



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

**Desarrollo de un Sistema de Tecnologías de
Computo, Comunicaciones y Geolocalización**

Tesis que presenta

Carlos Jonathan Ferreyra Rodríguez

Para obtener el grado de:

Maestro en Ciencias en Computación

Director de la Tesis:

Dr. Pedro Mejía Alvarez

Índice general

Resumen	VII
Abstract	IX
1. Introducción	1
2. Conceptos Generales	5
2.1. Cómputo Móvil	6
2.1.1. Sistemas Inalámbricos y Móviles	6
2.1.2. Añadiendo Dimensiones al Cómputo Móvil	7
2.2. Cómputo Ubicuo	10
2.3. Comunicación Móvil	11
2.3.1. El Sistema GSM	11
2.3.2. Arquitectura del Sistema GSM	12
2.3.3. GPRS	17
2.3.4. Arquitectura del Sistema GPRS	18
2.3.5. Redes inalámbricas 802.11	19
2.4. GPS	20
2.5. Sistemas Operativos	21
2.5.1. Componentes de un Sistema Operativo	23
2.5.1.1. Administrador de Procesos	24
2.5.1.2. Administrador de Memoria	25
2.5.1.3. Administrador de Dispositivos	26

2.5.1.4.	Administrador de Archivos	26
2.6.	Sistemas Operativos para Dispositivos Móviles	27
2.6.1.	Symbian OS	28
2.6.1.1.	Arquitectura	29
2.6.2.	iPhone OS	31
2.6.2.1.	Arquitectura	32
2.6.3.	Android	33
2.6.3.1.	Arquitectura	35
2.6.3.2.	Componentes de una Aplicación	37
2.6.3.3.	Ciclo de Vida de una Aplicación	38
3.	Diseño de la propuesta	41
3.1.	Funciones del Sistema	41
3.2.	Arquitectura del Sistema	43
3.2.1.	Casos de Uso	43
3.2.2.	Diagramas de Secuencias	46
3.2.3.	Módulos del Sistema	49
3.2.3.1.	Módulo de Persistencia	49
3.2.3.2.	Módulo de Conexión	52
3.2.3.3.	Módulo de Seguridad	52
3.3.	Flujo de Información del Sistema	53
3.3.1.	Cliente	53
3.3.1.1.	Consulta de Información	56
3.3.1.2.	Envío de Multas	57
3.3.1.3.	Rastreo de Usuarios	58
3.3.2.	Servidor	58
3.3.2.1.	Servidor Web-Móvil de Infracciones	59
3.3.3.	Sistema Web de Infracciones	59

4. Pruebas y Resultados	61
4.1. Desempeño del Sistema	62
4.1.1. Tiempos de Respuesta sobre Redes WiFi y GPRS	62
4.1.1.1. Sockets TCP	62
4.1.1.2. Sockets UDP	65
4.1.1.3. Peticiones GET	68
4.1.1.4. Peticiones POST	69
4.1.1.5. Elección del Protocolo del Comunicaciones	73
4.1.2. Duración de la Batería del Cliente Móvil	74
4.1.3. Primer Tiempo de Ajuste del GPS	74
4.1.4. Uso del Sistema	76
4.1.4.1. Registro de Eventos	78
5. Conclusiones y trabajo futuro	79
5.1. Conclusiones	79
5.2. Trabajo Futuro	81
Índice de Tablas	83
Índice de Figuras	85
Bibliografía	87

Resumen

En la actualidad las capacidades de los dispositivos móviles se han incrementado considerablemente en comparación a hace 10 años, teniendo un poder de cómputo mayor, diversas formas de comunicarse y transmitir una mayor cantidad de información, sensores y dispositivos incorporados como el GPS que proporcionan un mayor conocimiento de su entorno. De esta forma surge la necesidad de desarrollar un proyecto capaz de integrar todas estas tecnologías en un framework para dispositivos móviles para la recolección de datos, capaz de comunicarse en ambas vías entre los dispositivos móviles y una base de datos central, para la realización de diversas tareas.

Como caso de estudio se ha planeado el desarrollo de un sistema de infracciones de tránsito de la policía del D.F., capaz de levantar infracciones *in situ* teniendo actualizado el servidor con todos los eventos ocurridos. Asimismo el oficial será capaz de verificar el estado de un vehículo por sus placas, indicando en dado caso si se encuentra robado, con alguna violación al reglamento de tránsito, etcétera.

Abstract

At present the capabilities of mobile devices have increased considerably compared to 10 years ago, having a greater computing power, various ways to communicate and transmit a greater amount of information, sensors and embedded devices such as GPS to provide greater knowledge of their environment. Thus arises the need to develop a project capable of integrating all these technologies in a system for mobile devices for data collection, able to communicate in both directions between mobile devices and a central database to perform various tasks.

As a case study is planned to develop a system of traffic infringements by police in Mexico City, capable of lifting the spot infringements taking the server updated with all the events. The officer also will be able to check the status of a vehicle on their plates, indicating in which case if it is stolen, with some violation of traffic rules and so forth.

Capítulo 1

Introducción

Los dispositivos móviles como lo son los teléfonos celulares, se han convertido en elementos de la vida cotidiana de tal forma que es más común cada vez adquirir un teléfono inteligente o *smartphone*. Estos teléfonos móviles incluyen dispositivos de entrada/salida como acelerómetros, pantallas táctiles y drivers para acceso a GPS que mejoran de forma notoria el uso de estos. Aunado con un plan de datos se obtiene una nueva plataforma para el desarrollo de aplicaciones móviles innovadoras.

Según la firma consultora *Morgan Stanley*[17], la tendencia principal del uso del Internet está basada sobre dispositivos móviles. Indicando que en este momento nos encontramos sobre la quinta era tecnológica la cual es el cómputo móvil. Las cuatro eras anteriores fueron la era del mainframe en la década de 1950 y 1960, la era de las mini-computadoras en la década de 1970, las computadoras personales en la década de 1980, el cómputo en Internet sobre la década de 1990. En esta nueva era la mayor cantidad de usuarios provendrán de dispositivos móviles de esta forma se espera que con el rápido crecimiento para el 2014 se iguale el número de dispositivos móviles a computadoras de escritorio conectados a Internet.

Una de las aplicaciones más interesantes del cómputo móvil se basa en servicios de geolocalización, la cual aporta un uso más personalizado del dispositivo. Contrario

de la computadora de escritorio de la cual se puede obtener una menor cantidad de información.

Existen en el mercado una gran cantidad de teléfonos celulares, la mayoría con sistemas operativos propietarios, sin embargo recientemente se ha desarrollado un sistema operativo móvil libre el cual no ofrece barreras artificiales. El Sistema Operativo *Android* ofrece a los desarrolladores la posibilidad de crear aplicaciones libres[14] que permiten tomar una completa ventaja de las características del teléfono e incluso permiten modificar el SO según se requiera a las necesidades del desarrollo.

En este trabajo de tesis se plantea el desarrollo de una aplicación de captura de datos en distintas localidades geográficas utilizando el dispositivo móvil. Se pretende que este dispositivo móvil este conectado a un servidor de cómputo fijo al cual transmitirá información en tiempo real.

La aplicación a desarrollar sobre el teléfono celular consiste en un sistema de infracciones vehiculares. El sistema esta conformado por dos componentes principales: un teléfono móvil desde el cual se realizan diversos eventos como son consulta de datos y aplicación de multas, y un sistema servidor de *Web* el cual recibe y responde a las solicitudes por parte del móvil y muestra un portal *Web* para monitorear en tiempo real el desarrollo del sistema.

Objetivo General

Desarrollar un sistema basado en tecnologías móviles (teléfono celular) capaces de integrar diversas tecnologías de localización y comunicación para la creación del caso de estudio de un sistema de infracciones.

Objetivos Particulares

- Analizar y comprender el desarrollo de software para el sistema operativo *Android*.

- Implementar la lectura del *GPS* para conocer la ubicación del teléfono celular.
- Desarrollar las interfaces necesarias para operar el teléfono celular.
- Implementar un servidor local con soporte a bases de datos.
- Implementar la comunicación con un servidor sobre diversos medios (*WiFi/GPRS/3G*).
- Desarrollo del sistema de infracciones.

Para desarrollar el trabajo de tesis planteado y al mismo tiempo cumplir con los objetivos propuestos, se ha seguido con una metodología que consta de los siguientes pasos:

Revisión del caso de estudio: El caso de estudio es una aplicación para servicios en el área gubernamental, el cual permitirá llevar un control en tiempo real sobre la inserción de nuevos eventos en el sistema por parte de los trabajadores, en este caso policías de tránsito consultando y aplicando multas vehiculares. Toda la información solicitada es enviada a través del móvil hacia un servidor el cual se encarga de responder las solicitudes pertinentes. El objetivo de la revisión es determinar la arquitectura del sistema.

Análisis de los procesos de geolocalización, comunicación http, y almacenamiento local en el móvil: En este paso se define la funcionalidad de cada uno de estos procesos así como sus limitantes y su aplicación.

Desarrollo de los módulos de geolocalización, comunicación http, y almacenamiento local en el móvil: Una vez analizada la operación de los procesos en cuestión, se desarrollan de tal forma que se puedan emplear para unirse en el sistema propuesto.

Integración de los módulos. Desarrollo, pruebas e instalación de la aplicación propuesta.

Definición de métricas: Las métricas propuestas permitirán evaluar el desempeño de la aplicación. Se definirán algunas métricas tales como el consumo de energía, el tiempo consumido por operación y la precisión del módulo de geolocalización.

Realización de pruebas y análisis de resultados. En esta etapa se evaluará la aplicación de acuerdo a las métricas establecidas y se estudiarán los resultados obtenidos.

El contenido del presente trabajo de tesis se organiza de la siguiente manera: En el Capítulo 2, se abordan conceptos generales de la telefonía celular, de las formas de comunicación y geolocalización. En el Capítulo 3 se presenta la arquitectura del sistema de recolección de datos del trabajo de campo y se describen cada uno de sus componentes. En el Capítulo 4 se presenta el caso de estudio y los resultados obtenidos de acuerdo a las métricas obtenidas. Finalmente en el Capítulo 5 se presentan las conclusiones del trabajo de tesis y el trabajo futuro.

Capítulo 2

Conceptos Generales

La tecnología móvil ha avanzado significativamente en los últimos 10 años, de tal forma que se ha vuelto parte común de nuestro entorno diario. A tal punto que un teléfono celular, por ejemplo, es mucho más que un dispositivo para hacer llamadas telefónicas. Entre las características a mencionar mas relevantes es que actualmente el teléfono celular es una computadora portátil que entre otras funciones, permite la comunicación entre varios usuarios, la creación y reproducción de archivos multimedia, el envío de información a tasas de transferencia cada vez mayores, la localización de dispositivos en tiempo real, así como la incorporación de una mejor interfaz mediante el uso de diversos sensores que proporcionan un ambiente más natural al usuario.

El desarrollo de aplicaciones para dispositivos móviles, a diferencia de aplicaciones para computadoras de escritorio, presenta una serie de nuevos retos. El primero de estos retos consiste en las limitantes propias del dispositivo móvil, como la duración de la pila, el tamaño de la pantalla, el teclado reducido, así como la disponibilidad de vías de comunicación específicas. Además, otro reto importante consiste en en la necesidad de conocer una serie de tecnologías involucradas, como son los sistemas operativos propietarios, los lenguajes de programación y ambientes de desarrollo especiales. En este capítulo se presentan los conceptos necesarios que dan pie al desarrollo del presente trabajo de tesis.

2.1. Cómputo Móvil

Los sistemas de computación móvil son sistemas de cómputo que pueden ser ubicados en distintas localidades geográficas y sus capacidades pueden ser empleadas mientras el dispositivo se encuentra en movimiento [12]. Para distinguir a los sistemas de computación móvil de otros tipos de sistemas de cómputo, se pueden identificar por las tareas para las que fueron diseñados y por la forma en que son operados. Existen muchas funciones que un sistema computacional móvil puede realizar y que un sistema computacional estacionario no puede. Estas funcionalidades son la razón de caracterizar de forma independiente a los sistemas computacionales móviles sobre los estacionarios.

Aunque los sistemas computacionales móviles han existido a lo largo de sus contrapartes estacionarias. La mayoría de las herramientas, metodologías y arquitectura en la ingeniería de software actuales están actualmente dirigidas a las necesidades de sistemas estacionarios.

2.1.1. Sistemas Inalámbricos y Móviles

En la conectividad inalámbrica, los dispositivos de computación móvil se conectan con otros dispositivos sobre la red de muchas formas. De hecho, esta cuestión ha sido una gran fuente de confusión entre comunicaciones inalámbricas y el cómputo móvil. Los dispositivos de cómputo móvil no necesitan ser inalámbricos, ejemplo de esto son las calculadoras y relojes electrónicos [12].

Los sistemas de comunicación inalámbrica son un tipo de comunicación. Lo que distingue a un sistema de comunicación inalámbrica de otros, es que el canal de comunicación es el medio que utiliza para comunicarse. En este caso, es un medio no guiado (espacio). Hay una variedad de canales de ondas guiadas, como la fibra óptica

o cables metálicos. Los sistemas de comunicación inalámbrica no utilizan una guía de onda para guiar a lo largo de la señal electromagnética desde el emisor al receptor. Se basan en el hecho de que las ondas electromagnéticas pueden viajar por el espacio si no hay obstáculos que las bloqueen. Los sistemas inalámbricos de comunicación a menudo se utilizan en sistemas de computación móvil para facilitar la conectividad de red, pero no son sistemas de computación móvil.

Recientemente, las redes computacionales han evolucionado a pasos agigantados. Estas redes han empezado a cambiar fundamentalmente la manera en que vivimos. Las redes y el cómputo distribuido son dos de los segmentos más grandes que son el foco de los esfuerzos actuales de la informática. Las redes y los dispositivos computacionales están cada vez más interlazados. La mayoría de los sistemas de computación móvil de hoy, pueden conectar a una red a través de cable o de conexiones inalámbricas. Debido a la naturaleza de los sistemas de computo móvil, la conectividad a la red de los sistemas móviles utiliza cada vez con mas frecuencia a sistemas de comunicaciones inalámbricas en lugar de los cables. Aunque no es un requisito para un sistema móvil para ser inalámbrico, la mayoría de sistemas móviles son inalámbricos. Sin embargo, debemos destacar que la conectividad en una conexión inalámbrica y la movilidad son ortogonales en lo que se refiere a su naturaleza a pesar de que pueden ser complementarias.

