 Modificar Regiones.

**Descripción:** El LoSeGui permitirá al usuario modificar la información de las regiones registradas en el sistema.

**Entradas:** Información de la región a modificar.

**Origen:** El usuario proporcionará la información necesaria para la modificación de la información de la región mediante el teclado y mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** En caso de ocurrir algún error, muestra el mensaje correspondiente.

**Precondiciones:** La región a modificar debe estar registrada en el sistema.

**Postcondiciones:** El LoSeGui debe modificar la información de la región seleccionada, es decir, debe actualizar la información en la base de datos, proporcionada por el usuario.

**Efectos laterales:** Ninguno.

**LoSeGui:** RF3.4.3 Eliminar Regiones

**Descripción:** El LoSeGui permitirá al usuario eliminar regiones.

**Entradas:** Identificador de la región a eliminar.

**Origen:** El usuario seleccionará la región através el mouse de la computadora.

**Destino:** Servidor y base de datos del LoSeGui.

**Salidas:** Mensaje de eliminación satisfactoria

**Precondiciones:** La región a eliminar debe estar registrada en el sistema.

No debe tener asociada ningún dispositivo móvil activo.

**Postcondiciones:** El LoSeGui debe eliminar la región proporcionada, es decir, cambiar el estado de la región a eliminado en la base de datos.

**Efectos laterales:** Una vez eliminada ya no podrá asociarse a dispositivos móviles.

### A.3. Diccionario de Datos

Un *Diccionario de Datos* es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se utilizan en un sistema. En éste capítulo encontrará la definición de los datos que utilizará el LoSeGui.

La descripción de los datos aquí realizada, se puede observar finalmente en el diagrama entidad relación de la base de datos diseñada e implementada para el LoSeGui, ver Figura A.1.



### A.3.1. Rol Usuario

Esta entidad se creó para tener un registro y control de los tipos de usuario que manejará el sistema. Además, la funcionalidad que el sistema proporcione a un usuario dependerá de su rol.

**Caracter válido** = [A - Z | a - z]

**Rol Usuario** = idRol + Rol + Descripción.

- idRol = {0-9}<sup>11</sup>
- rol = {Caracter Válido}<sup>50</sup>
- descripción = {Caracter Válido}<sup>255</sup>

### A.3.2. Usuario

La entidad Usuario se refiere a las personas que podrán hacer uso del LoSeGui. La importancia de que se maneje esta entidad radica en que el sistema debe ser utilizado únicamente por el personal autorizado y para permitir el acceso a los mismos el sistema manejará un registro de los mismos.

**Caracter válido** = [A - Z | a - z]

**Usuario** = idUsuario + Nombre + Apellido Paterno + Apellido Materno + Contraseña + Estado.

- idUsuario = {Caracter Válido}<sup>15</sup>
- idRolUsuario = {Caracter Válido}<sup>15</sup>
- Nombre = {Caracter Válido}<sup>15</sup>
- Apellido Paterno = {Caracter Válido}<sup>15</sup>
- Apellido Materno = {Caracter Válido}<sup>15</sup>
- Contraseña = {Caracter Válido}<sup>15</sup>
- Estado = {0 | 1}
  - 0 = Eliminado
  - 1 = Registrado

### A.3.3. Dispositivos Móviles

Los Dispositivos Móviles son el punto medular del LoSeGui. los dispositivos móviles tienen los siguientes atributos:

**Caracter válido** = [A - Z | a - z][0 - 9]

**Dispositivo Móvil** = idDispositivo + idRegion + idEstado + Número telefónico

- idDispositivo = {Caracter Válido}<sup>15</sup>
- idRegion = {1 - 9}<sup>1</sup>+{0 - 9}<sup>9</sup>
- idEstado = {1 - 3}<sup>1</sup>
- Número telefónico = {Dígito}<sup>10</sup>

### A.3.3.1. Dispositivo en Movimiento

Los Dispositivos en Movimiento son los aquellos en los que un portador se identificó y éste se encuentra en movimiento. Los dispositivos en movimiento tendrán los siguientes atributos:

**Caracter válido** = [A - Z | a - z][0 - 9]

**Dispositivo en Movimiento** = idDispositivoMovimiento + idDispositivo + idPortador + idUsuario + idTiempoMonitoreo + fechaInicio + fechaFinal.

- idDispositivoMovimiento = {1 - 9}<sup>10</sup>
- idDispositivo = {Caracter Válido}<sup>10</sup>
- idPortador = {0 - 9}<sup>10</sup>
- idUsuario = {Caracter Válido}<sup>15</sup>
- idTiempoMonitoreo = {0 - 9}<sup>10</sup>
- fechaInicio = Año + / + Mes + / + Día + Hora+ : +Minuto+ : + Segundo
- fechaFinal = Año + / + Mes + / + Día + Hora+ : +Minuto+ : + Segundo

### A.3.3.2. Región

La entidad región (geocerca) se refiere al área dentro de la cual los dispositivos móviles deben desplazarse.

**Caracter válido** = [A - Z | a - z]

**Flotante** = [0 - 9]\* + . + [0 - 9]\*

**POINT** = [Flotante] + , + [Flotante]

**Región** = idRegion + Nombre + Descripción + Polígono

- idRegion = {Caracter Válido}<sup>15</sup>
- Nombre = {1 - 9}<sup>1</sup>+{0 - 9}<sup>9</sup>
- Descripción = {1 - 3}<sup>1</sup>
- Polígono = [POINT + ,] \* [POINT]

### A.3.3.3. Posición

La entidad Posición se refiere a las coordenadas geográficas reportadas por los dispositivos que se encuentra en movimiento.

**Caracter válido** = [A - Z | a - z]

**Flotante** = [0 - 9]\* + . + [0 - 9]\*

**POINT** = [Flotante] + , + [Flotante]

**Posición** = idPosicion + idDispositivo + idDispositivosMovimiento + idPortador + idUsuario + posicion + fecha

- idPosición = {0- 9}<sup>10</sup>
- idDispositivo = {Caracter Válido}<sup>10</sup>
- idDispositivoMovimiento = {1 - 9}<sup>10</sup>
- idPortador = {0- 9}<sup>10</sup>
- idUsuario = {Caracter Válido}<sup>15</sup>
- posicion = {0 - 9}<sup>10</sup>
- fecha= Año + / + Mes + / + Día + Hora+ : +Minuto+ : + Segundo

### A.3.3.4. Alerta

La entidad Alerta se refiere cuando un dispositivo a reportado una alerta, es decir, a abandonado su región asignada o no se ha desplazado durante un determinado tiempo.

**Caracter válido** = [A - Z | a - z]

**Alerta** = idAlerta + idDispositivo + idPosicion + fechaAlerta + idTipoAlerta + idTiempo

- idAlerta = {0- 9}<sup>10</sup>
- idDispositivo = {Caracter Válido}<sup>10</sup>
- idPosicion = {0 - 9}<sup>10</sup>
- fechaAlerta = Año + Mes + Día
- idTiempoAlerta = {0 - 9}<sup>11</sup>
- idTiempo = {0 - 9}<sup>11</sup>





# Anexo B

## Casos de Uso Principales

En el presente capítulo encontrará: la descripción, clases involucradas y trayectoria principal de los casos de uso más importantes en el Sistema de Localización y Seguimiento de Dispositivos Móviles.

Un caso de uso se define como el escenario en el cual se describe la interacción de un usuario con el sistema. En la fig. B.1 se muestra el diagrama con los casos de usos más importantes del Sistema de Localización y Seguimiento de Dispositivos Móviles y en ellos recae la mayor funcionalidad del mismo.

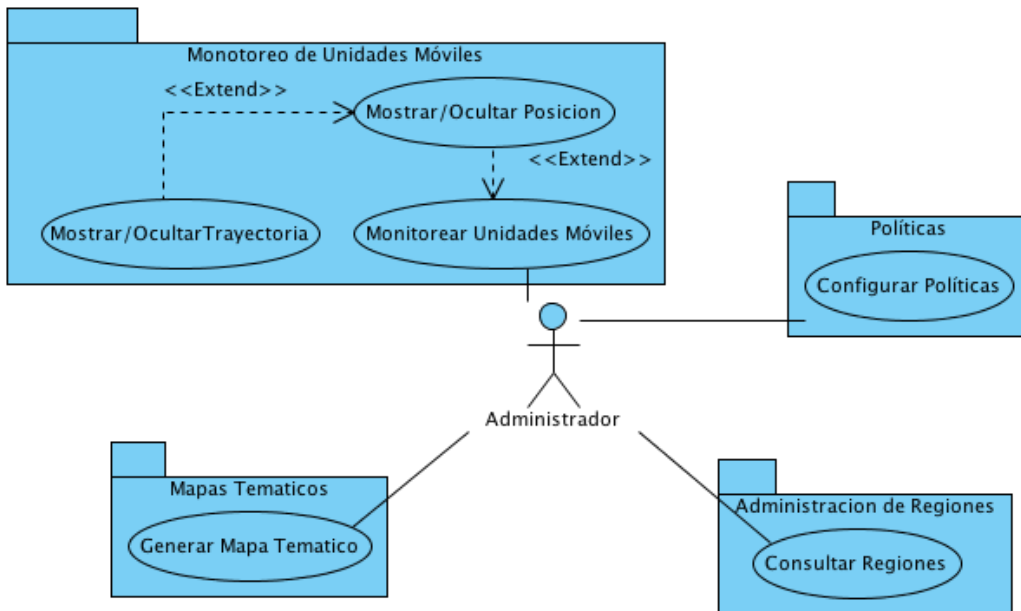


Figura B.1: Diagrama general de Casos de Uso



## B.1. UC1 Monitorear Dispositivo Móvil

### B.1.1. Descripción completa

El Sistema de Localización y Seguimiento de Dispositivos Móviles permite a los usuarios seleccionar, de los dispositivos móviles activos, las que desee monitorear. El usuario podrá monitorear más de un dispositivo simultáneamente.