2.1.2. Añadiendo Dimensiones al Cómputo Móvil

Debería ser obvio observar que cualquier sistema de computación móvil también puede ser fijo. Sólo bastaría si detenemos su movimiento y pasaría a ser un equipo estacionario. Por lo tanto, podemos decir que los sistemas de computación móvil son un superconjunto de los sistemas informáticos fijos. Por tal razón, tenemos que observar a aquellos elementos que están fuera del subconjunto de computación estacionaria. Estas dimensiones agregadas ayudaran a seleccionar las variables que nos permitan

conquistar los problemas de la informática móvil [12]. Las dimensiones de movilidad (Figura 2.1) son los siguientes:

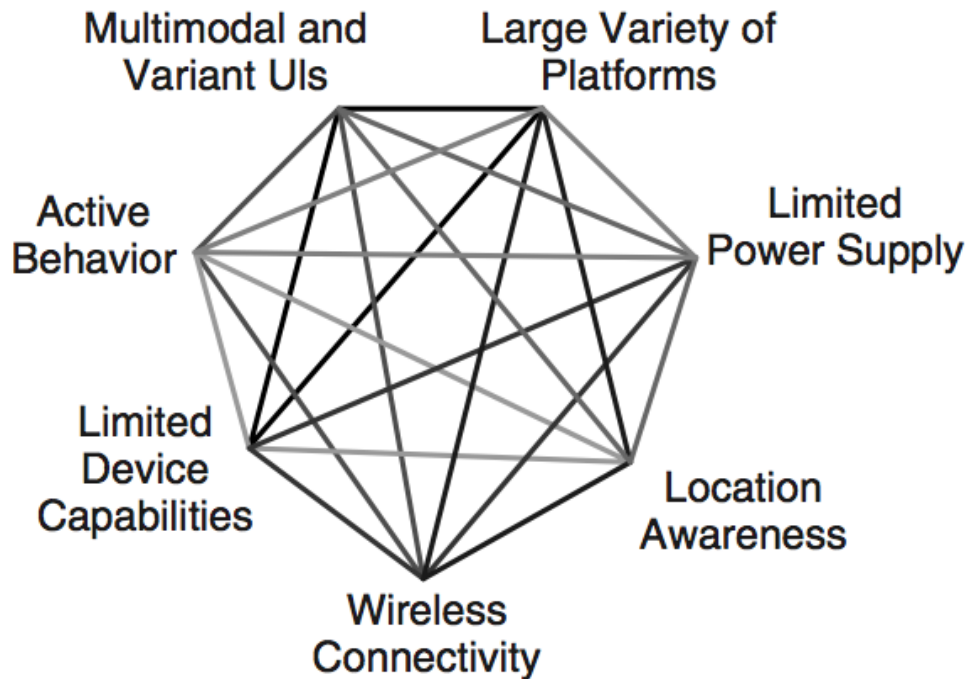


Figura 2.1: Dimensiones extras del cómputo móvil

- **Conocimiento de la ubicación.** Un dispositivo móvil está en constante cambio. El cambio de ubicación del dispositivo móvil y la aplicación móvil presentan a los diseñadores de las aplicaciones de dispositivos y software con grandes dificultades. Sin embargo, también nos presenta la oportunidad de utilizar la ubicación para mejorar la aplicación. Estos retos y oportunidades se pueden dividir en dos categorías generales: localización y ubicación de la sensibilidad.
- **Calidad de servicio (QoS) de la conectividad de la red.** Ya sea que se utilice una red alámbrica o una conexión inalámbrica, la movilidad significa la pérdida de la fiabilidad de la red de conectividad. El desplazamiento de un lugar a otro crea barreras físicas que resultan en un tiempo de desconexión de la red. Si

una aplicación móvil se utiliza en un sistema móvil alámbrico, el sistema móvil debe estar desconectado entre los intervalos en los que está conectado por el cable. Siempre es una cuestión de si un puerto de acoplamiento está disponible cuando sea necesario por no hablar de la calidad y el tipo de conexiones de red conectividad en ese puerto de acoplamiento. En el caso de la conectividad de red inalámbrica, las condiciones físicas pueden afectar significativamente la calidad de servicio (QoS).

- **Capacidades limitadas del dispositivo.** No es deseable tener un dispositivo móvil demasiado grande, por lo tanto los dispositivos móviles son de un tamaño moderado. Esta limitación en el tamaño físico restringe la capacidad de almacenamiento volátil y no volátil y el procesamiento en los dispositivos móviles. Las aplicaciones móviles son de recursos limitados, por lo cual aunque los diseñadores de aplicaciones modernas planeadas para ser ejecutadas en computadoras personales y los servidores que se preocupan menos de los recursos de sistema (como memoria y procesamiento), las aplicaciones móviles deben ser más eficientes para manejar estas limitaciones de recursos.
- **Fuente de alimentación limitada.** Así como existen limitaciones de almacenamiento y movilidad física que afecta la movilidad de la red, existen también limitaciones en cuanto a el tiempo de operación de los dispositivos móviles debido a que su fuente de energía son las baterías de tamaño reducido.
- **Apoyo a una amplia variedad de interfaces de usuario.** Los usuarios utilizan aplicaciones estacionarias mientras trabajan en un PC o en dispositivo similar. El teclado, el ratón y el monitor han demostrado ser interfaces bastante eficaces de usuario para tales aplicaciones. Esto no es del todo aplicable para las aplicaciones móviles. Ejemplos de algunas interfaces alternativas son las interfaces de usuario de voz, pantallas más pequeñas, el lápiz y otros dispositivos señaladores, las pantallas táctiles y los teclados miniatura.
- **Proliferación de la plataforma.** Debido al tamaño reducido de los dispositi-

vos móviles y debido a que poseen menos hardware que una PC, por lo general estos tienen un precio reducido. Esto significa que más fabricantes puedan competir en la producción de estos dispositivos. Estos dispositivos más baratos y más pequeños, por lo general se suelen utilizar para propósitos especiales.

- **Transacciones activas.** La mayoría de las aplicaciones estacionarias de hoy tienen una restricción que puede reducir los beneficios de un sistema de aplicaciones móviles enormemente: El usuario debe iniciar todas las interacciones con el sistema. Llamamos a los sistemas de este tipo como sistemas pasivos, ya se encuentran en espera de alguna señal externa que el usuario les indique realizar. En aplicaciones estacionarias esto funciona bien. La mayoría de la gente utiliza una computadora de forma estática o pasiva (sin realizar cambios de posición geográfica).

Como ejemplo de sistema activo, surgió recientemente el modelo *Push*. En este modelo, durante la comunicación un productor de información anuncia la disponibilidad de ciertos tipos de información, y un consumidor interesado se adhiere a esta información. De esta forma el productor publica periódicamente esta información. Los sistemas *push*, por definición, son sistemas activos. Por ejemplo, un usuario particular puede navegar por la Web y al comprar algunos productos en línea, se le puede notificar del cambio en el precio de una acción en particular. En este ejemplo, el sistema asume un papel activo en el inicio de la comunicación con el usuario sobre un tema en particular.

2.2. Cómputo Ubicuo

El cómputo ubicuo se define como la integración de la informática en el entorno de la persona, de tal forma que los ordenadores no se perciban como objetos diferenciados [15]. Aunque la idea de tal entorno surgió en la década de los ochentas y su evolución ha empezado a acelerarse recientemente en los últimos años. Dentro del campo de la computación y de los sistemas de información, la computación ubicua

establece el inicio de una nueva era computacional. La primera era de la computación estuvo dominada por los grandes sistemas informáticos, los cuales se concebían para dar servicios a muchos usuarios. La segunda era fue marcada por la individualización del cómputo, en donde el modelo era un dispositivo de cómputo por usuario. Existe una etapa posterior, llamada cómputo móvil que se distingue por considerar aspectos de movilidad tanto del usuario así como de los dispositivos de cómputo. La principal diferencia entre el cómputo móvil y cómputo ubicuo es que en el primero los dispositivos siguen prestando sus servicios de manera independiente y en la última se conjugan de tal manera que los dispositivos y los servicios que ofrecen están inmersos en el entorno de las personas.

2.3. Comunicación Móvil

Los dispositivos móviles, tales como los teléfonos inteligentes, pueden contar con varias interfaces de comunicación. Estos dispositivos poseen por lo general una interfaz para servicio telefónico celular tal como GSM, pero adicionalmente pueden contar con una o más interfaces tales como sensores Infrarrojos, USB (Universal Serial Bus), Bluetooth o Wi-Fi. A continuación se describen las interfaces de comunicación más importantes presentes en los dispositivos móviles.

2.3.1. El Sistema GSM

El Sistema Global para Comunicaciones Móviles (GSM por sus siglas en inglés) es la tecnología que soporta la mayoría de las redes de telefonía celular en el mundo. Actualmente la tecnología GSM es usada por más de dos mil millones de personas en más de 200 países en el mundo y contabiliza el 81 % del mercado mundial de telefonía celular, que es de alrededor de 212 países [7]. Esta ubicuidad significa que los subscriptores puedan usar sus teléfonos a través del mundo empleando los operadores locales.

En 1982, el principal cuerpo de gobierno de los operadores de telecomunicaciones europeos, conocido como CEPT (*Conférence Européenne des Postes et Télécommunications*) creó el comité *Groupe Spéciale Mobile*, lo cual dio origen a las siglas GSM.

La tarea que se le asignó al comité fue definir un nuevo estándar para un sistema de radio celular para toda Europa en la banda de los 900 Mhz [16]. Con el transcurso del tiempo, CEPT evolucionó en una nueva organización, que se conoce como el Instituto Europeo de Estándares de Comunicación (ETSI por sus siglas en inglés), la cual no cambió la tarea de GSM. La meta de GSM fue sustituir las tecnologías nacionales ya sobrecargadas y por lo tanto caras de los países miembros, con un estándar internacional.

A diferencia de las tecnologías celulares predecesoras, en GSM sus señales de voz y datos son digitales considerandola de segunda generación.

GSM es una red celular y es una red formada por celdas de radio, cada una con su propio transmisor, conocidas como estaciones base. Estas celdas son usadas con el fin de cubrir diferentes áreas para proveer cobertura sobre una área más grande que el de una sola celda.

2.3.2. Arquitectura del Sistema GSM

La red GSM utiliza una estructura celular como se muestra en la figura 2.2. La idea básica de una red celular es dividir la gama de frecuencias disponible, para asignar solo partes de ese espectro de frecuencias a cualquier estación transmisora, y así reducir el alcance de una estación base para reutilizar las escasas frecuencias tanto como sea posible. Una de las principales metas de la planeación de la red es reducir la interferencia entre diferentes estaciones base.

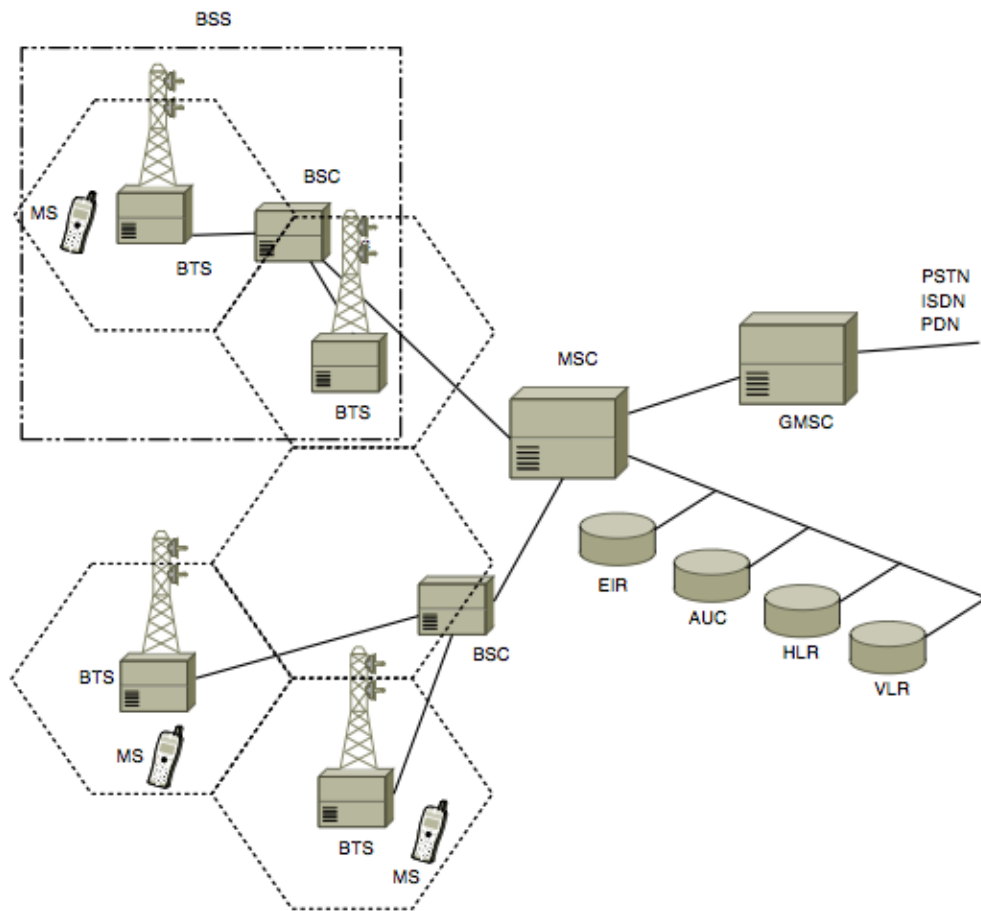


Figura 2.2: Estructura celular de red GSM

Además de la ventaja de reutilizar frecuencias, una red celular también tiene las siguientes ventajas :

- Un número en aumento de estaciones base incrementa el costo de la infraestructura y líneas de acceso.
- Todas las redes requieren que, a medida que la estación móvil se mueve, una llamada activa sea entregada de una célula a otra; proceso conocido como entrega (handover).
- La red tiene que mantenerse informada de la ubicación aproximada de la estación móvil, aun sin una llamada en curso, para ser capaz de entregar una llamada entrante a esa estación móvil.
- El segundo y tercer punto requieren una amplia comunicación entre la estación móvil y la red, así como entre los distintos elementos de la red.

Una red GSM está compuesta por varios elementos: la estación móvil (MS), el módulo de identificación del suscriptor (SIM), el subsistema de estación base (BSS), la estación base transmisora (BTS), la estación base controladora (BSC), la unidad de transcodificación y adaptación (TRAU), el centro de conmutación de servicios móviles (MSC), el registro de ubicación de origen (HLR), el registro de ubicación de visitante (VLR), y el registro de identidad del equipo (EIR). Todos estos juntos forman una red pública terrestre móvil (PLMN por sus siglas en inglés).

La figura 2.3 muestra una visión general de los subsistemas GSM, cuyos elementos se describen a continuación:

- **Estación móvil.** GSM-PLMN contiene tantos MS como sean posibles, los cuales están disponibles en varios estilos y clases de potencia.

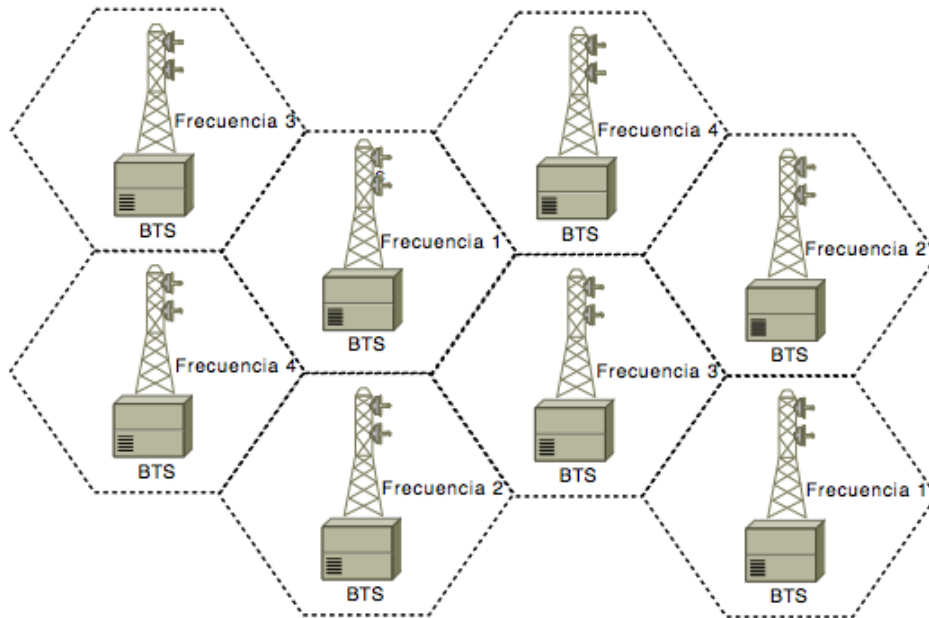


Figura 2.3: Arquitectura del Sistema GSM

- **Módulo de identidad del suscriptor (SIM).** GSM distingue entre la identidad del suscriptor y la del equipo móvil. El SIM determina el número de directorio y las llamadas facturadas a un suscriptor. El SIM es una base de datos en el lado del usuario. Físicamente, consiste de un chip, el cual el usuario debe insertar en el teléfono GSM antes de que pueda ser usado. El SIM se comunica directamente con el VLR e indirectamente con el HLR.
- **Subsistema de estación base (BSS).** A través de la Interfaz-aire el BSS ofrece una conexión entre las MS de un área limitada y el subsistema de la red de conmutación (NSS). El BSS consiste de los siguientes elementos: uno o más BTS, un BSC y un TRAU.
- **Estación base transmisora-receptora (BTS).** Un gran número de BTSs se encargan de las tareas relacionadas con el radio y proporcionan la conectividad entre la red y la estación móvil (MS) por medio de la Interfaz-aire.

- **Estación base controladora (BSC).** Los BTS de un área están conectados al BSC por medio de una interfaz llamada la interfaz Abis. El BSC se encarga de las funciones centrales y del control de los subsistemas (BSS). El BSS comprende el BSC mismo y los BTSs conectados.
- **Unidad de transcodificación y adaptación (TRAU).** En un sistema GSM, la compresión de datos es llevada a cabo tanto en el MS como en el TRAU. Desde la perspectiva de la arquitectura, el TRAU es parte del BSS.
- **Centro de conmutación de servicios móviles (MSC).** Un gran número de BSCs son conectados al MSC por medio de la interfaz-A. El MSC es muy similar a un teléfono digital regular, intercambia y es accedido por redes externas exactamente de la misma manera. Las principales tareas de un MSC son el enrutamiento de llamadas entrantes y salientes, y la asignación de canales de usuario en la interfaz-A.
- **Registro de ubicación de origen(HLR).** El HLR es un subcentro de una red GSM. Un HLR es un repositorio que almacena los datos de un gran número de suscriptores. Un HLR puede ser considerado como una gran base de datos que administra los datos de literalmente cientos de miles de suscriptores. Cada PLMN requiere de al menos un HLR.
- **Registro de ubicación de visitante (VLR).** El VLR fue ideado de tal manera que el HLR no se sobrecargaría con peticiones de datos acerca de sus suscriptores. Al igual que el HLR, un VLR contiene datos del suscriptor, pero solo parte de los datos que están en el HLR y solo mientras el suscriptor particular circula en el área por la cual el VLR es responsable.
- **Registro de identidad de equipo (SIM).** Debido a que las identidades de los suscriptores y de sus equipos móviles están separadas, los equipos robados pueden ser reutilizados simplemente usando cualquier SIM válido. Para prevenir

esa clase de abuso cada equipo terminal GSM contiene un identificador único, el identificador internacional de equipo móvil (IMEI por sus siglas en inglés).

2.3.3. GPRS

GSM fue originalmente diseñado para soportar únicamente conexiones de conmutación de circuitos al nivel de la interfaz de radio, con tasas de transferencia de datos de usuario de hasta 9.6 Kb/s [16]. La utilización de un circuito conmutado significa que el usuario ocupa un canal completo de tráfico durante todo el tiempo que dura la llamada, aun cuando este canal solamente sea utilizado durante una pequeña fracción de tiempo. Cuando se trata de tráfico en ráfaga el resultado es una utilización de los recursos de radio de forma altamente ineficiente [1]. Esto puede ser superado utilizando un servicio de conmutación de paquetes. Debido a que el canal solo será asignado cuando sea necesario, y liberado tan pronto como termine la transmisión de los paquetes. De esta manera varios usuarios pueden compartir un mismo canal físico.

Para corregir las ineficiencias antes señaladas han sido desarrolladas dos tecnologías: el paquete de datos celular digital (CDPD por sus siglas en inglés) y el servicio de radio general de paquetes (GPRS por sus siglas en inglés).

GPRS es un nuevo servicio portador para GSM que mejora significativamente y simplifica el acceso inalámbrico a redes de paquetes de datos. GPRS intenta optimizar los recursos de radio y de red, manteniendo una estricta separación entre los subsistemas de radio y red, aunque el subsistema de red es compatible con otros como el acceso de radio GSM. Los recursos de interfaz de radio pueden ser compartidos dinámicamente entre circuitos conmutados y el servicio de paquetes, como una función de la carga de servicio y de las preferencias del operador. La tasa de transferencia puede variar de 9 Kb/s a aproximadamente de 150 Kb/s por usuario.

La transmisión de paquetes GPRS permite un método de cobro más amigable

para el usuario que la ofrecida por los servicios de conmutación de circuitos. En los servicios de conmutación de circuitos, el cobro se basa en la duración de la conexión y normalmente se factura por minuto o por segundo. En contraste con esto, con los servicios de conmutación de paquetes, la transferencia de datos regularmente es facturada por la cantidad de información transmitida. La ventaja para el usuario es que puede estar en línea por un largo periodo de tiempo, pero será facturado por el volumen de datos transmitidos.

2.3.4. Arquitectura del Sistema GPRS

Con el fin de integrar GPRS en la arquitectura GSM existente se requieren dos nuevos componentes adicionales de red: el nodo de apoyo de pasarela GPRS (GGSN por sus siglas en inglés), y el nodo de apoyo de servicio GPRS (SGSN por sus siglas en inglés). El GGSN es responsable de la entrega y enrutamiento de los paquetes de datos entre la estación móvil (MS) y las redes de paquetes de datos externas (PDN por sus siglas en inglés) por medio del punto de referencia Gi. El SGSN es responsable de la entrega de paquetes de datos desde y hacia las estaciones móviles dentro de su área de servicio. Sus tareas incluyen el encaminamiento y transferencia de paquetes, el manejo de la movilidad, el manejo de enlace lógico y las funciones de autenticación y facturación [6]. El registro de ubicación del SGSN almacena información y perfiles de usuario de todos los usuarios GPRS registrados con este SGSN.

El nodo de apoyo de pasarela (GGSN) actúa como una interfaz entre la red troncal y las redes externas de paquetes de datos. Este convierte los paquetes provenientes del SGSN al formato apropiado del protocolo de paquetes de datos (PDP por sus siglas en inglés) y los envía a la correspondiente red de paquetes de datos. En la dirección opuesta, las direcciones PDP de los paquetes de datos entrantes son convertidas a la dirección GSM del usuario destino. En la figura 2.4 se ilustra la arquitectura del sistema.

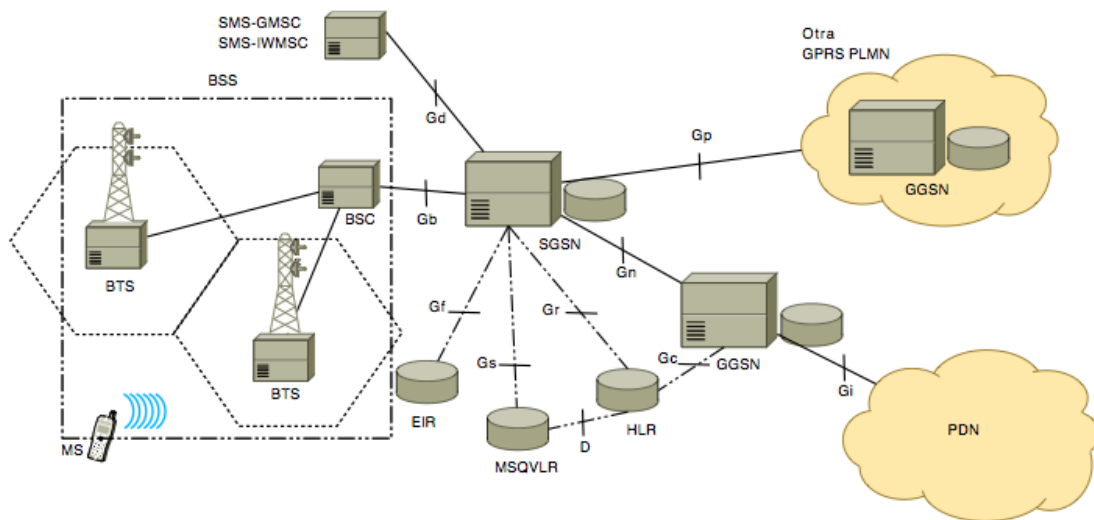


Figura 2.4: Arquitectura del sistema GPRS

2.3.5. Redes inalámbricas 802.11

Los protocolos IEEE 802.11 y sus esquemas de transmisión son uno de los logros más notables de normalización de protocolos. Una gran cantidad de dispositivos se encuentran en la actualidad basados en esta norma. Empezó como una extensión inalámbrica para redes de área local en 1997 y desde entonces, ha sido mejorado y ampliado gradualmente hacia una muy flexible y bien entendida tecnología. Debido a que el protocolo 802.11 fue construido para sistemas de radio en el espectro sin licencia, prácticamente no existe una limitación en esta norma. El espectro sin licencia es a menudo armonizado en todo el mundo, lo que significa que dichos sistemas de radio se pueden utilizar prácticamente en cualquier región del planeta. Debido a su inherente simplicidad, el 802.11 es el estándar dominante para los sistemas comerciales de comunicación inalámbrica [6].

El IEEE publicó el estándar original IEEE 802.11 en 1997 como una especificación para un esquema de transmisión y un protocolo de control de acceso al medio, para las redes de área local inalámbricas (WLAN). Una versión mejorada se publicó en 1999. Al mismo tiempo, el 802.11a y el 802.11b fueron los primeros subestándares

para extender el 802.11 y se publicaron en forma paralela en 1999. Actualmente el 802.11 está dividido en muchos más subestándares, cada uno de los cuales trata extensiones particulares. Al igual que el IEEE 802.3 (Ethernet) y el 802.5 (Token Ring), el estándar 802.11 se centra en las dos capas más bajas (1 y 2) del modelo de referencia OSI (*Open System Interconnection*). Este modelo de referencia divide la capa de control de enlace de datos (DLC) en las subcapas de control de enlace lógico (LLC) y control de acceso al medio (MAC). El 802.11 define los esquemas de transmisión de la capa física (capa 1 de OSI) y el protocolo MAC, pero no la funcionalidad del LLC.

Para el LLC, el sistema 802.11 puede depender de protocolos generales que son usados en otros estándares 802. Esta capa LLC es especificada independientemente para todas las redes 802 alámbricas o inalámbricas.

2.4. GPS

El GPS (Global Positioning System) es un sistema de navegación basado en satélites que fue desarrollado por el Departamento de Defensa de los Estados Unidos en el año de 1978. Este sistema ofrece posicionamiento continuo de información en cualquier parte del mundo. Debido a que sirve a un número ilimitado de usuarios y es utilizado con propósitos generales, el GPS es un sistema pasivo, es decir los usuarios únicamente reciben las señales de los satélites [5].

GPS consiste de una constelación de satélites organizados en planos orbitales el cual llegó a ser completamente operacional en julio de 1995. Cada satélite de dicha constelación transmite una señal que contiene la distancia entre el receptor y los satélites GPS y las coordenadas de los satélites en función del tiempo, entre otras cosas. Las señales son controladas por relojes atómicos de alta precisión, incluidos en cada satélite, debido a que una variación de una milésima de segundo en los cálculos puede resultar en una variación de 300 km.

La idea básica de GPS es muy simple: se puede conocer la ubicación sobre la tierra en un punto dado, si se conoce la ubicación de mínimo tres satélites en un momento dado para posteriormente obtener distancias hacia el punto de interés, la ubicación del punto puede ser determinada por triangulación. Una vez que el receptor obtiene las medidas de posicionamiento puede calcular las coordenadas de ubicación directamente en el dispositivo o enviar las medidas de vuelta al servidor de la red para que las procese. GPS brinda información en tres dimensiones: latitud, longitud y elevación.

Un punto a tomar en cuenta es el “Primer Tiempo de Ajuste”(TTF Time to first fix por sus siglas en ingles) el cual es el tiempo requerido por un receptor GPS para adquirir señales satelitales, datos de navegación y efectuar calculo a la solución de la posición llamado ajuste.

2.5. Sistemas Operativos

Un sistema operativo es un programa que administra el hardware de una computadora. Un aspecto interesante de los sistemas operativos es lo variado que son para completar las mismas tareas a diferentes escalas. Los sistemas operativos para *Mainframes* y para computadoras personales son diseñados principalmente para optimizar los recursos generales, mientras que los sistemas operativos para computadoras móviles, no solo es importante optimizar los recursos, debido a su tamaño y peso, sino que también es importante reducir su consumo de energía.

Para entender a un sistema operativo es necesario también entender el funcionamiento de un sistema computacional entero, ya que un sistema operativo administra cada pieza de hardware y software. El objetivo de un Sistema Operativo (SO) consiste en brindar primordialmente a los usuarios una conveniente interfaz del tipo “top-down”, dejando la complejidad en la parte de abajo y mostrando al usuario una

vista más simple del sistema. Las computadoras modernas consisten de procesadores, memorias, relojes, medios de almacenamiento, interfaces de red y una variedad de otros dispositivos. Visto de otra forma, el trabajo de un sistema operativo es de proveer una forma ordenada y controlada la ejecución de procesos, memorias y dispositivos de Entrada/Salida (I/O) entre los diferentes programas que compiten por estos recursos.

La administración de recursos incluye la multiplexión (compatir) recursos de dos formas: en tiempo y en espacio. Cuando un recurso es multiplexado, diferentes programas pueden hacer uso de él. Por ejemplo, con solo un unidad de procesamiento central (CPU) y múltiples programas queriendose ejecutarse sobre este, el sistema operativo aloja el primer programa en espera para ejecutarse durante un intervalo de tiempo en el CPU, luego los siguiente programas programas que aun siguen esperando y eventualmente el primer programa una vez más para iniciar de nueva cuenta el ciclo. Determinar como el recurso es multiplexado en el tiempo, quien es el siguiente a ejecutarse y por cuanto tiempo es parte de la tarea del sistema operativo.

El otro tipo de multiplexaje es en espacio. En lugar de que los clientes tomen su turno, cada uno toma una parte del recurso. Por ejemplo, la memoria principal es normalmente dividida en torno a la cantidad de aplicaciones ejecutandose, así cada uno puede residir al mismo tiempo. A sumiendo que existe suficiente memoria para albergar múltiples programas, es más eficiente tener varios programas en memoria que solo uno ocupándola toda, especialmente si solo necesitan una pequeña fracción del total. Esto plantea problemas de equidad, protección y similares, estos detalles son resueltos por el sistema operativo. Otro recurso que es multiplexado en el espacio es el medio de almacenamiento (disco duro). En sistemas de un solo medio, puede almacenar varios archivos al mismo tiempo. La asignación de espacio y mantener un control sobre los bloques es una típica operación de un sistema operativo

.

2.5.1. Componentes de un Sistema Operativo

El sistema operativo esta compuesto de cuatro subsistemas[8], cada uno controla distintas categorías de los recursos de un sistema computacional: unidad de procesamiento central, : memoria principal, dispositivos y archivos. Estos cuatro subsistemas son llamados Administrador de Procesador, Administrador de Memoria, Administrador de Dispositivos y Administrador de Archivos respectivamente. La interacción entre estos se muestra en la figura ??.

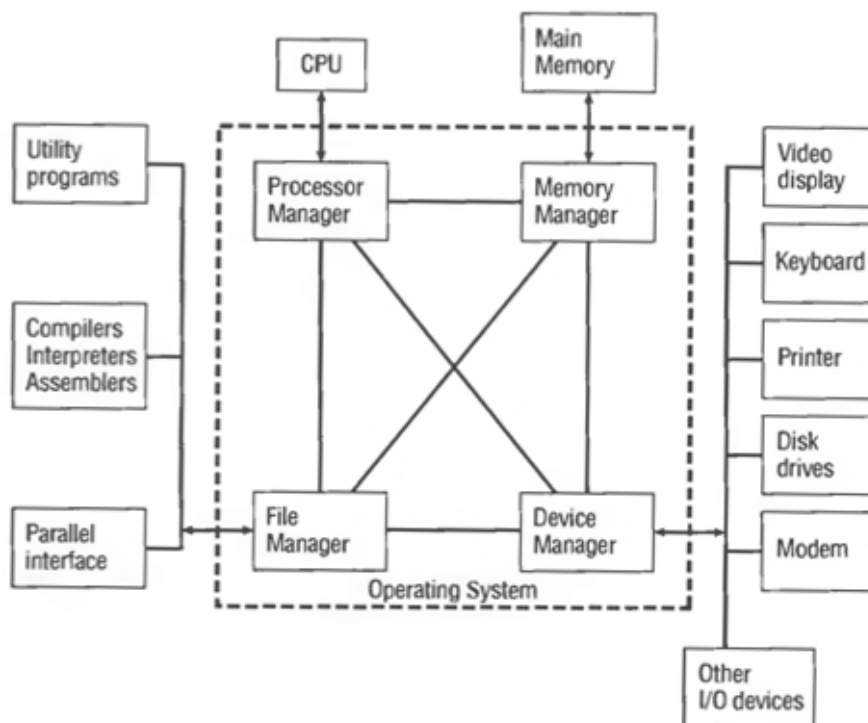


Figura 2.5: Componentes básicos de un sistema operativo

Sin tener en cuenta el tamaño de la configuración del sistema cada una de estos administradores llevan a cabo las siguientes tareas:

- Monitorear continuamente los recursos
- Cumplir políticas para determinar quien obtiene algún recurso, cuando y en que

cantidad.

- Asignación de recursos cuando sea necesario
- Cancelar la asignación o reclamar recursos cuando sea necesario.

2.5.1.1. Administrador de Procesos

Un programa no hace nada a menos que sus instrucciones sean ejecutadas por un CPU. Un programa en ejecución, es llamado un proceso.