### B.1.2. Atributos importantes

<b>Id:</b>	UC1
<b>Nombre:</b>	Monitorear Dispositivo Móvil
<b>Version:</b>	1.0
<b>Actores:</b>	Administrador, Supervisor
<b>Propósito:</b>	Monitorear dispositivos móviles.
<b>Resumen:</b>	Los usuarios pueden realizar la localización y seguimiento de la trayectoria de los dispositivos móviles que deseen.
<b>Entradas:</b>	Identificador de los dispositivos móviles a monitorear.
<b>Salidas:</b>	Mapa con una representación gráfica de la ubicación de los dispositivos móviles.
<b>Precondiciones:</b>	Los dispositivos móviles deberán estar registrados en el sistema y se deben de encontrar activos.
<b>Postcondiciones:</b>	El sistema obtiene la ubicación de los dispositivos móviles correspondientes a los identificadores proporcionados por el usuario y los representa gráficamente sobre un mapa.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RF3.1,RF3.2

### B.1.3. Trayectorias

#### Trayectoria principal: Principal

1. Hace clic sobre la pestaña Monitorear Dispositivos Móviles.
2. Busca las regiones registradas.
3. Busca los dispositivos móviles registrados y activos por cada región.
4. Muestra la pantalla Monitorear Dispositivos Móviles.
5. Muestra el identificador nombre asignado a los dispositivos móviles registradas agrupadas por región.

6.  Muestra un mapa georeferenciado utilizando el servicio de *Google*.
7.  Selecciona los dispositivos móviles a monitorear. [[PI UC2 Mostrar/Ocultar Posicion](#)]
8. Fin de trayectoria

B.1.4. Diagrama de Clases

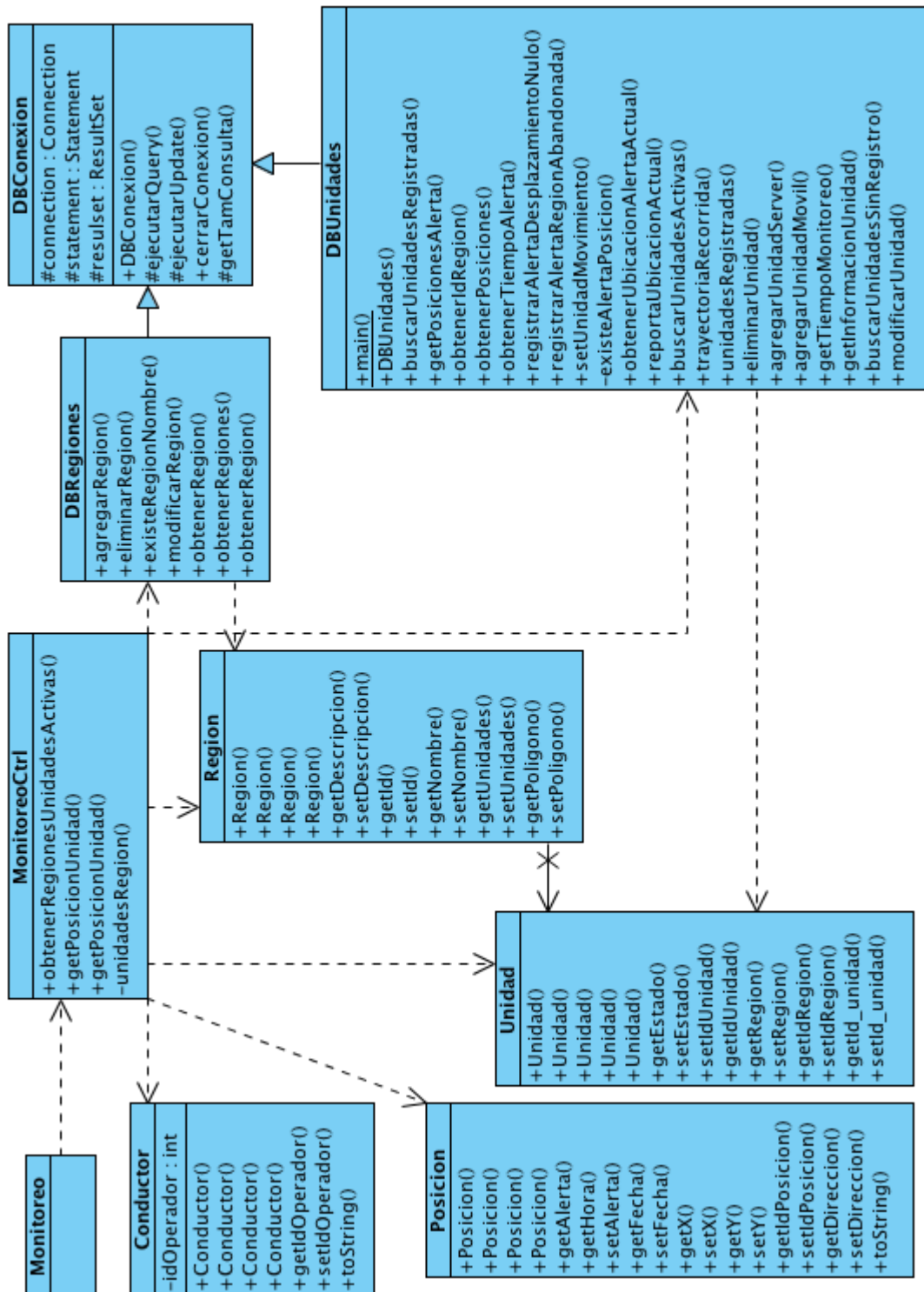


Figura B.2: Diagrama de Clases del CU Monitorear Dispositivo Móvil

### B.1.5. Diagrama de Secuencia

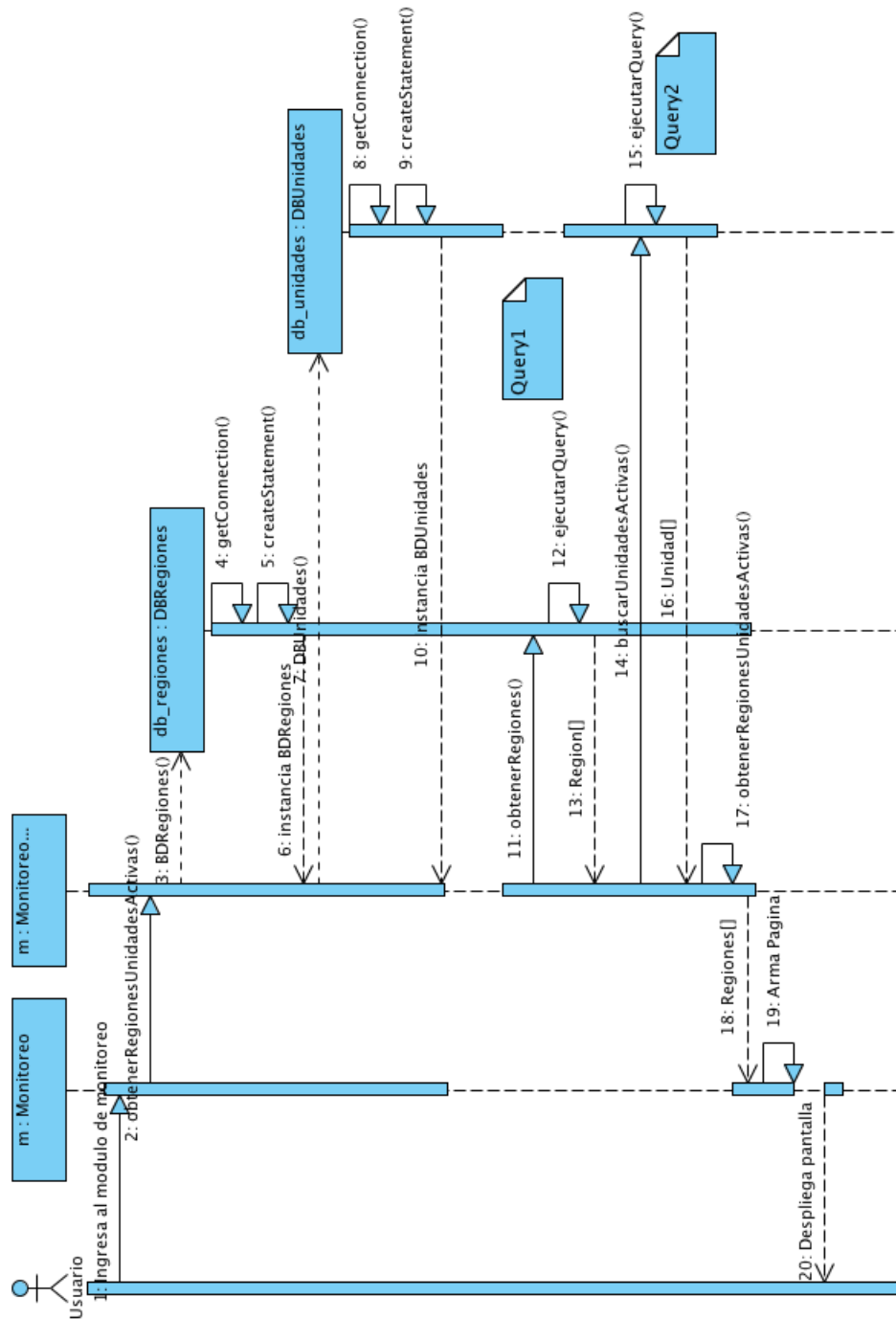


Figura B.3: Diagrama de Secuencia de CU Monitorear Dispositivo



## B.2. UC2 Mostrar/Ocultar Posición

### B.2.1. Descripción completa

El Sistema de Localización y Seguimiento de Dispositivos Móviles permite al usuario visualizar la posición de un dispositivo en un mapa que se encuentra en movimiento.