Un proceso necesita ciertos recursos -incluyendo tiempo de CPU, memoria, archivos y dispositivos I/O- para completar una tarea. Estos recursos también son asignados al proceso cuando es creado o es asignado mientras esta en ejecución.

Un proceso es una unidad de trabajo de un sistema. Como un sistema consiste de una colección de procesos, donde algunos de ellos son procesos del sistema operativo y el resto procesos de usuario. Todos estos procesos pueden ejecutarse potencialmente de forma concurrente.

El sistema operativo es responsable de las siguientes actividades en conjunto con el administrador de procesos de:

- Crear y eliminar procesos de usuario y sistema
- Suspender y resumir procesos
- Proveer mecanismos para la sincronización de procesos
- Proveer mecanismos para la comunicación de procesos
- Proveer mecanismos para el manejo de *deadlocks*

2.5.1.2. Administrador de Memoria

La memoria es un repositorio de acceso rápido para datos compartidos entre el CPU y dispositivos de I/O. El procesador central lee instrucciones de la memoria de los diversos programas en ejecución. La memoria principal es generalmente un gran dispositivo que el CPU es capaz de direccionar y acceder directamente. Para que el CPU pueda procesar datos desde el disco duro, esa información debe ser transferida a la memoria principal. De la misma forma, las instrucciones deben estar en la memoria para ser ejecutadas.

Para que un programa pueda ser ejecutado debe ser mapeado a direcciones absolutas y cargado en memoria. Al ser ejecutado el programa, se accesan las instrucciones de programa y datos de la memoria. Eventualmente, al terminar el programa el espacio en memoria es declarado libre y el siguiente programa puede ser cargado y ejecutado.

Para mejorar la utilización del CPU y la velocidad de respuesta de la computadora, las computadoras de uso general deben mantener varios programas en memoria, creando con esto la necesidad de un administrador de la memoria. Hay diferentes esquemas de administración de la memoria y la eficacia de un algoritmo dado depende de cada situación.

El administrador de memoria del sistema operativo es responsable de realizar las siguientes actividades:

- Llevar un registro de que partes de la memoria están siendo utilizados y por quien.
- Decidir que procesos o información debe ser movida o eliminada de la memoria
- Asignar y desasignar espacio de memoria cuando se necesite

2.5.1.3. Administrador de Dispositivos

Los dispositivos de entrada/salida (I/O) es la forma de interactuar la computadora con cada dispositivo periférico. Una de las principales funciones de la computadora es controlar los dispositivos I/O, emitir comandos a los dispositivos, atender interrupciones y manejar errores. También debe proporcionar una interfaz entre los dispositivos y el resto del sistema que sea fácil y sencilla de emplear. En la medida de lo posible, la interfaz debe ser la misma para todos los dispositivos.

El administrador de dispositivos del sistema operativo es responsable de llevar a cabo las siguientes actividades:

- Registrar el estado de cada dispositivo.
- Cumplir políticas para determinar que proceso debe acceder a que dispositivo y por cuanto tiempo.
- Asignar los dispositivos.
- Desasignar los dispositivos.

2.5.1.4. Administrador de Archivos

Para hacer mas conveniente la computadora a los usuarios, el sistema operativo provee una vista lógica y uniforme de la información almacenada. El sistema operativo abstrae de las propiedades físicas de los dispositivos de almacenamiento para definir una unidad de almacenamiento lógica, el archivo. El sistema operativo mapea archivos en medios físicos y el acceso a estos archivos el los dispositivos del almacenamiento.

La administración de archivos es uno de los componentes más visibles de un sistema operativo. Las computadoras pueden almacenar información en diferentes tipos de medios físicos. Cada uno de estos medios posee sus propias características y organización física. El sistema operativo implementa el concepto abstracto de un archivo

por un administrador de archivos. Asimismo, los archivos son organizados en directorios para hacer más fácil su uso. Finalmente con múltiples usuarios teniendo acceso a archivos, es deseable el control, de como pueden ser accesibles.

El administrador de archivos del sistema operativo es responsable por las siguientes actividades:

- Crear y borrar archivos.
- Crear y borrar directorios para organizar archivos.
- Soporte básico para la manipulación de archivos y directorios.
- Mapear en un segundo dispositivo de almacenaje.
- Respalidar archivos en un medio de almacenaje estable (no volátil).

2.6. Sistemas Operativos para Dispositivos Móviles

Un sistema operativo móvil es aquel que controla un dispositivo móvil al igual que una computadora de escritorio es controlada por un sistema operativo tradicional . Sin embargo, los sistemas operativos móviles en cierto grado son más simples y están orientados a la conectividad inalámbrica, a los formatos multimedia y a las diferentes formas de introducir información a estos.

La importancia creciente de los dispositivos móviles ha intensificado la competencia entre diversos desarrolladores de tecnología como *Google*, *Microsoft*, *Apple*, *Nokia* por capturar al mercado más grande.

Los sistemas operativos para móviles pueden ser encontrados en diversos teléfonos inteligentes incluyendo *Symbian OS*, *iPhone OS*, *RIM's BlackBerry*, *Windows Phone*, *Linux*, *Palm WebOS*, *Android* y *Maemo*. *Android*, *WebOS* y *Maemo* se han convertido

en los que encabezan de sistemas operativos basados en *Linux* y el *iPhone OS* deriva de un sistema operativo basado en *BSD* y *NeXTSTEP*, los cuales están basados en *Unix*.

Los sistemas operativos más comunes usados en teléfonos inteligentes (en el segundo cuarto del 2010) son:

- Symbian Os
- Iphone Os
- Android

A continuación se menciona una breve descripción de los sistemas operativos mencionados.

2.6.1. Symbian OS

Symbian se formó en 1998 por *Ericsson*, *Motorola*, *Nokia* y *Pision* para proporcionar un estándar común y habilitar el mercado en masa de una nueva generación de dispositivos inalámbricos [11]. Matsushita se unió a Symbian en 1999, y en enero de 2002 la empresa conjunta Sony-Ericsson tomó participación en Symbian y en abril de 2002 Siemens también se unió a Symbian como accionista. Desde el inicio, la meta de Symbian fue desarrollar un sistema operativo y una plataforma de software para teléfonos móviles avanzados. Para este propósito, el sistema operativo EPOC desarrollado por *Pision* formó una base sólida. Es un sistema operativo modular multitarea de 32 bits diseñado para dispositivos móviles.

Symbian es un sistema operativo multitarea con características que incluyen un sistema de archivos, un marco de interfaz gráfica de usuario, soporte para multimedia, una pila TCP/IP y bibliotecas soporte de características de comunicación encontradas en los teléfonos inteligentes [2].

El núcleo del SO Symbian consiste de la base (microkernel y controladores de dispositivos), *middleware* (servidores de sistema, seguridad y marco de aplicaciones), y comunicaciones (telefonía, mensajería y redes de área personal). Este núcleo permanece común a los diferentes dispositivos que soportan al SO *Symbian*. Cuando el SO Symbian se monta al nuevo hardware, la base necesita ser cambiada, pero esto no afecta las capas superiores.

La arquitectura del sistema es modular y está diseñado con un enfoque orientado a objetos.

La mayoría de las operaciones están basadas en un modelo cliente-servidor que permite a todas las aplicaciones usar los servicios proporcionados por el sistema, así como otras aplicaciones. El SO *Symbian* también contiene un marco de seguridad que proporciona administración de certificados y módulos de criptografía.

El microkernel se ejecuta directamente en el procesador en modo privilegiado. Es responsable del manejo de la energía y del manejo de la memoria, posee los controladores de los dispositivos, y también es la capa de interfaz entre el hardware y el software.

2.6.1.1. Arquitectura

La arquitectura de las versiones del sistema operativo Symbian desde la versión siete a la nueve comparten la misma agrupación en capas [`arquitectura_symbian`] como se muestra en la figura 2.6.

- **Capa *framework UI*** . La capa más alta que el SO Symbian provee *frameworks* y librerías para la construcción de la interfaz de usuario incluyendo la herencia de clases para el control de la interfaz de usuario y otros *frameworks* y utilidades, incluyendo clases concretas *widgets* usadas para los componentes de la interfaz.
- **Capa de Aplicación de Servicios**. Está capa provee soporte independiente

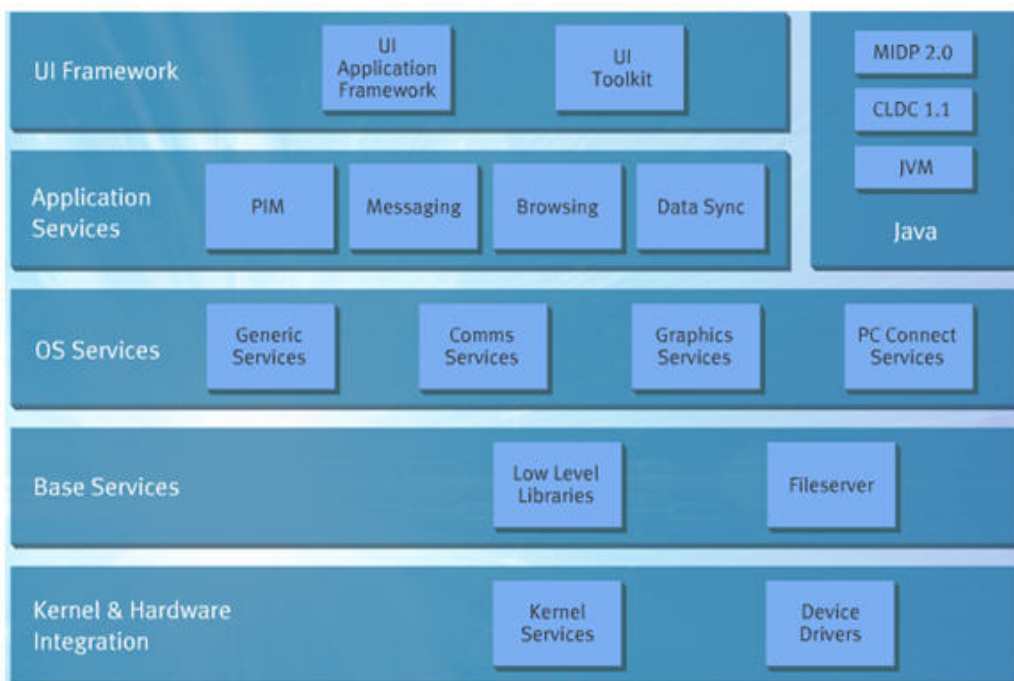


Figura 2.6: Arquitectura del Sistema Operativo Symbian

de la interfaz de usuario para aplicaciones en el SO Symbian . Estos servicios están divididos en tres grupos:

- Servicios a nivel de servicios. Como una aplicación base del *framework* usada por todas las aplicaciones.
 - Servicios proveedores de lógica de tecnología específica. Proveen una manera estandarizada de comunicarse con tecnologías particulares, como la mensajería y protocolos multimedia.
- **Java ME.** Java abarca el *framework* de interfaz de usuario y la capa de aplicación Servicios, abstrayendo elementos para aplicaciones Java.
 - **Capa de Servicios del SO.** La capa del *middleware* del SO Symbian proporciona servicios, *frameworks* y librerías que extienden el sistema base en un sistema operativo completo. Los servicios son divididos en cuatro bloques principales que proporcionan la mayor parte de la tecnología específica.

- Servicios genéricos de sistema operativo
 - Servicios de comunicaciones
 - Servicios multimedia y de gráficos
 - Servicios de conectividad.
- **Capa de servicios base.** La capa fundamental de SO Symbian proporciona el nivel más bajo de los servicios del lado del usuario, dependiendo sólo de la operación núcleo del sistema.
 - **Servicios del núcleo y la capa de interfaz de hardware.** La capa más baja de SO Symbian contiene el núcleo del sistema operativo y de apoyo de componentes que se derivan a las interfaces con el hardware subyacente, incluidos los controladores del dispositivo lógico y físico.

2.6.2. iPhone OS

iPhone OS. Es un sistema operativo desarrollado por Apple Inc. para los dispositivos móviles iPod Touch, iPhone y iPad. Está basado en una variante del Mach Kernel de Mac OS X [13].

La arquitectura subyacente del sistema y muchos módulos, son similares a los que se encuentran en Mac OS X. El núcleo en el *iPhone OS* se basa en una variante del mismo núcleo básico Mach que se encuentra en Mac OS X. Sin embargo algunas de las variantes entre ellos son [10]:

- **Solo una ventana.** Al contrario de un sistema operativo de escritorio donde múltiples programas coexisten, cada uno con la habilidad de crear y controlar múltiples ventanas, *iPhone* solo permite una ventana para trabajar.

- **Acceso Limitado.** Al contrario de una computadora donde el usuario puede tener acceso a todo desde los programas que ejecuta, el *iPhone* restringe las acciones que una aplicación puede ejecutar.
- **No hay un recolector de basura.** Se ha mencionado con anterioridad que *Cocoa Touch* usa *Objective-C 2.0*, pero ninguna característica de administración está presente en el *iPhone*.

2.6.2.1. Arquitectura

La arquitectura del IOS es similar a la arquitectura básica encontrada en un Mac OS X. En el nivel más alto, iOS actúa como un intermediario entre el hardware y las aplicaciones que aparecen en pantalla, pero en contadas ocasiones las aplicaciones se comunican directamente con el hardware [9]. Por el contrario, las aplicaciones se comunican con el hardware por un sistema de interfaces que protegen cambios en el hardware por parte de la aplicación. Esta abstracción hace fácil escribir aplicaciones que trabajen consistentemente en dispositivos con diferentes capacidades de hardware.

La implementación de las tecnologías iOS pueden ser visualizadas como un conjunto de capas, los cuales son mostradas en la figura 2.7. Las capas inferiores incluyen servicios fundamentales y tecnologías de las cuales todas las aplicaciones dependen. En las capas superiores están contenidas los servicios más sofisticados y las tecnologías.

- **Capa *Cocoa Touch*.** La capa de *Cocoa Touch* contiene la clave de muchos *frameworks* para el desarrollo de aplicaciones iOS. Define la capa de aplicaciones básicas y el apoyo a tecnologías clave como es el *multitasking*, la entrada básica *touch*, las notificaciones push y muchos servicios del sistema de alto nivel.

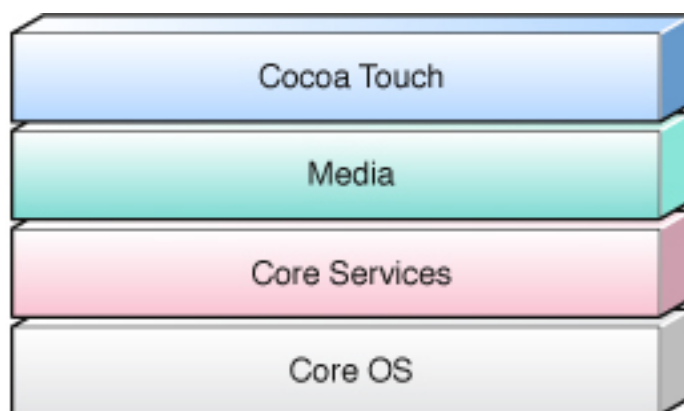


Figura 2.7: Arquitectura del Sistema Operativo iOS

- **Capa de Medios.** La capa de medios contiene las tecnologías de gráficos, audio y video orientadas a la creación de contenido multimedia.
- **Capa núcleo de servicios.** La capa núcleo de servicios contiene los servicios fundamentales que todas las aplicaciones utilizan. Aún si estas no emplean estos servicios de forma directa, muchas partes de una aplicación son construidas por encima de esta capa.
- **Capa núcleo del SO.** La capa del núcleo del SO contiene dos características de bajo nivel que la mayoría de otras tecnologías se basan. Incluso si no se emplean de forma directa son utilizadas por otros *frameworks* y en situaciones donde se requiere tratar a un nivel más detallado de seguridad o comunicaciones con un accesorio de hardware externo, se requiere utilizar los *frameworks* de esta capa.

2.6.3. Android

Android es un nuevo sistema operativo para teléfonos móviles desarrollado por Google y la *Open Handset Alliance*. La existencia de múltiples plataformas privadas en el mercado imposibilita un estándar en las tecnologías móviles, por lo cual al ser *Android* un SO abierto crea dicha alternativa.