### B.2.2. Atributos importantes

<b>Id:</b>	UC2
<b>Nombre:</b>	Mostrar/Ocultar Posición
<b>Version:</b>	1.0
<b>Actores:</b>	Administrador, Supervisor
<b>Propósito:</b>	Visualizar la posición del dispositivo seleccionado en un mapa.
<b>Resumen:</b>	El usuario debe hacer clic en el <input type="checkbox"/> correspondiente al dispositivo del que se requiere visualizar su ubicación
<b>Entradas:</b>	Identificador del dispositivo del que se requiere conocer su posición
<b>Salidas:</b>	Representación gráfica de la posición del dispositivo.
<b>Precondiciones:</b>	El dispositivo móvil debe de estar activo, es decir, un portador debió iniciar sesión en el dispositivo.
<b>Postcondiciones:</b>	Al deshabilitar o habilitar el <input type="checkbox"/> de algún dispositivo, éste ofrece o bloquea el acceso al CU8 Mostar/Ocultar Trayectoria
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RF3.4.2





### B.2.3. Trayectorias

#### Trayectoria principal: Principal

1. Hacer clic sobre el  correspondiente al dispositivo del que desea conocer su posición [Trayectoria A] [[PE UC8 Mostrar/Ocultar Trayectoria](#)].
2. Inserta el dispositivo en la lista de dispositivos que se están monitoreando.
3. Busca la ultima posición reportada por el dispositivo.
4. Representa gráficamente la posición sobre el mapa.
5. Fin de trayectoria

#### Trayectoria alternativa A: Ocultar Posición

**Condición:** El Usuario no desea conocer la posición actual del dispositivo.

- A1**  Hace clic sobre  correspondiente al dispositivo, que se desea no seguir monitoreando.
- A2**  Elimina el dispositivo de la lista de dispositivos que se están monitoreando.
- A3**  Elimina la representación gráfica del dispositivo del mapa.
- A4** Fin de trayectoria

B.2.4. Diagrama de Clases

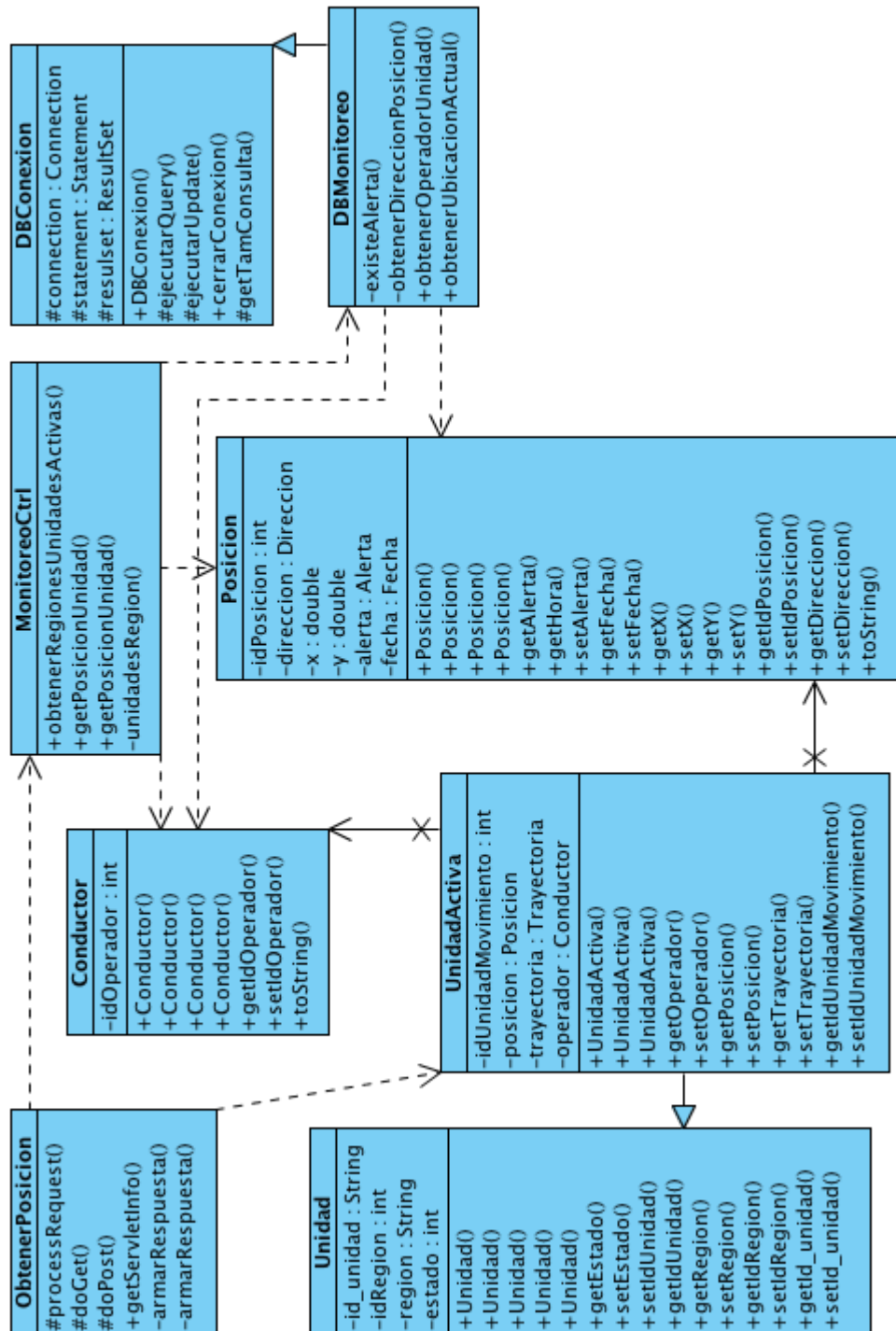


Figura B.4: Diagrama de Clases Mostrar/Ocultar Posición



### B.2.5. Diagrama de Secuencia

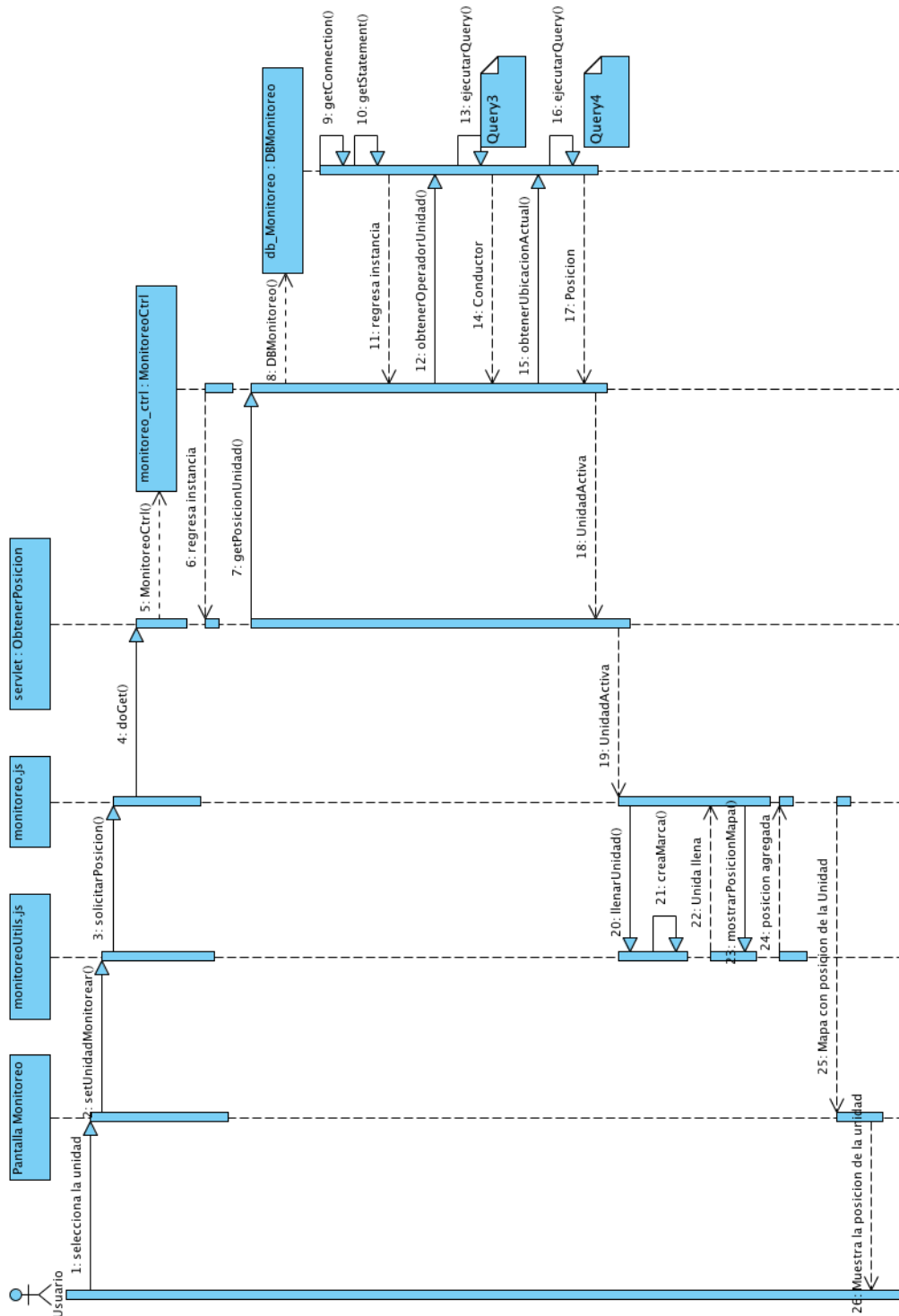


Figura B.5: Diagrama de Secuencia Mostrar/Ocultar Posición



## B.3. UC3 Mostrar/Ocultar Trayectoria

### B.3.1. Descripción completa

Debido a que el usuario necesita conocer las últimas posiciones por las cuales a pasado un dispositivo. El Sistema de Localización y Seguimiento de Dispositivos Móviles permite al usuario visualizar la trayectoria de un dispositivo móvil.

### B.3.2. Atributos importantes

<b>Id:</b>	UC3
<b>Nombre:</b>	Mostrar/Ocultar Trayectoria
<b>Version:</b>	1.0
<b>Actores:</b>	Administrador, Supervisor
<b>Propósito:</b>	Visualizar la trayectoria del dispositivo seleccionado en un mapa.
<b>Resumen:</b>	El usuario debe hacer clic en el <input type="checkbox"/> correspondiente al dispositivo del cual desea conocer su trayectoria.
<b>Entradas:</b>	Identificador del dispositivo del cual requiere conocer su trayectoria.
<b>Salidas:</b>	Representación gráfica de la trayectoria del dispositivo.
<b>Precondiciones:</b>	El dispositivo debe estar en movimiento, es decir, la casilla <input type="checkbox"/> correspondiente al dispositivo debe de estar seleccionada.
<b>Postcondiciones:</b>	Se debe mostrar la trayectoria del dispositivo seleccionado.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RF3.4.2




### B.3.3. Trayectorias

#### Trayectoria principal: Principal

1. Hacer clic sobre el  correspondiente al dispositivo [Trayectoria A].
2. Busca las 10 últimas posición reportadas por el dispositivo.
3. Relaciona las posiciones y las une mediante segmentos líneas.
4. Representa la representa gráficamente sobre el mapa.
5. Fin de trayectoria

#### Trayectoria alternativa A: Ocultar Trayectoria

**Condición:** El Usuario no desea conocer la trayectoria del dispositivo.

- 
- A1**  Hace clic sobre  correspondiente al dispositivo que ya no desea conocer su trayectoria.
- A2**  Elimina la representación gráfica de la trayectoria del dispositivo del mapa.
- A3** Fin de trayectoria

### B.3.4. Diagrama de Clases

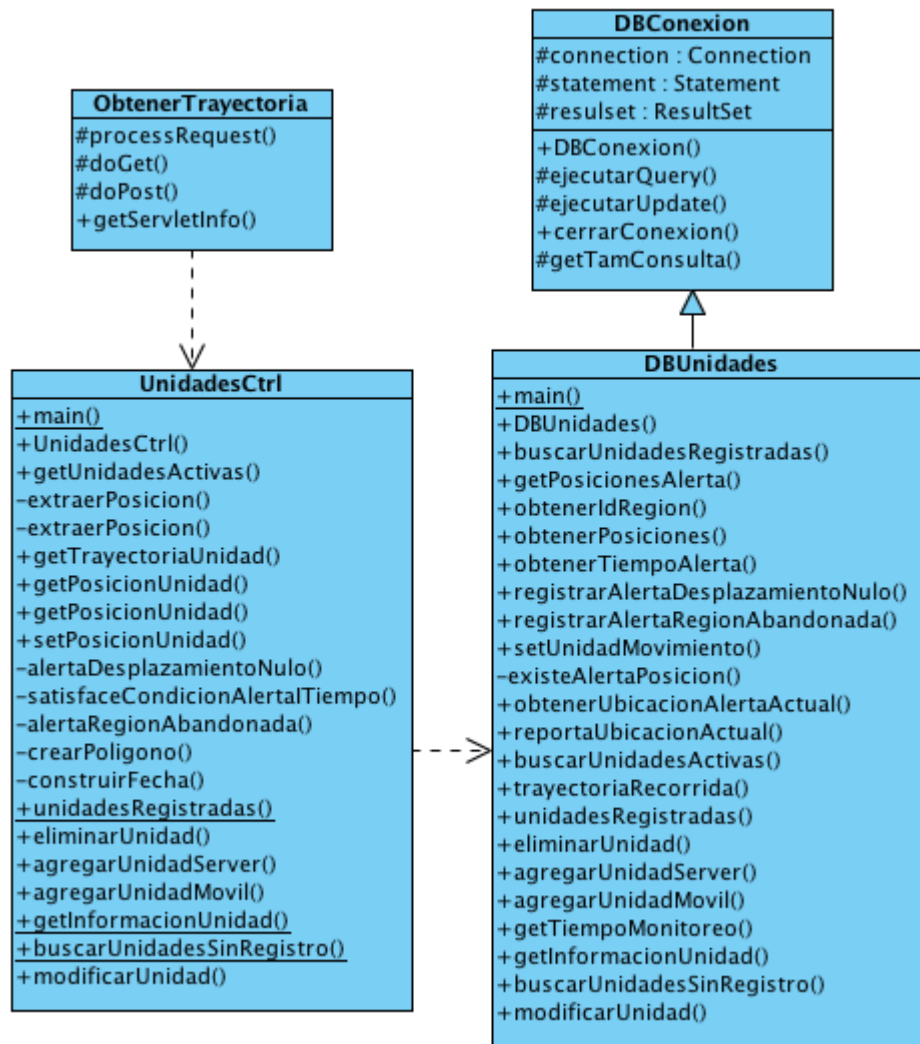


Figura B.6: Diagrama de Clases Mostrar/Ocultar Trayectoria

### B.3.5. Diagrama de Secuencia

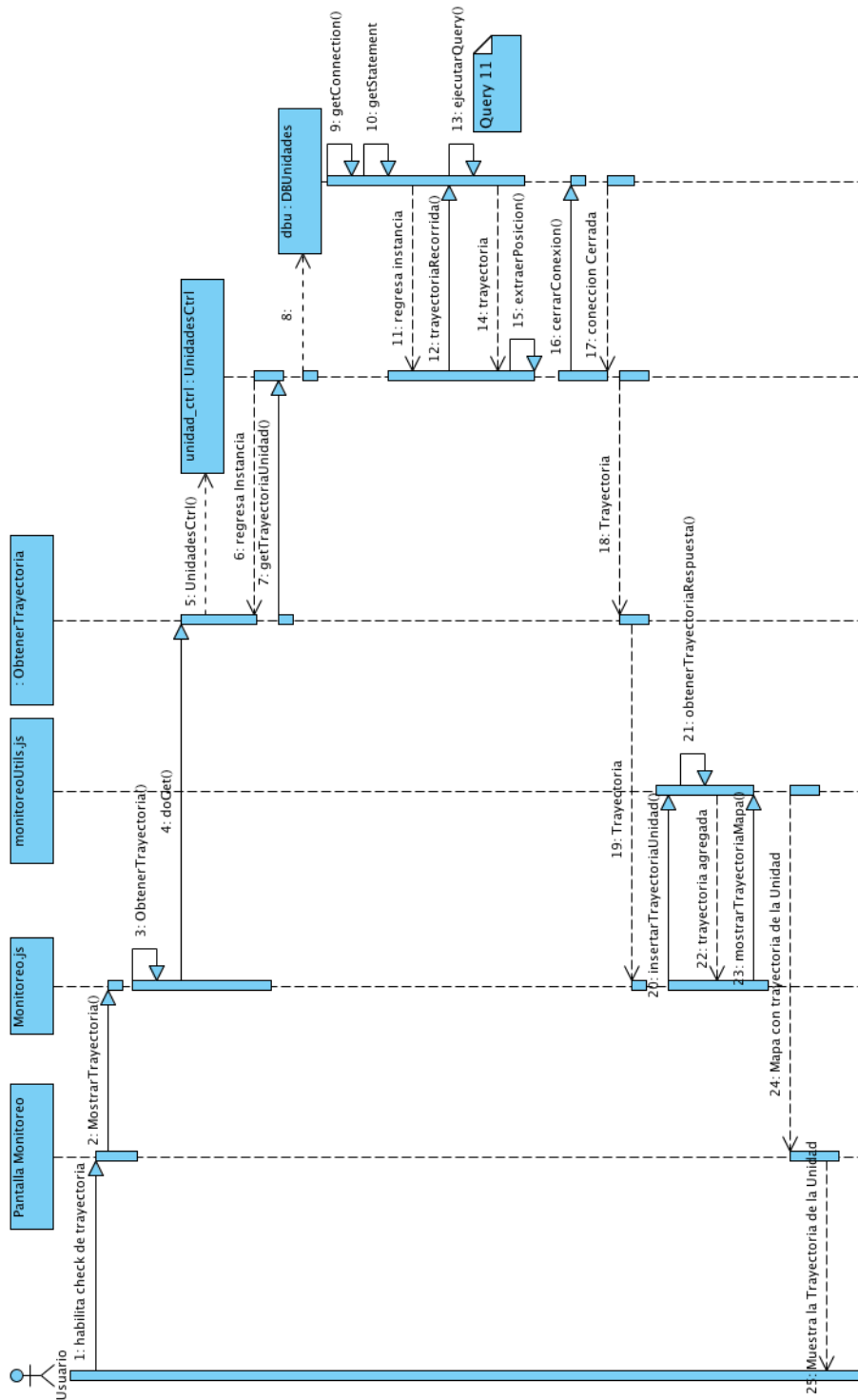


Figura B.7: Diagrama de Secuencia Mostrar/Ocultar Trayectoria



## B.4. UC4 Generar Mapa Temático Historial de Posicionamiento

### B.4.1. Descripción completa

Dado que en ocasiones los usuarios del Sistema de Localización y Seguimiento de Dispositivos Móviles no pueden monitorear todos los dispositivos que desean y percatarse de los detalles del recorrido de los mismos, el Sistema de Localización y Seguimiento de Dispositivos Móviles permite la generación de *mapas temáticos*. Los *mapas temáticos* se construirán con la finalidad de proporcionar a los usuarios un medio que les permita conocer y analizar, en caso de ser requerido, la posición y desplazamiento de los dispositivos móviles en un determinado intervalo de tiempo.