Varias de las características principales a destacar del SO son [3]:

- **Una plataforma abierta y libre desarrollo basado en Linux.** La plataforma no está ligada de forma exclusiva a un proveedor de servicios o equipos que puedan restringir su uso.
- **Una arquitectura basada en componentes.** Las partes de una aplicación pueden ser utilizadas de diferentes formas a la planteadas por el desarrollador, de tal manera que es posible substituir componentes nativas del equipo, con sus propias versiones mejoradas.
- **Conjunto de servicios integrados.** Servicios basados en la utilización del GPS o la triangulación de redes celulares permiten personalizar la aplicación dependiendo de su ubicación. Una base de datos SQL completa que optimiza el almacenamiento local para la conexión ocasional y sincronización. La navegación y visualización de mapas pueden ser embebidos directamente en las aplicaciones, entre otros.
- **Administración automática sobre ciclo de vida de una aplicación.** Los programas están aislados entre sí por múltiples capas de seguridad, proporcionando un nivel de estabilidad mayor que otros teléfonos inteligentes.
- **Portabilidad sobre una amplia gama de hardware.** Los programas de *Android* están escritos en Java y ejecutados por la máquina virtual *Dalvik*, de modo que el código es portable sobre diferentes arquitecturas por mencionar ARM, X86, entre otras.

El Software Development Kit (SDK) de Android consiste en una serie de herramientas para ayudar al desarrollo de aplicaciones para Android. Esto incluye tanto un plugin para el IDE de Eclipse, emulador, herramientas de depuración, constructor de diseño visual, registro del monitor, entre otros.

2.6.3.1. Arquitectura

La arquitectura esta basada en una pila de software que incluye el SO, *middleware* y aplicaciones núcleo. Se basa en un kernel de Linux 2.6 para la funcionalidad del núcleo del sistema y se ejecuta el código escrito en el lenguaje de programación Java en una máquina especialmente diseñada virtual denominada Dalvik. Dalvik ejecuta archivos que están en las clases de Java compilado optimizado para minimizar la huella de la memoria [4]. La arquitectura general del sistema se encuentra en la figura 2.8.



Figura 2.8: Arquitectura del Sistema Operativo Android

La pila de software esta dividida en cuatro capas, que incluyen cinco diferentes grupos:

- **Capa de aplicación**

La plataforma de software Android viene con un conjunto de aplicaciones básicas como un navegador, cliente de correo electrónico, envío y recepción de

SMS, mapas, calendario, contactos entre otros. Todas estas aplicaciones fueron escritas utilizando el lenguaje de programación *Java*. Es importante mencionar que todas estas aplicaciones pueden correr simultáneamente, es posible escuchar música y leer un correo electrónico al mismo tiempo. Esta capa es la más conocida por los usuarios de los móviles.

- **Capa *framework* de aplicación**

El *framework* de aplicación es un software el cual es usado para implementar una estructura estándar de una aplicación para un sistema operativo específico. Con la ayuda de los administradores, *content providers*, y otros como los servicios es posible reensamblar funciones empleadas por otras aplicaciones existentes.

- **Capa de librerías**

Todas las librerías disponibles están escritas en C/C++. Estas serán llamadas desde una interfaz de *Java*. Estas incluyen el administrador de navegación, gráficos en 2D y 3D, manejo de archivos multimedia, bases de datos y el motor del navegador Web Webkit.

- **Capa *runtime***

Consiste de dos componentes. El primero es un conjunto de librerías *core* que provee la mayor parte de la funcionalidad disponible en el *core* de las librerías de el lenguaje de programación *Java*. El segundo la *Dalvik virtual machine* (DVM), es un componente que opera como un interprete entre la aplicación y el sistema operativo. Cada aplicación que es ejecutada sobre Android esta desarrollada en *Java* y el SO no es capaz de entender directamente el lenguaje del programa. De esta forma al no poder interpretarlo, los programas en *Java* son leídos y ejecutados por la DVM que actúa como intermediario entre ambos. Cada aplicación ejecuta su propio proceso, el cual es una instancia de la DVM. Una ventaja de este mecanismo es que diferentes programas no afectan a terceros, es decir si un programa se bloquea o para de forma inesperada, todo el sistema se sigue

ejecutando de forma normal.

- **Capa de *kernel***

Android se basa en un *kernel de linux 2.6* para un conjunto de servicios del sistema como es la seguridad, administración de memoria, procesos, red, entre otros. El *kernel* también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

2.6.3.2. Componentes de una Aplicación

Una característica central de Android es el hecho que una aplicación, puede hacer uso de elementos de otras aplicaciones (provisto por aquellas que lo permitan). De esta manera la aplicación no incorpora código de otra aplicación para hacer la liga, por el contrario, sólo comienza un segmento de la otra aplicación cuando se requiera.

El sistema debe ser capaz de comenzar el proceso de aplicación cuando esta se necesite e instanciar los objetos de *Java* para ese acometido. Por consiguiente, al contrario de las aplicaciones de la mayoría de los sistemas, las aplicaciones de Android no tienen un solo punto de entrada, sino poseen esencialmente componentes que el sistema puede instanciar y ejecutar cuando se necesiten. Existen cuatro tipos de componentes:

- **Actividades.** Una aplicación puede o no tener una interfaz gráfica, si la tiene, posee al menos una Actividad. La forma más sencilla de observar una Actividad es relacionarla a una pantalla visible, a menudo no existe una relación uno a uno entre las Actividades y las pantallas de la interfaz gráfica.
- **Servicios.** Si una aplicación tendrá una ciclo de vida largo es preferible colocar las tareas que tardan mucho o que no se conoce su tiempo de terminación. Por ejemplo, una tarea de sincronización de datos que se ejecuta frecuentemente. Al contrario que una Actividad un servicio no posee una interfaz gráfica, sin embargo se puede ejecutar en un segundo por un periodo indefinido de tiempo.

- **Broadcast receivers.** Un *Broadcast receiver* es un componente que escucha y reacciona a eventos globales. Por ejemplo, se puede registrar uno para reaccionar a una llamada entrante y ejecutar una debida acción.
- **Content providers.** Si una aplicación administra información y requiere exponerla a otras, un *Content provider* debe ser implementado. De esta forma, si un componente de una aplicación necesita información de otra aplicación se emplea su *Content provider* para poder acceder.

2.6.3.3. Ciclo de Vida de una Aplicación

Internamente, cada ventana de la interfaz de usuario es representada por una Actividad. Cada Actividad tiene su propio ciclo de vida, por el contrario una aplicación es una o más Actividades sobre un proceso de *Linux* para contenerlo.

En Android, una aplicación puede considerarse viva aún si su proceso ha muerto. Dicho de otra manera, el ciclo de vida de una Actividad no esta ligado con el ciclo de vida de un Proceso. Los Procesos solo son contenedores para distintas Actividades.

Durante el ciclo de vida, cada Actividad de un programa de Android puede estar en uno de los varios estados existentes, como se muestra en la figura 2.9. Al implementar la funcionalidad del programa no se posee el control sobre el estado del programa en si, esta función es llevada a cabo por el sistema, sin embargo se reciben notificaciones cuando el estado va a cambiar.

Al sobrescribir una serie de métodos listados a continuación, en la aplicación Android los llamara cuando sea necesario y actuar de forma adecuada ante dichos cambios de estado de la Aplicación.

- **onCreate(Bundle)** Llamada cuando la Actividad es iniciada por primera vez.
- **onStart()** Indica que la Actividad va a ser desplegada al usuario.
- **onResume()** Es ejecutada cuando la actividad comienza a interactuar con el usuario.

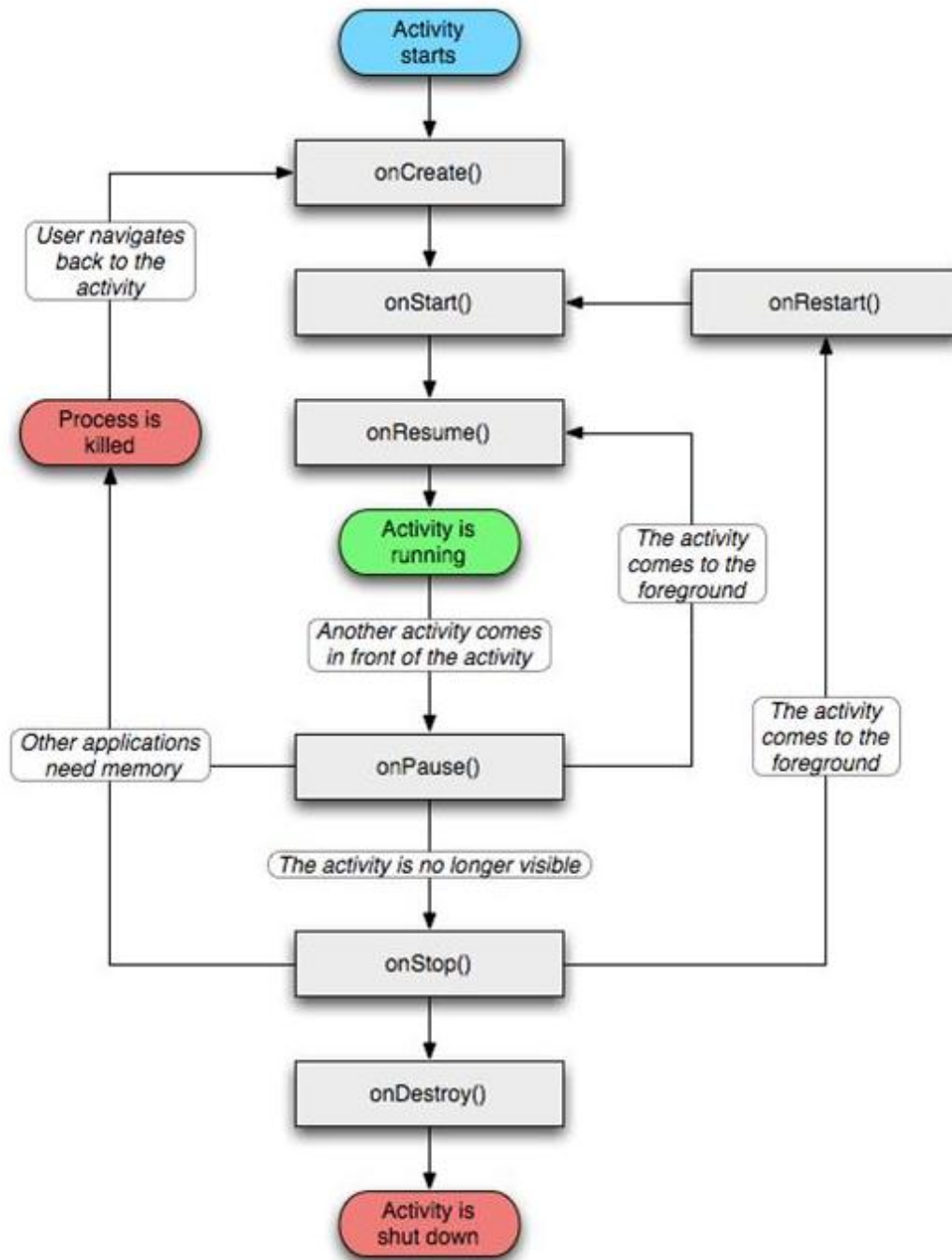


Figura 2.9: Ciclo de vida de un Actividad en Android

- **onPause()** Es invocada cuando la actividad va a pasar a un segundo plano.
- **onStop()** Es llamada cuando la actividad no esta visible al usuario y no se ocupara en un corto plazo.
- **onRestart()** Cuando este método es invocado, indica que la actividad será vuelta a mostrar al usuario desde un estado detenido.
- **ondestroy()** Es llamado antes de ser destruida la actividad.
- **onSaveInstanceState(Bundle)** El SO invocara esta método para salvar el estado de la interfaz de usuario, como la posición de un cursor.
- **onRestoreInstanceState(Bundle)** Es llamado cuando la aplicación esta siendo reinicializada desde un estado previamente guardado por el método *onSaveInstanceState(Bundle)*, por defecto restaura el estado de la interfaz de usuario.

Las Actividades no se ejecutan de una forma lineal y pueden ser detenidas o terminar el proceso de *Linux* que las almacena para dar espacio a más Actividades. Por ello es importante que la aplicaciones sean diseñadas considerando estos principios.

Capítulo 3

Diseño de la propuesta

En este capítulo se presenta la funcionalidad y la arquitectura empleada para el desarrollo del caso de estudio, que consistió en una aplicación de teléfono celular para infracciones de tráfico.

3.1. Funciones del Sistema

El sistema de cómputo utilizado en esta aplicación consistió en un cliente y un servidor. La aplicación cliente se instaló sobre un teléfono celular de marca HTC Hero con el Sistema Operativo Android 1.6. Las interfaces de comunicación utilizadas son Bluetooth, GPRS y 802.11 (WIFI). La aplicación del servidor se instaló en una computadora de escritorio con sistema operativo Linux. Se utilizó el framework Django 1.2.4 para responder las peticiones Web y como base de datos se empleo Postgres 8.4.

Las funciones genéricas que se llevan a cabo en esta aplicación son las siguientes:

- Iniciar la aplicación en el teléfono celular para realizar la multa de tráfico correspondiente.

- Llenar la forma con los datos necesarios para realizar la multa.
- Hacer contacto con el GPS para detectar la posición geográfica de la multa, a fin de anexar el mapa google a la forma de tráfico.
- Enviar la forma de multa al servidor.
- Verificar en el servidor si el automóvil es robado o si cuenta con alguna otra multa (pago de tenencia, otra multa de trafico, etc).
- Dar seguimiento a los teléfonos celular por su posición geográfica durante un día de trabajo.
- Dar seguimiento a las multas realizadas por cada policía, durante un día determinado de trabajo.
- Envío de las multas a las direcciones postales o a las direcciones de correo electrónico de los automovilistas.

Otros requerimientos funcionales adicionales son los siguientes:

- El envío de la información a través de la red debe ser seguro y contar con alguna forma de encriptación.
- La información enviada entre cliente y servidor no debe perderse aun cuando momentáneamente el cliente quede sin red. Para lo cual adicionalmente se requiere que sea posible acceder mas de un medio de comunicación.
- La información debe estar siempre respaldada y protegida a fin de que no se pueda perder o corromper. Se debe proporcionar una forma de tolerancia a fallos para el caso en que falle el cliente o el servidor.

Los requisitos no funcionales que debe cumplir el sistema son los siguientes:

- Bajo tiempo de respuesta por debajo de los 5 segundos por función.

- Alto tiempo de operación continua del cliente, por arriba de las 6 horas.
- Interface de cliente amigable y protegida en su acceso.
- El sistema debe ser portable para poder instalarse en otros tipos de aparatos celulares y ser expandible para soportar cambios en la tecnología.
- El sistema debe ser diseñado para ser mantenible.

3.2. Arquitectura del Sistema

La arquitectura del sistema esta planteada de forma general como se muestra en el diagrama 3.1. Se posee una serie de trabajadores de campo (Personal de Policía) con dispositivos móviles los cuales están recolectando información continuamente. Debe también existir un enlace web mediante WiFi o la red celular con un servidor conectado a la Internet con una preferencia por el uso del WiFi pues no con lleva un costo inherente el envío de la información. En caso de que no exista una conexión con el servidor la información debe almacenarse de forma local hasta que se posea de nuevo un enlace.

3.2.1. Casos de Uso

En el sistema propuesto se tienen tres actores principales, los trabajadores de campo, administradores y conductores. Estos actores llevan a cabo las siguientes funciones:

- **Trabajadores de Campo.**

Los trabajadores de campo, realizan la mayor actividad para la operación del sistema, como se puede apreciar en la figura 3.2. En el caso particular estos pueden llevar a cabo las siguientes funciones:

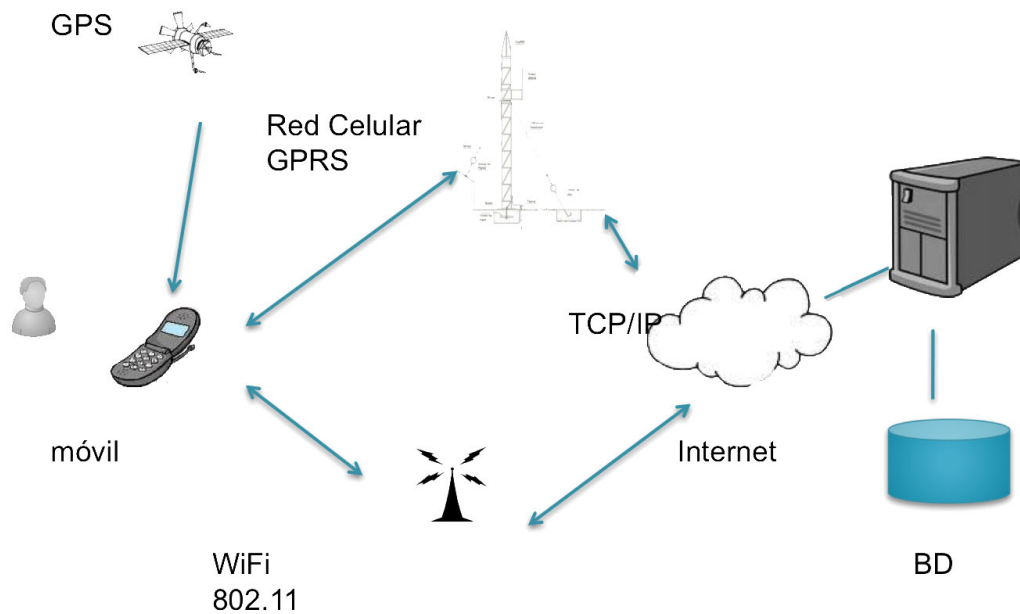


Figura 3.1: Arquitectura básica empleada para el caso de estudio

- Inicializa el sistema y registrar un nuevo evento (multa de tráfico).
- Obtener los datos del vehículo y del conductor.
- Comunicarse a un servidor de web mediante acceso a internet móvil.

■ Administradores.

Los administradores (como es mostrado en la figura 3.3) supervisan que la operación del sistema se efectúa de la forma adecuada y continua. También estos tienen como actividad examinar la consistencia de cada información recibida de los clientes. Otra actividad de los administradores del servidor es la obtención de estadísticas para checar tendencias.

■ Conductores.

EL tercer actor en el sistema es el conductor de un vehículo. Este actor solo proporciona información al sistema a petición del trabajador de campo. Además,

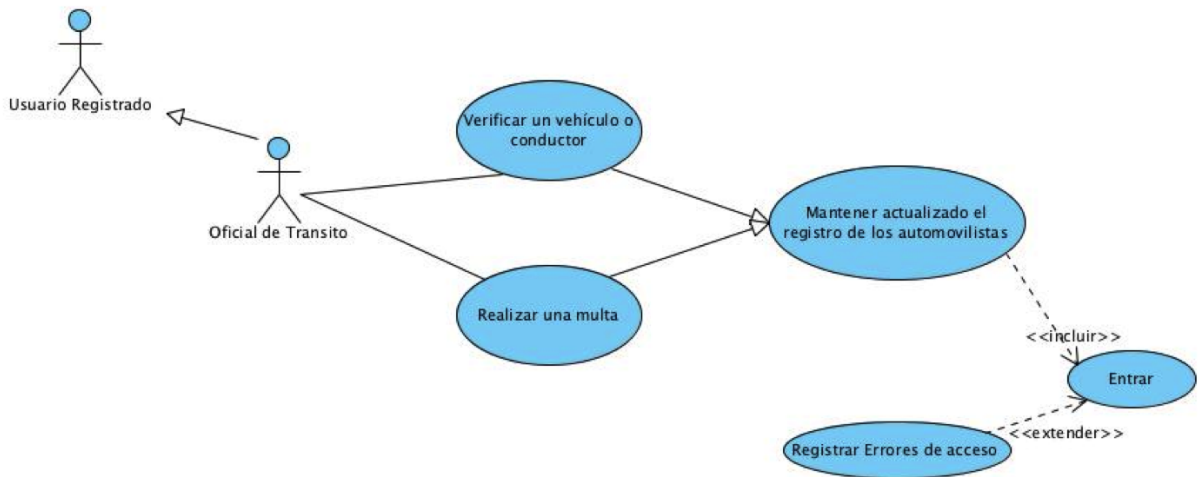


Figura 3.2: Caso de estudio para los trabajadores de campo

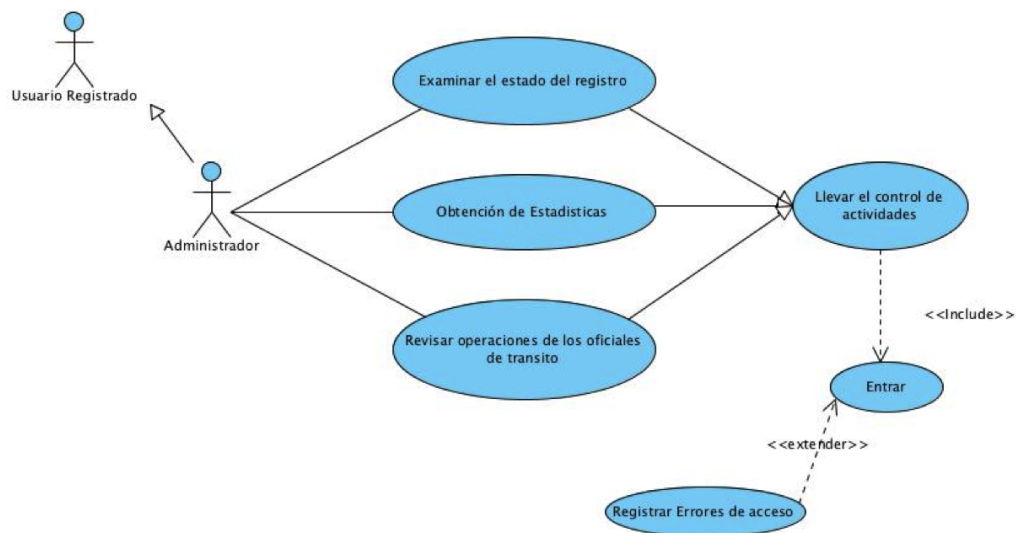


Figura 3.3: Caso de estudio para los administradores del sistema

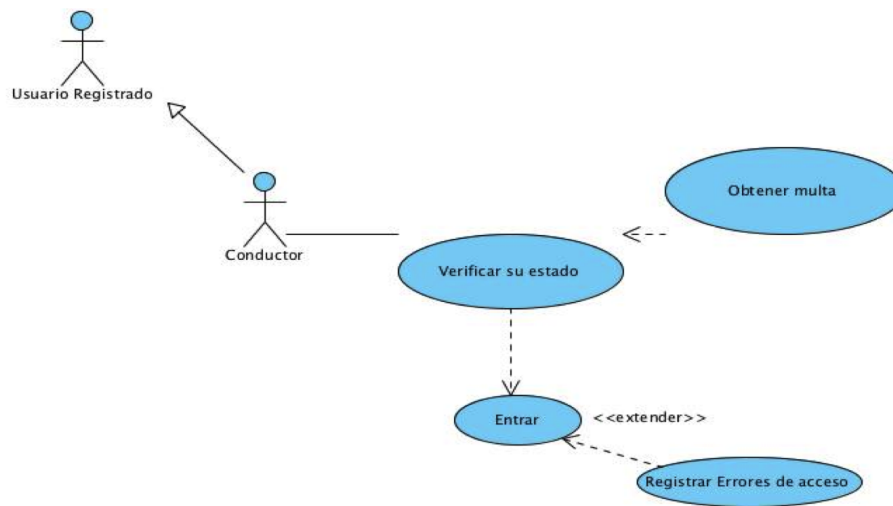


Figura 3.4: Caso del estudio para los conductores

reciben las multas de tráfico a sus domicilios o direcciones de correo electrónico, figura 3.4.

3.2.2. Diagramas de Secuencias

El cliente móvil (teléfono celular) lleva a cabo diversas funciones, como es la consulta de información, registro de eventos y el seguimiento de su posición a intervalos de tiempos regulares. Dichas funciones son explicadas de una forma más amplia a continuación:

- **Consulta de información.**

Como se muestra en el diagrama de secuencias 3.5, al requerir una cierta información el trabajador de campo, ingresara a la aplicación móvil introduciendo sus código de ingreso para autentificarlo. La aplicación verificará estos datos para permitir el acceso o negarlo. Así mismo el trabajador de campo rellena una formulario para efectuar una petición de multa. Esta información pasa de la actividad actual a un servicio parte del proceso de la aplicación para administrar la solicitud web. En el caso de que el móvil detecta que cuenta con conexión a la red (con lo cual el servidor es visible y alcanzable), se efectúa una petición

de consulta. En el caso contrario la aplicación móvil envía un mensaje de error al usuario.

Después de en enviar el mensaje el móvil se queda en espera de la respuesta para actualizar la pantalla en caso de encontrarse la debida información.

- **Registro de un evento.**

El efectuar un nuevo registro en el sistema como es el caso de una multa, al igual que en el caso anterior el usuario se loguea en la aplicación móvil y rellena un formulario para ingresar la multa, la información de la multa pasa a un servicio para administrar la solicitud web, la posición GPS y en caso necesario guardar los datos en una base de datos local hasta que exista una conexión con el servidor.

El caso de uso de registrar un evento involucra al cliente como es mostrado en el diagrama 3.6. También involucra al móvil de una forma más activa y el servidor almacenando la información generada. A diferencia de una consulta, al registrar un nuevo evento se debe poseer un mecanismo para almacenar la información de forma local en caso de existir un error en el canal de comunicaciones.

Al momento de recibir la información el servicio de la aplicación, todo el proceso siguiente se lleva en un segundo plano para seguir ejecutando la aplicación.

El servicio solicita al sistema la posición actual del GPS. Esta función puede tener un retraso considerable hasta obtener la primera coordenada dependiendo del primer tiempo de ajuste del GPS (TTFF), una vez obtenida la localización verifica si hay conexión a la red, si no es el caso se guarda en una base de datos local *SQLITE* y cada intervalo de tiempo el cual está definido a cinco minutos por defecto, se verifica la conexión para enviar los registros pendientes. Por el

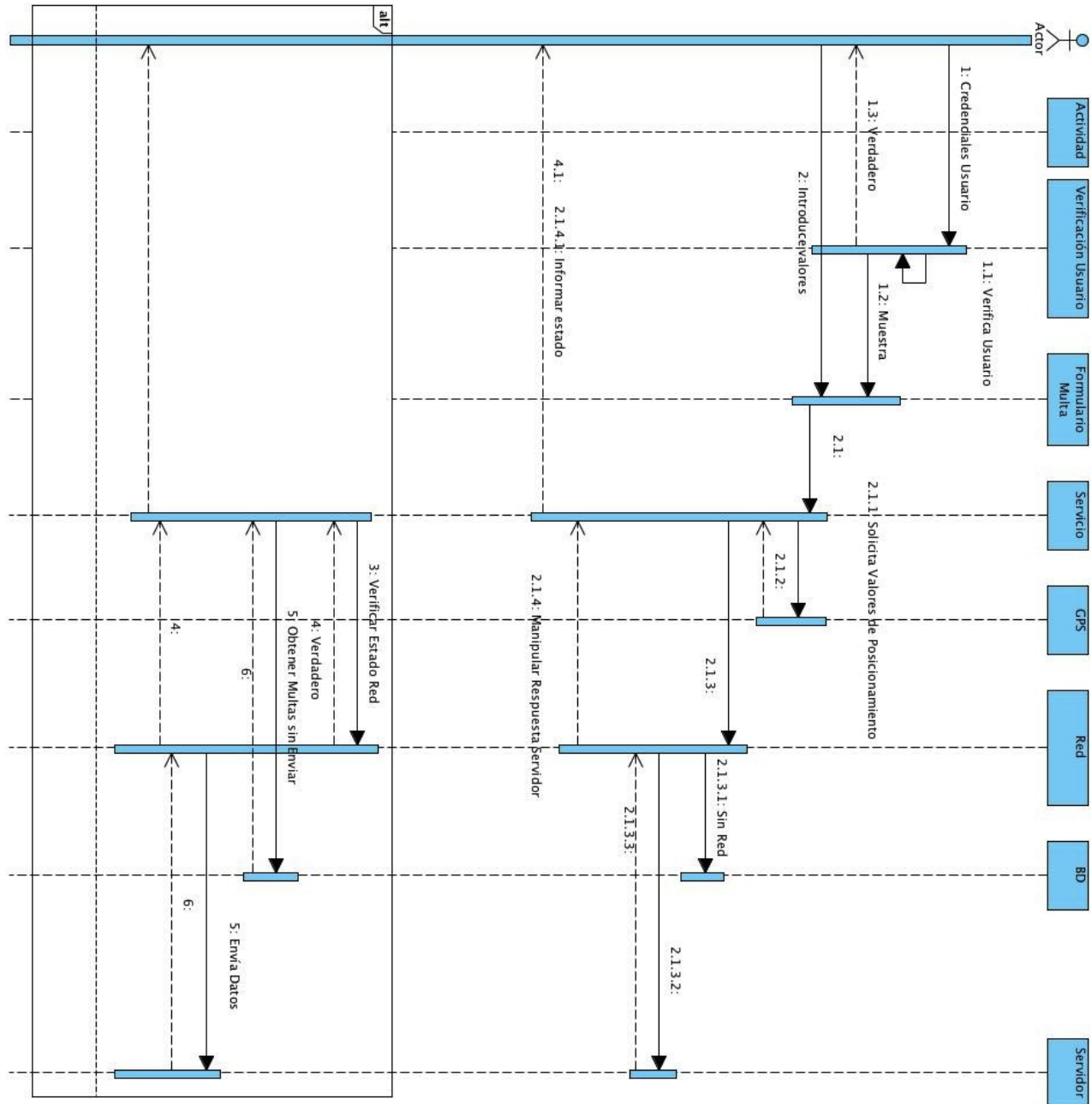


Figura 3.5: Diagrama de secuencias para la consulta de información

contrario si existe una conexión a la red, envía la información y se guarda en la base de datos local hasta que se confirme por parte del servidor su correcta recepción .

- **Seguimiento de posición.**

La tarea de seguimiento de posición de los usuarios siempre se ejecuta en un segundo plano sin una interfaz de usuario (UI) presente, cada intervalo de tiempo el cual tienen un valor de cinco minutos por defecto, se activa un evento que ejecuta el pedido de posición. El mecanismo de seguimiento de posición es muy similar al registro de un evento mencionado anteriormente, cada dato generado es guardado en la base de datos local hasta que se confirma su recepción por parte del servidor.

3.2.3. Módulos del Sistema

Para la ejecución del presente sistema se requirió el uso de tres módulos generales para su correcto funcionamiento : módulo de persistencia, módulo de conexión, módulo de seguridad. Los cuales permiten la correcta operación del sistema.

3.2.3.1. Módulo de Persistencia

Se entiende por persistencia de modo genérico, a la propiedad de los datos para que estos sobrevivan de alguna forma a pesa de pesar de cortes de energía. Salvar y recuperar datos es un requerimiento esencial de la mayoría de las aplicaciones. Sin esta capacidad, los datos se pierden al retirarse la corriente eléctrica.

En el caso del sistema de infracciones se presenta la persistencia por parte del cliente y servidor de dos formas: base de datos y “preferencias” por llave valor.