### B.4.2. Atributos importantes

<b>Id:</b>	UC4
<b>Nombre:</b>	Generar Mapa Temático Historial de Posicionamiento
<b>Version:</b>	1.0
<b>Actores:</b>	Administrador, Supervisor
<b>Propósito:</b>	Generar mapas temáticos.
<b>Resumen:</b>	El sistema genera mapas temáticos con base en la información histórica de localización y desplazamiento de los dispositivos móviles deseados en un determinado intervalo de tiempo.
<b>Entradas:</b>	Fecha, hora de inicio y hora final.
<b>Salidas:</b>	Mapa temático correspondiente a los parámetros de entrada o mensaje de error con el texto: <i>No hay informacion que satisfaga los parametros proporcionados</i> ".
<b>Precondiciones:</b>	Debe existir información correspondiente a los datos de entrada en la base de datos.
<b>Postcondiciones:</b>	El sistema deberá mostrar el mapa temático generado.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RU1

### B.4.3. Trayectorias


**Trayectoria principal:** Principal

1.  Hace clic sobre el botón Mapas Temático Historial de Posicionamiento.

2.  Muestra la pantalla de Mapas Temáticos Historial de Posicionamiento. [Trayectoria A]
3.  Ingresa la información solicitada, ver fig. B.4.4.
4.  Hace clic sobre el botón Generar Mapa.
5.  Verifica que todos los campo se hayan llenado. [Trayectoria B]
6.  Busca la información que satisface los parametros de entrada para generar el mapa temático. [Trayectoria C]
7.  Genera el mapa temático.
8.  Genera el reporte del mapa temático en forma tabular.
9.  Muestra el Mapa Temático y el reporte correspondiente.
10. Fin de trayectoria





### Trayectoria alternativa A: Cancelar

**Condición:** El usuario no desea generar un mapa.

- A1  Oprime el botón atrás del navegador o cambia de pestaña.
- A2 Fin de trayectoria



### Trayectoria alternativa B: Campos sin llenar

**Condición:** Al proporcionar la información obligatoria para generar el mapa se omitieron campos.

- B1  Detecta que al menos un campo obligatorio fue omitido.
- B2  Muestra un mensaje correspondiente al error.
- B3  Proporciona la información requerida.
- B4  El caso de uso continúa en la trayectoria principal en el paso 4.
- B5 Fin de trayectoria

### Trayectoria alternativa C: No Existe Información

**Condición:** Con los datos que se ingresaron es imposible generar un mapa temático

- C1  Muestra un mensaje de error: *No hay informacion que satisfaga los parametros proporcionados.*
- C2  Continúa el caso de uso en la trayectoria principal en el paso 3.
- C3 Fin de trayectoria