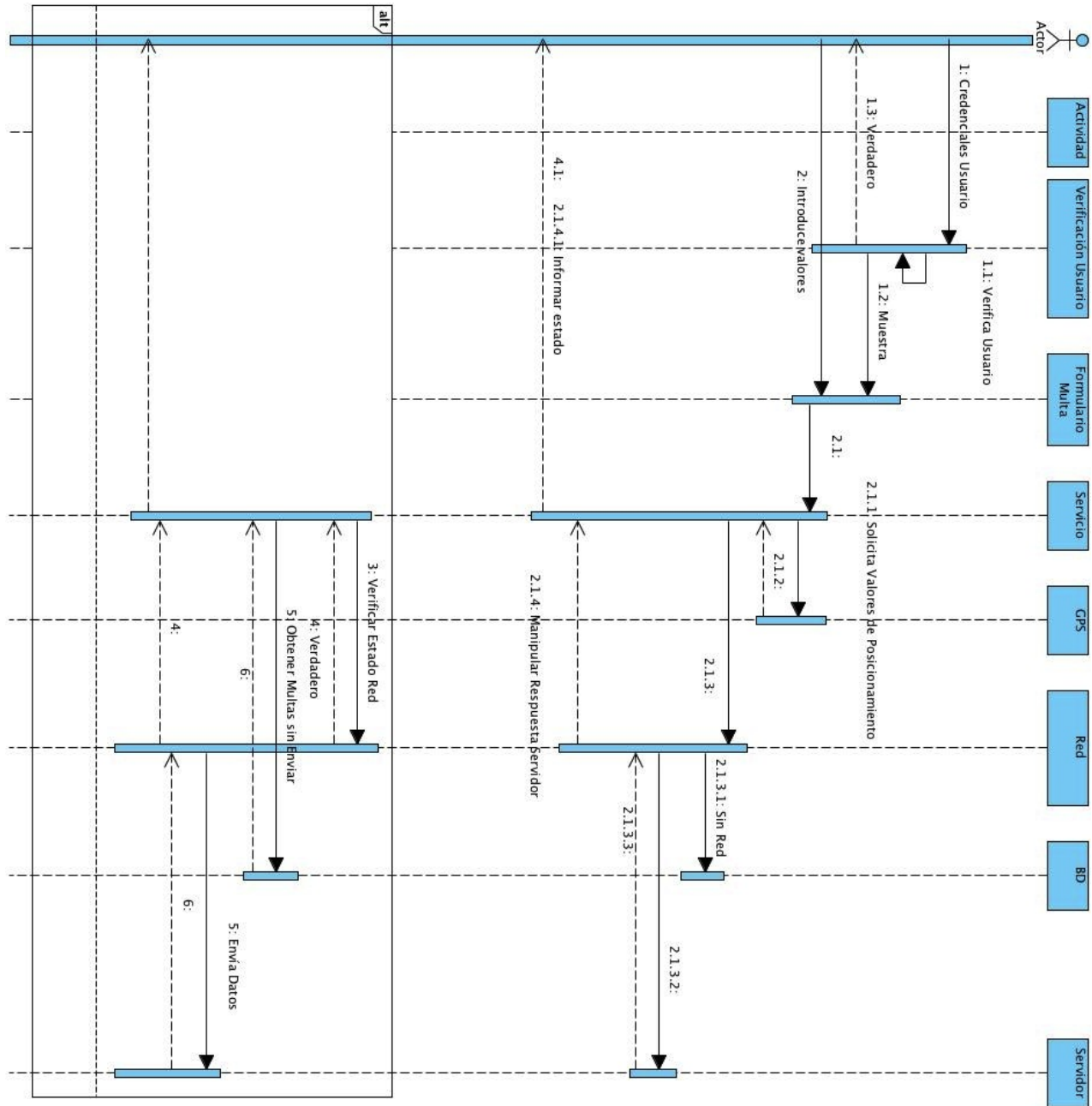


Figura 3.6: Diagrama de secuencias para la consulta de información

Del lado del cliente la persistencia es implementada en el sistema con dos mecanismos, “preferencias” llave-valor, el cual puede entenderse como una tabla hash que asocia llaves o claves con valores y una base de datos. Las “preferencias” llave-valor son empleadas para guardar configuraciones de ejecución de la aplicación, en la cual es posible guardar cualquier tipo de dato primitivo, sin embargo no se recomienda su uso para guardar datos genéricos de la aplicación. Además existe una base de datos local *Sqlite* para almacenar todos los eventos que son enviados al servidor y recuperarlos en caso de presentarse un error para ser enviados de nueva cuenta.

Por parte del servidor la persistencia se presenta con una base de datos *Postgres* la cual almacena los eventos enviados por los clientes para su posterior manipulación.

Al iniciarse la aplicación móvil, lee de las “preferencias” la configuración actual para realizar un correcto funcionamiento, como es la contraseña actual, la URL del servidor para realizar peticiones, periodo de reenvío de la información en caso de error en la comunicación, etcétera.

Al enviar un evento sensible el cliente lo almacena en una base de datos local y es guardado hasta que se reciba una confirmación por parte del servidor de su correcta recepción. En caso de presentarse un error o no recibir una confirmación del servidor, se posee un proceso ejecutándose en segundo plano el cual verifica cada cierto intervalo de tiempo, cinco minutos es el lapso por defecto, si existen eventos por enviar y efectuar su reenvío en caso de ser necesario.

Por parte del servidor se tiene conectada una base de datos, al recibir un nuevo evento por parte de los diversos se verifica que no se posea anteriormente en la base de datos comparando las “estampillas de tiempo” del evento entrante contra los almacenados en la Base de Datos, de existir una repetición se descarta y almacena en una bitácora de eventos.

3.2.3.2. Módulo de Conexión

La comunicación entre el cliente y el servidor esta basada en el protocolo de comunicación TCP/IP, y e protocolo de aplicación http/https.

Al ser una aplicación planeada para ejecutarse en cualquier parte de la ciudad, es necesario tener la capacidad de comunicarse por redes GPRS y WiFi y una forma de conmutar entra ambas redes cuando sea necesario.

La mayor parte del tiempo se plantea que la aplicación se comunique sobre redes GPRS las cuales proveen una gran movilidad, sin embargo presentan un ancho de banda muy limitado y traen consigo un costo inherente. Por ello al encontrarse sobre este tipo de redes sólo se debe enviar la mínima información necesaria para el funcionamiento del sistema. Al situarse sobre una red Wifi conocida se puede hacer el envío de toda la información no vital que complemente la funcionalidad de la aplicación, como es el caso de contenido multimedia (p.ej. fotografías que demuestran el hecho registrado). Dicho envío del contenido multimedia, se ejecuta en un segundo plano pues la existencia de redes WiFi conocidas a las cuales conectarse de forma segura son pocas.

3.2.3.3. Módulo de Seguridad

Teniendo datos sensibles en la operación de la aplicación es necesario contar con un mecanismo de protección de los datos. De no contar con esta protección la información podría ser leída por terceros no autorizados, bloquear la información no llegando al servidor, o en un peor escenario manipular y generar datos falsos.

De forma local las contraseñas del sistema son guardas en forma a través de su

función hash o digesto. Una función hash es una función de una sola vía que genera un resultado teórico único con una entrada única, al comparar estos resultados se puede comprobar el correcto ingreso de las contraseñas. De no tomar esta medida la aplicación del sistema puede recaer en terceros y ser decompilada mostrando las contraseñas en claro como parte del código fuente.

En la aplicación desarrollada el canal de comunicaciones entre el cliente y el servidor es cifrado por Https mediante RSA, utilizando un certificado por parte del servidor SSL, lo cual permite por parte de los clientes autenticar a ambos actores al establecer una comunicación.

3.3. Flujo de Información del Sistema

El sistema de infracciones esta dividido en dos partes principalmente, un servidor y diversos clientes móviles los cuales están en comunicación con el servidor. Estos clientes deben ser capaces de soportar la posible intermitencia en el canal de comunicaciones. A continuación se describe con mayor detalle cada una de las funciones de ambos.

3.3.1. Cliente

Si se tiene una solución basada en un navegador, existe el problema de tener una conexión a internet desigual que puede original una disponibilidad intermitente de la aplicación, por ello se opto por el uso de dispositivos celulares móviles los cuales tienen varias ventajas los cuales incluyen: servicios basados en geolocalización, imágenes de la cámara y acelerómetros. Estos servicios se pueden combinar con datos disponibles en linea para ofrecer una mejor experiencia para el usuario.

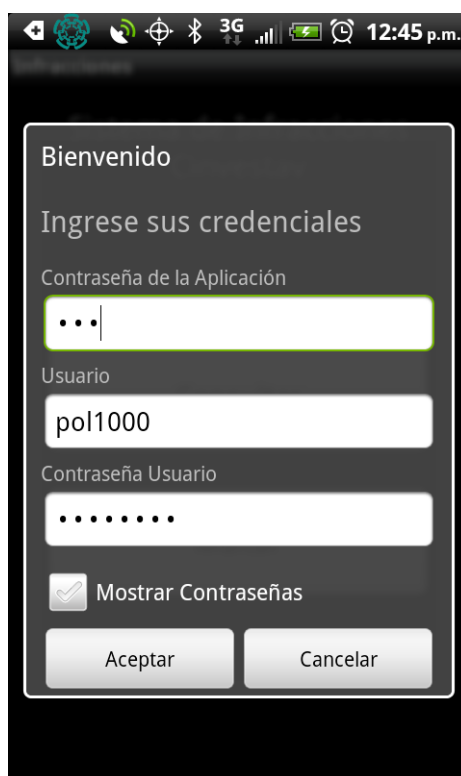


Figura 3.7: Pantalla para ingresar al sistema de Infracciones

Por la parte del cliente, para el desarrollo de la aplicación propuesta se llevan a cabo tres tareas, rastreo de los usuarios, consulta de datos y envío de multas.

Cualquier usuario que desee utilizar el sistema de infracciones deberá ingresar al sistema con una contraseña autenticada como se muestra en la figura 3.8. La primera contraseña es para permitir el acceso a la aplicación, las credenciales siguientes son empleadas al identificarse ante el servidor para realizar cualquier operación.

El usuario de la aplicación puede realizar dos actividades de forma activa, consultar un dato o realizar un evento, como se muestra en la figura 3.8

Al ingresar a las opciones de la aplicación, este se presenta con una serie de parámetros configurables,(mostradas en la figura 3.9), como son:

- La URL del servidor.
- El intervalo de espera para el envío de la información en caso de error.

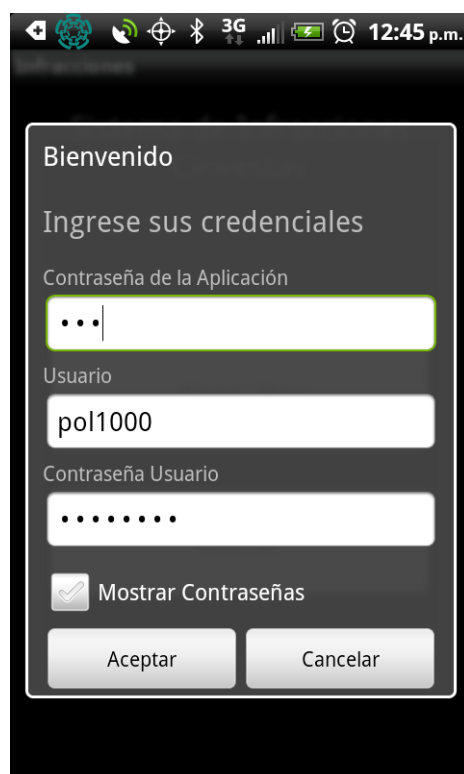


Figura 3.8: Pantalla para de opciones del sistema de infracciones

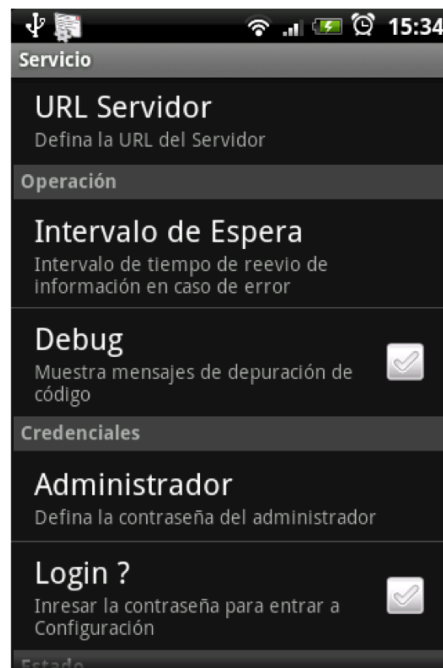


Figura 3.9: Pantalla menu del sistema de infracciones

- Una opción de debug (depuración) que muestra opciones adicionales y mensajes extras.
- Redefinir la contraseña del administrador para entrar a este menú así con la opción de inhabilitarla.

3.3.1.1. Consulta de Información

Durante cada día de operación del sistema, existe la necesidad de consultar diversa información como es el estado de un vehículo o un conductor para tener más elementos y aplicarlos en una correcta acción.

Para obtener la información deseada, el oficial de tránsito, rellena un breve formulario y la información es enviada al servidor.

La forma en la que opera este módulo de consulta, como se muestra en la figura

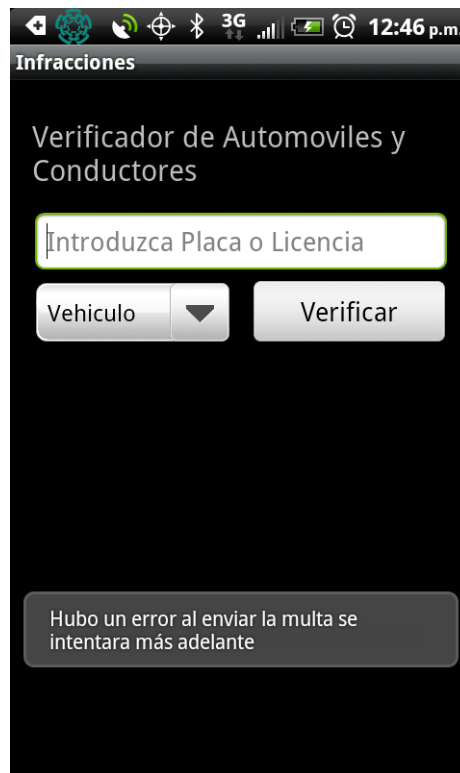


Figura 3.10: Pantalla de consulta de un vehículo o conductor del sistema de infracciones

3.10 consiste primeramente en la verificación del estado de la red. Si se detecta una conexión se envía la petición y se espera por su respuesta. En caso contrario simplemente manda un mensaje de error.

3.3.1.2. Envío de Multas

Al realizar una multa se rellena un formulario como se muestra en la figura 3.11, la cual consta de la identificación del conductor, el número de seria de vehículo, el tipo de multa por la que se registra el evento, una descripción breve del suceso y una fotografía que será enviada en cuanto se posea conexión a la WiFi. Una vez relleno el formulario, este pasa a segundo plano para obtener la coordenada del GPS junto con el envío seguro de la información.



The screenshot shows a mobile application interface for reporting a traffic violation. The screen is titled "Infracciones" and contains several input fields: "Vehículo" with a text box containing "Placa", "Conductor" with a text box containing "Licencia", and "Multa" with a dropdown menu showing "5". Below these fields is a "Fotografía" button, followed by a "Descripción:" label and a large text area containing the prompt "Mencione brevemente los hechos ocurridos." At the bottom is a "Multar" button.

Figura 3.11: Pantalla para efectuar una multa del sistema de infracciones

3.3.1.3. Rastreo de Usuarios

Como parte complementaria del sistema, la aplicación envía al servidor su posición actual, cada intervalo de tiempo dado. De esta forma conoce la ubicación de cada usuario en un tiempo dado para localizarlo en caso de ser necesario, y se puede hacer un seguimiento a través del tiempo de su posición.

3.3.2. Servidor

El servidor del sistema de infracciones está dividido en dos secciones principalmente una parte "web móvil" para la interacción con los clientes móviles, y una "web tradicional" para la consulta de la información de los datos recabados por los trabajadores de campo.

3.3.2.1. Servidor Web-Móvil de Infracciones

Esta parte del servidor escucha todas las peticiones de los clientes móviles y atiende todos los eventos generados. Se tiene un portal web el cual está a la escucha de peticiones Https mediante el método POST. Al recibir una petición este verifica las credenciales. En caso de ser errónea guarda un registro de las peticiones equivocadas, dependiendo si se trata de una consulta o una multa efectúa la operación necesaria. Y respondiendo al cliente con la información solicitada en caso de pedirlo o solo un *acknowledge* indicando que la operación se llevo correctamente

3.3.3. Sistema Web de Infracciones

Para poder visualizar la información recolectada por los trabajadores de campo, se emplea un portal Web como se muestra en la figura 3.12. En este portal se muestran todos los eventos generados en un mapa para tener su fácil ubicación. Dentro del portal se pueden hacer una serie de tareas como obtener multas por número de policía, conductor o vehículo y mostrarlos en un mapa, y el rastreo de un trabajador de campo durante un día dado.

Asimismo en este portal se puede obtener una serie de estadísticas que permiten tener un mayor control. Por ejemplo en el sistema se pueden detectar los puntos con mayor incidencia de multas, o el tipo de multa que se efectúa más frecuentemente, en otras estadísticas.



Figura 3.12: Portal Web del sistema de infracciones

Capítulo 4

Pruebas y Resultados

Con el fin de revisar la funcionalidad de sistema propuesto, se presentan una serie de pruebas sobre distintos factores de la aplicación. Entre estos factores están el tiempo de respuesta, la duración de la pila batería, el tiempo para obtener la primera posición del satélite y la prueba de cada función de la aplicación.