### B.4.4. Diagrama de Clases

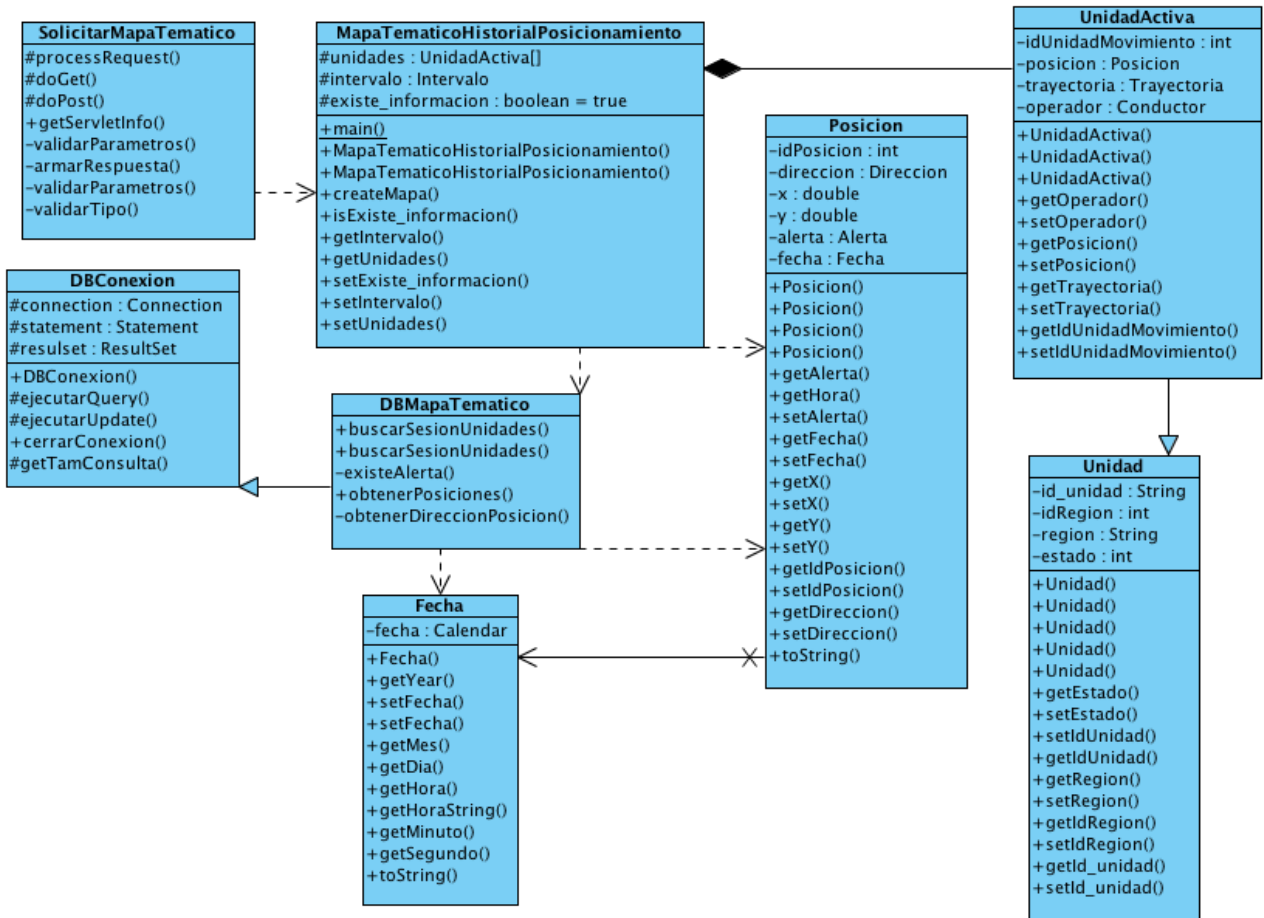


Figura B.8: Diagrama de Clases Mapa Temático Historial Posicionamiento



### B.4.5. Diagrama de Secuencia

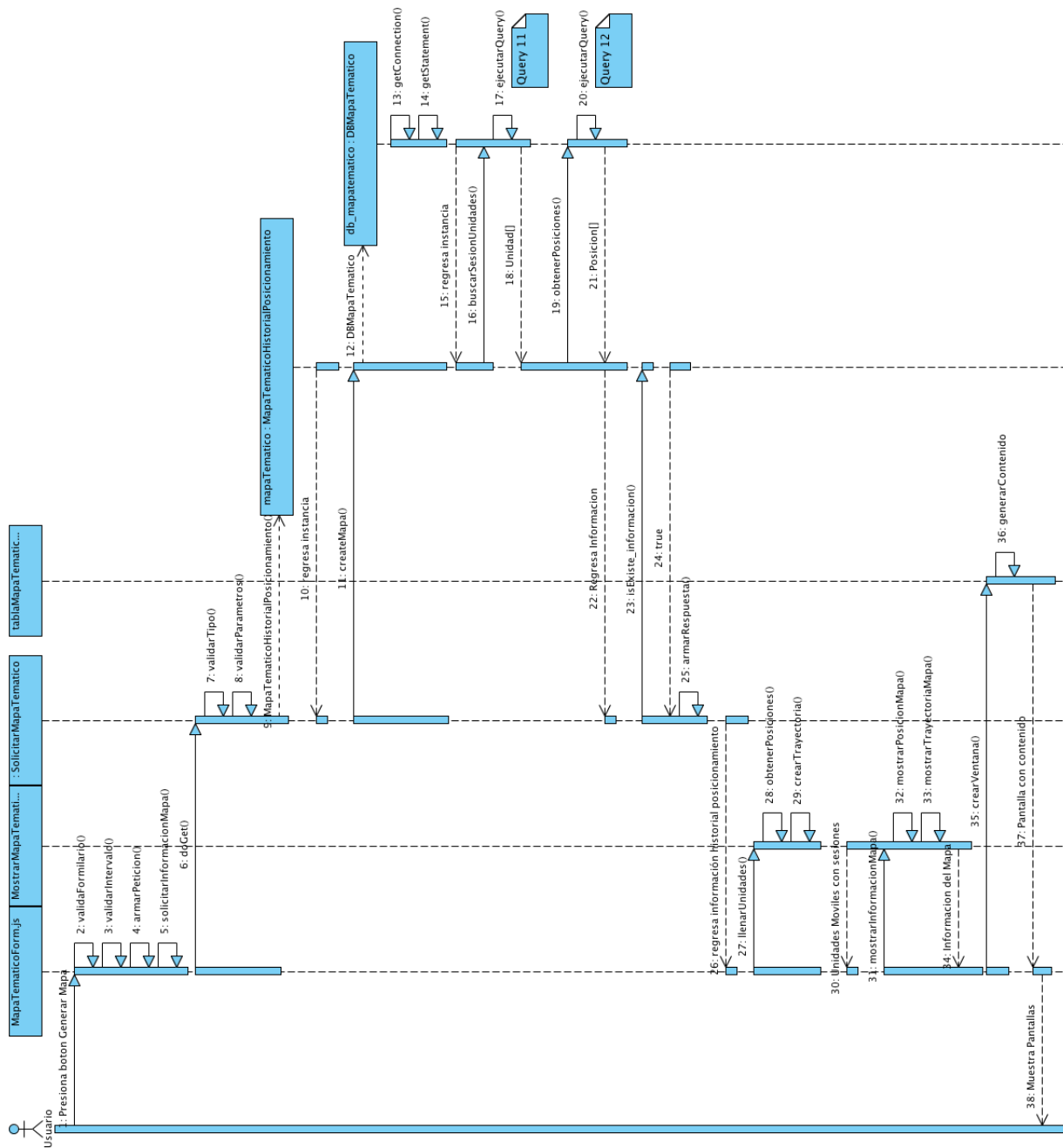


Figura B.9: Diagrama de Secuencia Mapa Temático Historial Posicionamiento



## B.5. UC5 Generar Mapa Temático Punto de Interés

### B.5.1. Descripción completa

Los *mapas temáticos respecto a puntos de interés* son mapas que se construyen con base en información histórica de posicionamiento de los dispositivos móviles y su relación de cercanía con puntos de interés, para una fecha e intervalo de tiempo específicos.




### B.5.2. Atributos importantes

<b>Id:</b>	UC5
<b>Nombre:</b>	Generar Mapa Temático Punto de Interés
<b>Version:</b>	1.0
<b>Actores:</b>	Administrador, Supervisor
<b>Propósito:</b>	Generar un mapa temático respecto a un punto de interés.
<b>Resumen:</b>	El sistema genera mapas temáticos con base en un punto de interés dado por el usuario, además de un intervalo de tiempo.
<b>Entradas:</b>	Punto de interés y radio
<b>Salidas:</b>	Mapa temático generado con la información proporcionada.
<b>Precondiciones:</b>	Para la generación del mapa temático debe existir información, histórica de posicionamiento, que satisfaga los parámetros proporcionados.
<b>Postcondiciones:</b>	El sistema debe generar y visualizar el mapa temático solicitados, junto con un reporte en forma tabular del mismo.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RF3.2.2

### B.5.3. Trayectorias


#### Trayectoria principal: Principal

1. Hace clic sobre el botón Mapas Temático Punto de Interés. [Trayectoria A]
2. Muestra la pantalla de Mapas Temáticos Punto de Interés.
3. Ingresa la información solicitada, ver fig. ??.
4. Efectua el caso de uso [\[PI UC6 Obtener Punto de Interés\]](#),
5. Hace clic sobre el botón .
6. Verifica que todos los campo se hayan llenado. [Trayectoria B]

7.  Busca la información que satisface los parametros de entrada para generar el mapa temático.[Trayectoria C]
8.  Genera el mapa temático.
9.  Muestra el Mapa Temático.
10. Fin de trayectoria





### **Trayectoria alternativa A: Cancelar**

**Condición:** El usuario no desea generar un mapa.

- A1**  Oprime el boton atrás del navegador.
- A2** Fin de trayectoria



### **Trayectoria alternativa B: Campos sin seleccionar**

**Condición:** Al proporcionar la información obligatoria para generar el mapa se omitieron campos.

- B1**  Detecta que al menos un campo obligatorio fue omitido.
- B2**  Muestra un mensaje de error correspondiente.
- B3**  Proporciona la información requerida.
- B4**  El caso de uso continúa en la trayectoria principal en el paso 4.
- B5** Fin de trayectoria

### **Trayectoria alternativa C: No Existe Información**

**Condición:** Con los datos que se ingresaron es imposible generar un mapa temático

- C1**  Muestra un mensaje de error: *No hay informacion que satisfaga los parametros proporcionados.*
- C2**  Continúa el caso de uso en la trayectoria principal en el paso 3.
- C3** Fin de trayectoria

### B.5.4. Diagrama de Clases

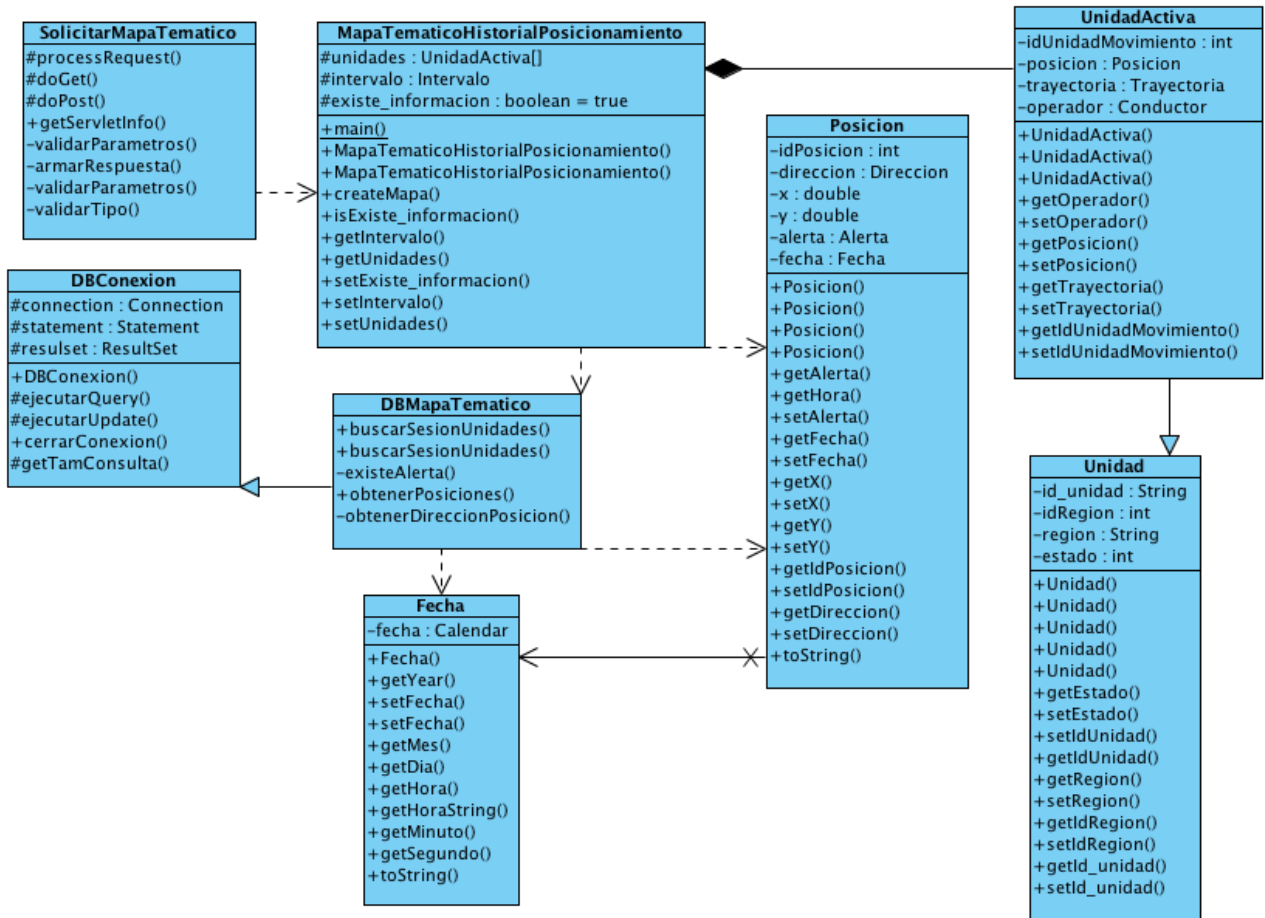


Figura B.10: Diagrama de Clases Mapa Temático Historial Posicionamiento con Punto de Interés

### B.5.5. Diagrama de Secuencia

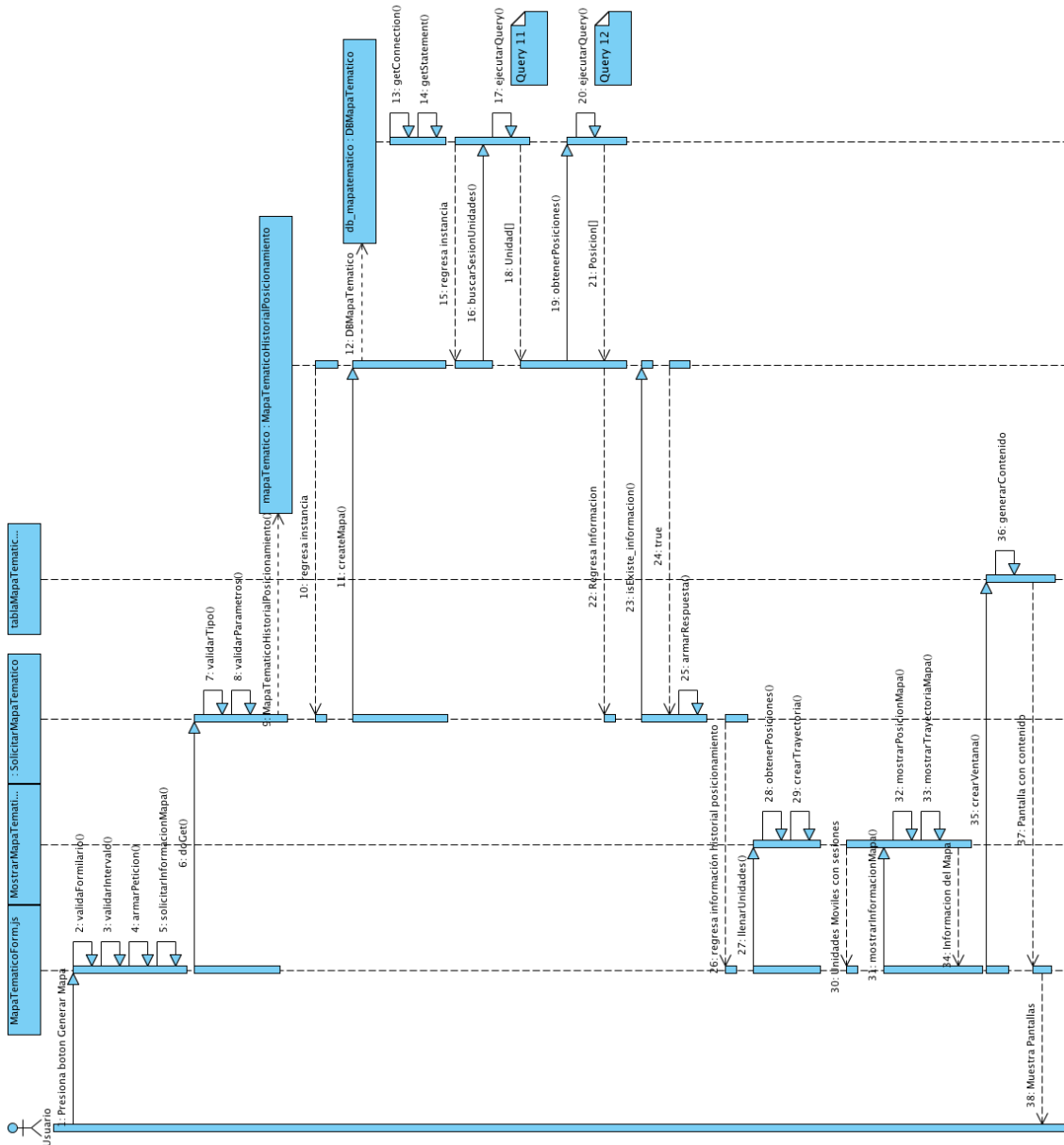


Figura B.11: Diagrama de Secuencia Mapa Temático Historial Posicionamiento con Punto de Interés



## B.6. UC6 Obtener Punto de Interés

### B.6.1. Descripción completa

El Sistema de Localización y Seguimiento de Dispositivos Móviles permite a los usuarios obtener las coordenadas de un punto de interés dentro del mapa base del Distrito Federal.

### B.6.2. Atributos importantes

<b>Id:</b>	UC6
<b>Nombre:</b>	Obtener Punto de Interés
<b>Version:</b>	1.0
<b>Actores:</b>	Administrador, Supervisor
<b>Propósito:</b>	Obtener punto de interés en un mapa
<b>Resumen:</b>	Obtener punto de interés haciendo un clic sobre un punto de un mapa
<b>Entradas:</b>	Punto en el mapa
<b>Salidas:</b>	Coordenada(Longitud y Latitud)
<b>Precondiciones:</b>	No
<b>Postcondiciones:</b>	El Sistema de Localización y Seguimiento de Dispositivos Móviles obtiene la coordenada
<b>Prioridad:</b>	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RU1

### B.6.3. Trayectorias

#### Trayectoria principal: Principal

1. Muestra el mapa base en pantalla.
2. Hace clic sobre el Boton
3. Hace clic sobre el mapa, es decir, en el punto que desea obtener.
4. Obtiene Coordenada.
5. Fin de trayectoria



## B.7. UC7 Configurar Políticas

### B.7.1. Descripción completa

El Sistema de Localización y Seguimiento de Dispositivos Móviles permite al usuario administrador configurar los parámetros bajo los cuales se emite la alerta de desplazamiento nulo. También puede modificar el intervalo de tiempo en el cual los dispositivos reportan su posición.

### B.7.2. Atributos importantes

<b>Id:</b>	UC7
<b>Nombre:</b>	Configurar Políticas
<b>Version:</b>	1.1
<b>Actores:</b>	Administrador
<b>Propósito:</b>	Configurar alertas de dispositivos con movimiento nulo.
<b>Resumen:</b>	El Usuario puede cambiar el intervalo de tiempo en que los dispositivos envían su posición y el intervalo de tiempo en que una alerta se disparará en caso de que un dispositivo no manifieste un desplazamiento significativo.
<b>Entradas:</b>	Intervalo de tiempo de monitoreo y el tiempo de desplazamiento nulo.
<b>Salidas:</b>	
<b>Precondiciones:</b>	El intervalo de tiempo de monitoreo y el tiempo desplazamiento nulo deben ser positivos ( $t > 0$ ).
<b>Postcondiciones:</b>	El sistema configurará la alerta de desplazamiento nulo al intervalo de tiempo dado y el tiempo de monitoreo almacenandolos en la base de datos.
<b>Prioridad:</b>	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
<b>Referencias:</b>	RF3.4.1, 3.4.2 y 3.6

### B.7.3. Trayectorias

#### Trayectoria principal: Principal

1. Hace clic sobre el botón Políticas.
2. Muestra la pantalla Políticas.[Trayectoria A]
3. Modifica el tiempo de monitoreo y/o alerta de desplazamiento nulo.

4. ☞ Hace clic sobre el botón .
5. ○ Establece la fecha final de la vigencia del tiempo de monitoreo y/o alerta de desplazamiento nulo a la fecha actual.
6. ○ Establece el nuevo tiempo de monitoreo y/alerta de desplazamiento nulo.
7. Fin de trayectoria

### Trayectoria alternativa A: Abandonar Políticas

**Condición:** El Usuario no desea cambiar las Políticas

**A1** ☞ Hace clic sobre el botón

**A2** Fin de trayectoria

#### B.7.4. Diagrama de Clases

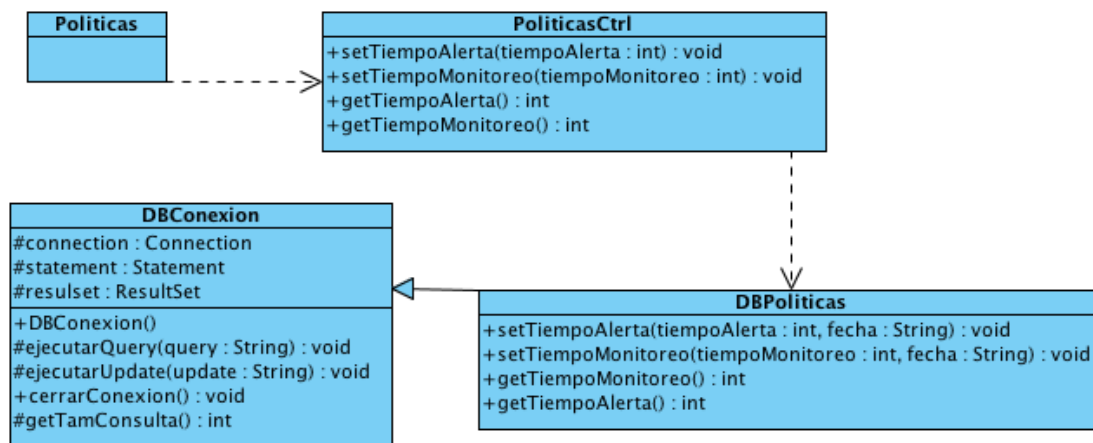


Figura B.12: Diagrama de Clases Configurar Políticas



### B.7.5. Diagrama de Secuencia

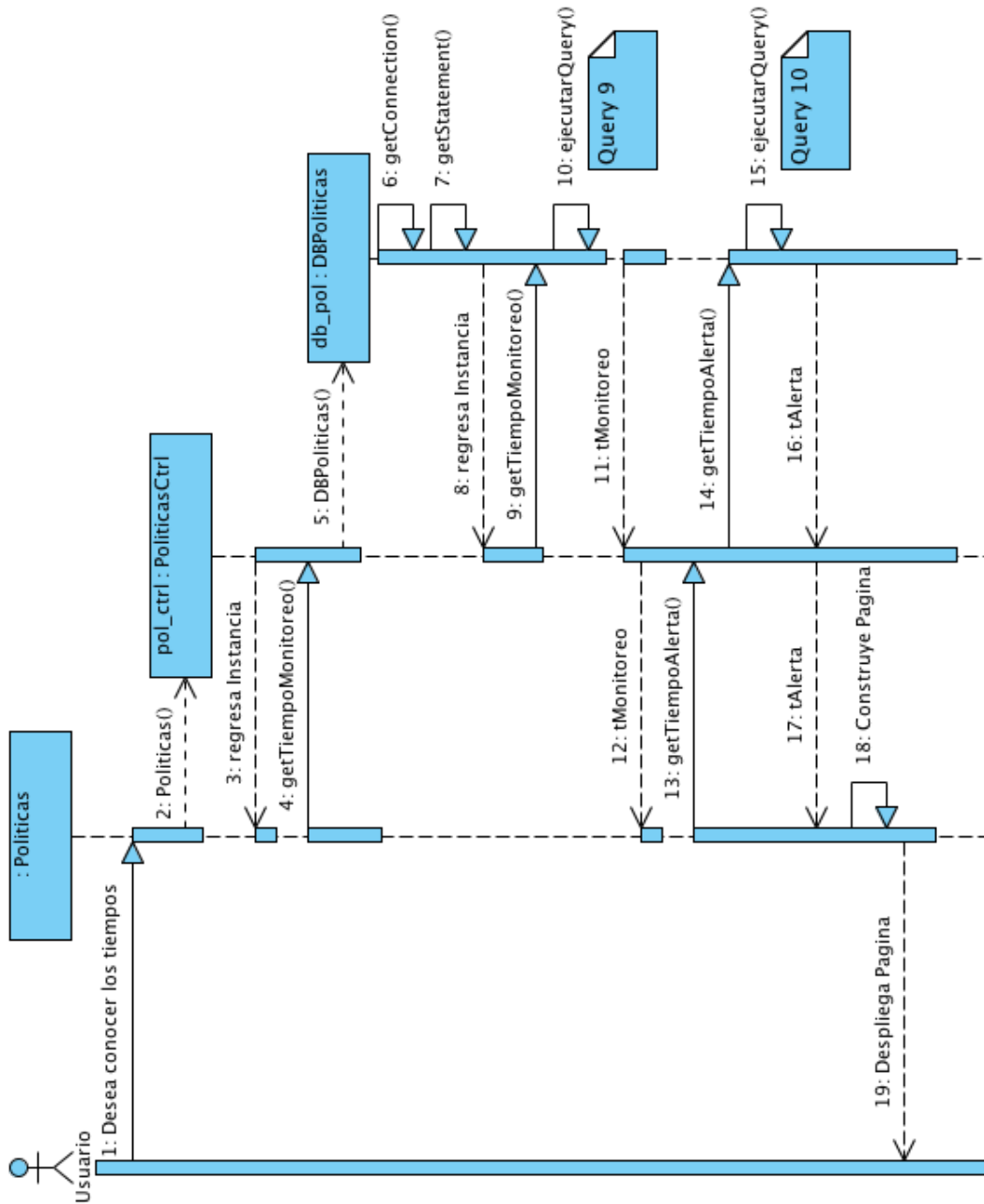


Figura B.13: Diagrama de Secuencia Configurar Políticas



# Bibliografía

- [1] *Professional Android Application Development*, chapter Hello, Android, pages 1 – 17. Wiley, 2009.
- [2] Kpper A. Location-based services: Fundamentals and operation. 2005.
- [3] Tsalgatidou A., Veijalainen J. Katasonov A., and Hadjiefthymiades S. Mobile e-commerce and location-based services: Technology and requirements. In *Proceedings of the 9<sup>th</sup> Scandinavian Reserch Conference on Geograpgical Information Services*, pages 1 – 4, 2003.
- [4] Reza B. *Mobile Computing Principles: Designing and Developing Mobie Applications with UML and XML*, chapter Mobility and Location-Based Services, pages 676 – 722. Cambridge University Press, 2005.
- [5] Len Bass, Paul Clements, , and Rick Kazman. *Software Architecture in Practice, Second Edition*. Addison Wesley, 2003.
- [6] Tim Berners-Lee, Roy T. Fielding, Larry Masinter, Henrik Frystyk, and Paul J. Hypertext Transfer Protocol – HTTP/1.1. May 1999. RFC-2616.
- [7] Tim Berners-Lee, Roy T. Fielding, Larry Masinter, Henrik Frystyk, and Paul J. HTTP Over TLS. May 2000. RFC-2818.
- [8] Reza B'Far. *Mobile Computing Principles: Designing and Developing Mobie Applications with UML and XML*, chapter Introduction to mobile Computing: Added Dimensions of Mobile Computing, pages 3 – 28. Cambridge University Press, 2005.
- [9] Jakub Borkowski, Jarno Niemela, and Jukka Lempiainen. Cellular location techniques supporting agps positioning. *IEEE Computer Society*, pages 429 – 433, 2005.
- [10] Taehwan Choi and Mohamed G. Gouda. HTTP Integrity: A Lite and Secure Web against World Wide Woes.
- [11] World Wide Web Consortium. Scripting and AJAX. 2011. <http://www.w3.org/standards/webdesign/script>.

- [12] Peter H. Dana. Global positioning system overview. Technical report, The Geographer's Craft Project, Department of Geography, The University of Colorado, <http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>, Enero 2000.
- [13] Jie Du and Matthew J. Barth. Next generation automated vehicle location systems: Positioning at the lane level. In *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, volume 9, pages 48 – 57, Marzo 2008.
- [14] Jie Du, Jason Masters, and Matthew Barth. Lane level positioning for in-vehicle navigation and automated vehicle location(avl) systems. *IEE Intelligent Transportation Systems Conference, Washington DC*, Octubre 2004.
- [15] Peter Eeles. What is a software architecture? <http://www.ibm.com/developerworks/rational/library/feb06/eeles/>.
- [16] M.A. Elahi, Y.A. Malkani, and M. Fraz. Design and implementation of real time vehicle tracking system. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1 – 5, Febrero 2009.
- [17] ESRI. Geographic information system. <http://www.esri.com/what-is-gis/index.html>.
- [18] Jason Farrell and George Nezelek. Rich internet application, the next stage of application development. *Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces*, June 25-28 2007.
- [19] Carlos Ferreyra and Pedro Mejía. *Desarrollo de un Sistema de Tecnologías de Cómputo, Comunicaciones y Geolocalización*. Departamento de Computación - CINVESTAV-IPN, 2011.
- [20] Maximiliano Firtman. *AJAX, Web 2.0 para profesionales*. Alfaomega, 2008.
- [21] Behrouz A. Forouzan. *Transmisión de Datos y Redes de Comunicaciones*, chapter Nivel de aplicación, pages 735 – 789. MCGRAW HILL, 2007.
- [22] Syeda Uzma Gardazi and Arshad Ali Shahid. Survey of Software architecture description and usage in software industry of pakistan. *2009 International Conference on Emerging Technologies*, pages 395 – 402, 2009.
- [23] David Garlan and Mary Shaw. An introduction to software architecture. In *Advances in Software Engineering and Knowledge Engineering*, pages 1–39. Publishing Company, 1993.
- [24] Joshua Greenfel. Automatic vehicle location(avl) for transit operation. *10th Mediterranean Electrotechnical Conference*, II:656 – 659, 2000.
- [25] W3C Technical Architecture Group. Architecture of the World Wide Web, Volume One. December 2004. <http://www.w3.org/TR/2004/REC-webarch-20041215/>.

- [26] M. Ilyas and I. Mahgoub. *Mobile computing handbook*. Auerbach Publications, 2005.
- [27] Jani Jarvinen, Javier DeSalas, and Jimmy LaMance. Assisted gps: A low infrastructure approach. Technical report, GPS World, The Business and Technogy of Global Navigation and Positioning, Marzo 2002.
- [28] La Jornada. Hay 3 mil 300 millones de usuarios de telefonía celular. Nov 2007. <http://www.jornada.unam.mx/2007/11/30/index.php?section=economia&article=030n1eco>.
- [29] Duo Li and Mai Kiuchi. Extension of Client-Server Applications to the Internet. *Proceedings of IEEE TENCON'02*, pages 355 – 358, 2002.
- [30] Jizhong LI and Muqing WU. A positioning algorithm of agps. *IEEE Computer Society, International Conference on Signal Processing Systems*, pages 385 – 388, 2009.
- [31] Chen Liyan. Application research of using design pattern to improve layered architecture. *2009 IITA International Conference on Control, Automation and Systems Engineering*, pages 303 – 306, 2009.
- [32] Hafedh Mili. *Reuse-Based Software Engineering: Techniques, Organization, and Controls*, chapter Architectural Frameworks, pages 215 – 238. Wiley Inter-Science, 2002.
- [33] Jörg Noack, Hamarz Mehmaneche, Homayoun Mehmaneche, and Andreas Zender. Architectural patterns for web applications, 2000.
- [34] Agrawal P. and Famolari D. Mobile computing in next generation wireless networks. In *Proceedings of the 3<sup>rd</sup> International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM Press, pages 32 – 39, 1999.
- [35] Yongwan Park and Fumiyuki Adachi. *Enhanced Radio Acces Technologies for Next Generation Mobile Communication*, chapter Overview of Mobile Communication, pages 1 – 37. Springer Netherlands, 2007.
- [36] Domenico Porcino. Location of third generation mobile devices: A comparision between terrestrial and satellite position systems. *Philips research Laboratories, UK*, 2001.
- [37] National Space Based Positioning, Navigation, and Timing Coordination Office. Global position system. Noviembre 2010. <http://www.gps.gov/systems/gps/>.
- [38] Erica Sadun. *The iPhone Developers CookBook*, chapter Introducing the iPhone SDK. Adison Wesley, 2008.
- [39] Ian Sommerville. *Ingenieria del Software*, chapter Diseño Arquitectónico, pages 215 – 238. Addison Wesley, 2002.
- [40] Ian Sommerville. *Ingenieria del Software*, chapter Procesos de Software. Prentice Hall, Enero 2005.

- [41] Albert Kai sun Wong, Tim Kam Woo, Albert Ting Leung Lee, Xiaoming Xiao, and Vincent Wing-Hei Luk Kwok Wai Cheng. An agps-based elderly tracking system. *Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology*, 2009.
- [42] Roy T. Fielding and Jeffrey and C. Mogul. Uniform Resource Identifier (URI): Generic Syntax. December 2005. RFC-3986.
- [43] Angus K.Y. Wong. Cell phones as mobile computing devices. *IEEE Computer Society*, Mayo/Junio 2010.
- [44] Jianping Yuan, Huamin Jia, and Qun Fang. Applications of gps to space vehicles: Analysis of space environment and errors. *IEEE AES Systems Magazine*, 1998.