Para probar el sistema se utilizo el siguiente escenario: La aplicación cliente se instalo sobre un teléfono celular HTC Hero con SO Android 1.6, con con interfaces de comunicación GPRS y 802.11 (WIFI), la aplicación servidora se instalo en una computadora de escritorio con sistema operativo Linux, con el framework Django 1.2.4 para responder las peticiones Web y como base de datos de empleo Postgres 8.4.

Para verificar la correcta operación se realizaron las siguientes pruebas:

1. Tiempos de Respuesta sobre Redes WiFi y GPRS
 - Sockets TCP
 - Sockets UDP
 - Peticiones GET
 - Peticiones POST

2. Duración de la batería del cliente móvil
3. Primer tiempo de Ajuste del GPS
4. Uso del sistema sobre un ambiente de semi-producción

4.1. Desempeño del Sistema

4.1.1. Tiempos de Respuesta sobre Redes WiFi y GPRS

El objetivo de estas pruebas es determinar el tiempo que transcurre al efectuar la comunicación entre el cliente móvil y el servidor, empleando diferentes medios de acceso a Internet como son las redes WiFi ó GPRS y empleando diferentes protocolos de comunicación.

Para llevar a cabo las pruebas de tiempo, por parte del cliente se desarrollo una aplicación en Android que ejecuta un número solicitado de veces una petición sobre diversos protocolos. Esto se llevo a cabo con el fin de determinar el protocolo más conveniente sobre el cual de desarrollar posteriormente la aplicación.

Dichas pruebas se desarrollaron variando el tamaño de los datos a enviar, desde 128 a 32K bytes incrementando el tamaño de estos en potencias de dos. Esto con el fin de comprobar el comportamiento de respuesta con comandos extensos.

4.1.1.1. Sockets TCP

En esta prueba la aplicación cliente crea un socket TCP entre el móvil y el servidor enviando datos de un tamaño constante, cierra el socket y comienza de nuevo. En esta prueba se mide el tiempo transcurrido de envío de la información desde el cliente hasta el momento de la recepción del *acknowledge* por parte del servidor. Esta prueba se repitió 1,000 veces sobre las redes WiFi y GPRS respectivamente.

Los datos obtenidos se muestran en la tabla 4.1 y el gráfico ?? para los tiempos de respuesta con una conexión a la red WiFi del móvil.

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	6	26.15	14	10	225.38	9039
256	6	28.15	15	10	188.88	9035
512	6	19.35	13	10	90.21	3527
1024	6	29.7	14	14	288.85	22812
2048	7	25.13	13	11	163.2	9022
4096	8	23.76	14	13	90.62	5452
8192	12	53.94	22	17	265.39	12023
16383	19	90.65	47	29	392.42	21480
32384	56	247.83	196	203	309.91	11287

Tabla 4.1: Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión WiFi

Es posible observar en la figura 4.1 que el tiempo de respuesta permanece por debajo de los 100ms aun a los 4KBytes, y llega a su tope máximo (200 ms) con 32 KBytes. Estos resultados son considerados aceptables, pues desde la percepción del cliente un retardo de este valor no es perceptible.

Las tabla 4.2 y la figura 4.2 muestran los tiempos de respuesta sobre la una red celular GPRS.

A diferencia de la red WiFi, la red GPRS presenta un grado de lentitud mayor, donde por ejemplo al realizar un envío de información de 2 KBytes posee un retardo de casi medio segundo. Este retardo es el límite máximo recomendado en aplicaciones de servicios web.

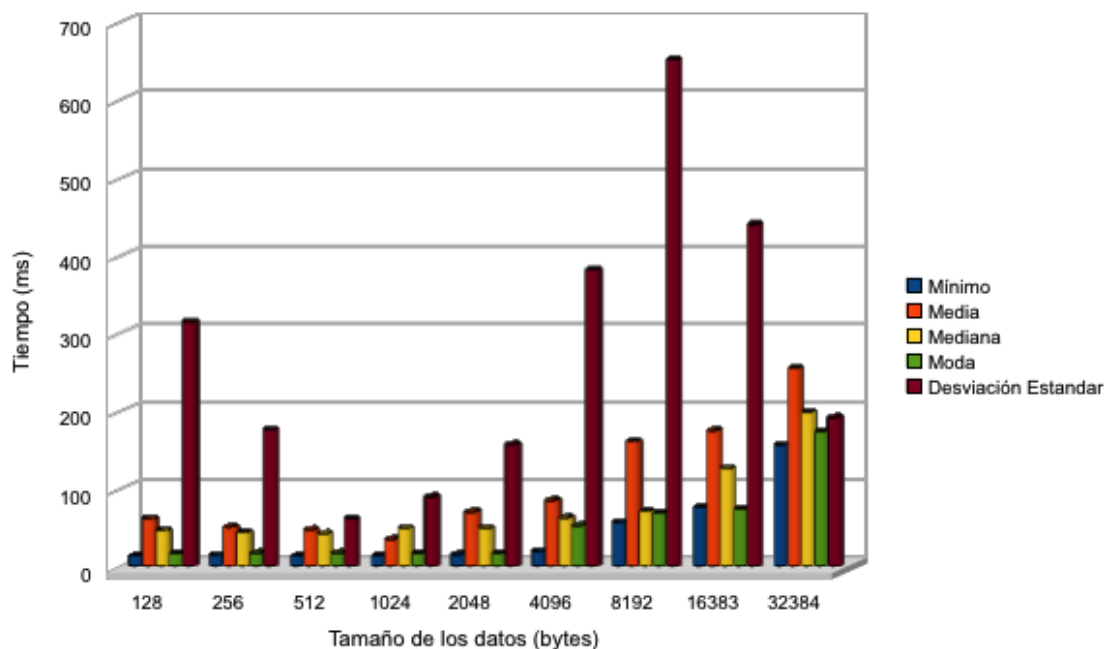


Figura 4.1: Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión WiFi

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	174	420.66	251.5	213	528.66	3489
256	189	464.54	277	245	945.63	4510
512	189	497.69	291	452	558.99	4239
1024	236	525.24	326	266	871.24	4918
2048	318	766.07	478.5	414	588.22	10172
4096	566	1099.99	875.5	791	961.01	10481
8192	783	1404.58	1212.5	995	840.44	5663
16383	834	1682.55	1386.6	1148	759.78	26156
32384	1074	2672.43	2096	2052	1520.74	37886

Tabla 4.2: Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión GPRS

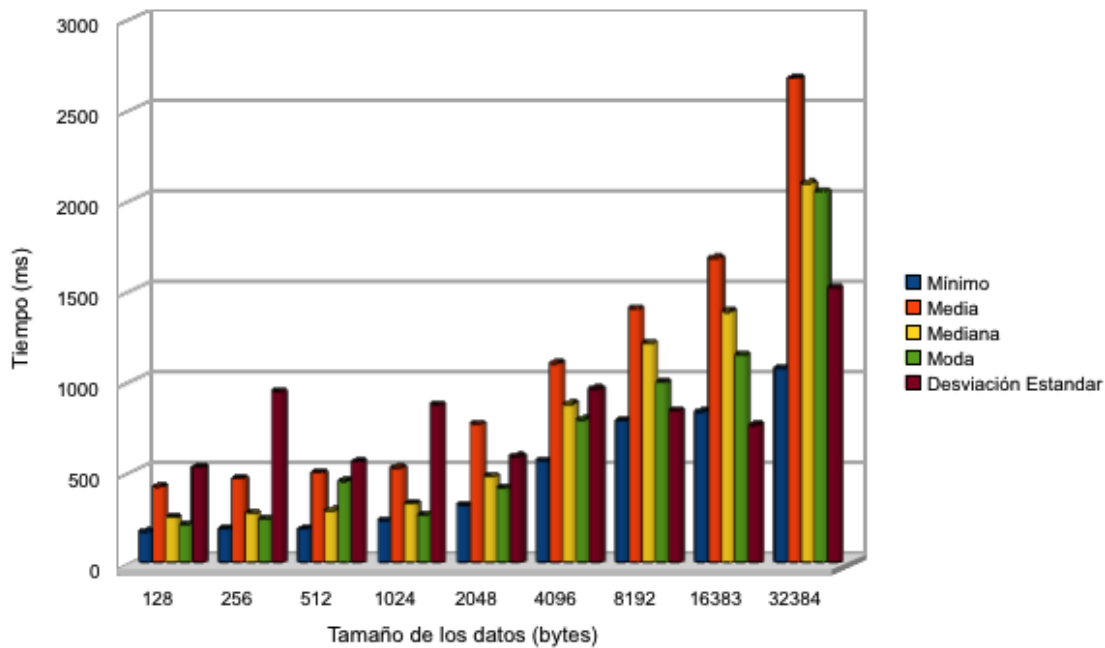


Figura 4.2: Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión GPRS

4.1.1.2. Sockets UDP

El aplicación cliente crea un socket UDP entre el móvil y el servidor enviando datos de un tamaño constante, cierra el socket y comienza de nuevo. En esta prueba se mide el tiempo transcurrido de transmisión de información desde el cliente al hasta la recepción del *acknowledge* por parte del servidor. Esta prueba se repitió 1,000 veces sobre las redes WiFi y GPRS.

Aunque los tiempos de respuesta (mostrados en la figura 4.3) son mucho menores incluso a las del socket TCP, se tuvo una perdida de la información bastante considerable alrededor del 80%, efecto que no ocurrió con los socket TCP.

Las tabla 4.4 y la figura 4.4 muestran los tiempos de respuesta sobre la una red celular GPRS.

La figura ?? muestra los resultados de la prueba ejecutando con sockets UDP y

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	1	1.94	2	2	2.88	48
256	1	2.22	2	2	4.14	59
512	1	1.43	1	1	3.68	38
1024	1	1.79	1	1	4.93	37
2048	1	2.7	1	1	6.55	45
4096	1	5.08	2	1	9.88	44
8192	1	11.89	2	2	14.16	50
16383	3	33.13	32	32	4.35	128
32384	58	63.62	64	64	7.84	166

Tabla 4.3: Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión WiFi

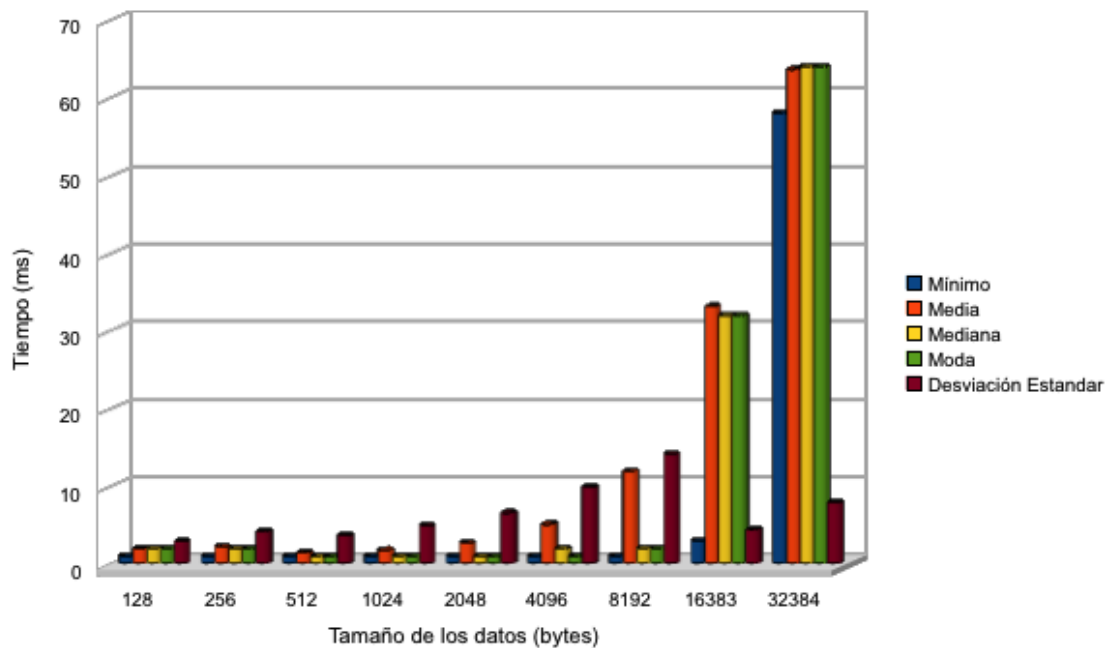


Figura 4.3: Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión WiFi

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	158	461.66	233.5	214	505.66	3209
256	194	501.54	278	235	957.63	4118
512	188	482.69	293	437	597.99	4254
1024	233	505.24	326	270	853.24	5271
2048	334	842.07	485.5	425	590.22	10787
4096	588	1057.99	930.5	790	882.01	10563
8192	728	1502.58	1138.5	950	886.44	5408
16383	907	1566.55	1249.6	1125	799.78	27194
32384	994	2529.43	2065	2257	1666.74	39429

Tabla 4.4: Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión GPRS

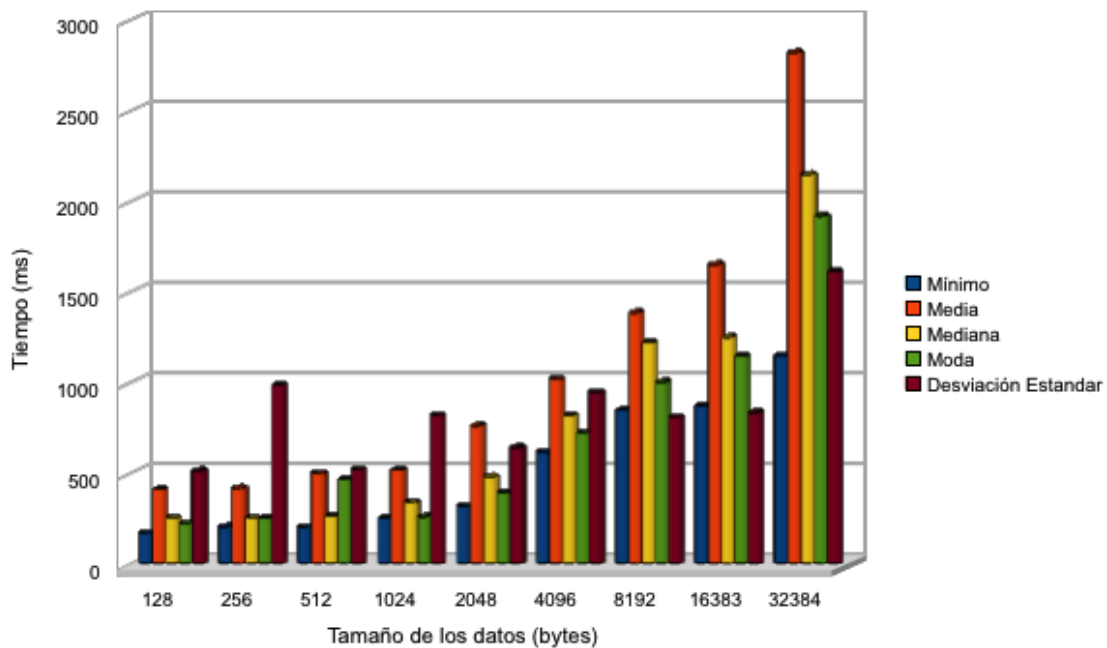


Figura 4.4: Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión GPRS

una conexión a la GPRS. Es posible observar que en esta prueba se obtiene un tiempo bajo en comparación de la prueba en WiFi. Sin embargo la pérdida de la información aumenta bastante (cerca de 70 %).

4.1.1.3. Peticiones GET

En esta prueba el cliente ejecuta una petición Http tipo *Get* con el servidor, enviando datos de un tamaño constante. La prueba consiste en la medición del tiempo transcurrido en la transmisión de información desde el cliente hasta la recepción del *acknowledge* por parte del servidor. Esta prueba se realizó 1,000 veces sobre las redes WiFi y GPRS.

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	14	62.13	47	17	313.33	9074
256	14	54.95	47	18	183.25	4678
512	14	47.91	47	19	67.68	1791
1024	14	40.65	49	18	93.1	1774
2048	16	66.55	54	19	154.92	2378
4096	20	95.29	61	57	362.4	9049
8192	58	156.9	77	70	716.51	12205
16383	73	184.85	123	78	464.21	6867
32384	152	263.54	206	187	202.89	3292

Tabla 4.5: Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión WiFi

Como se aprecia en la figura 4.5 los tiempos de retardo hasta los 8 KBytes de datos esta por debajo de los 100 (ms) y a 32 KBytes casi supera los 200 ms. Esto

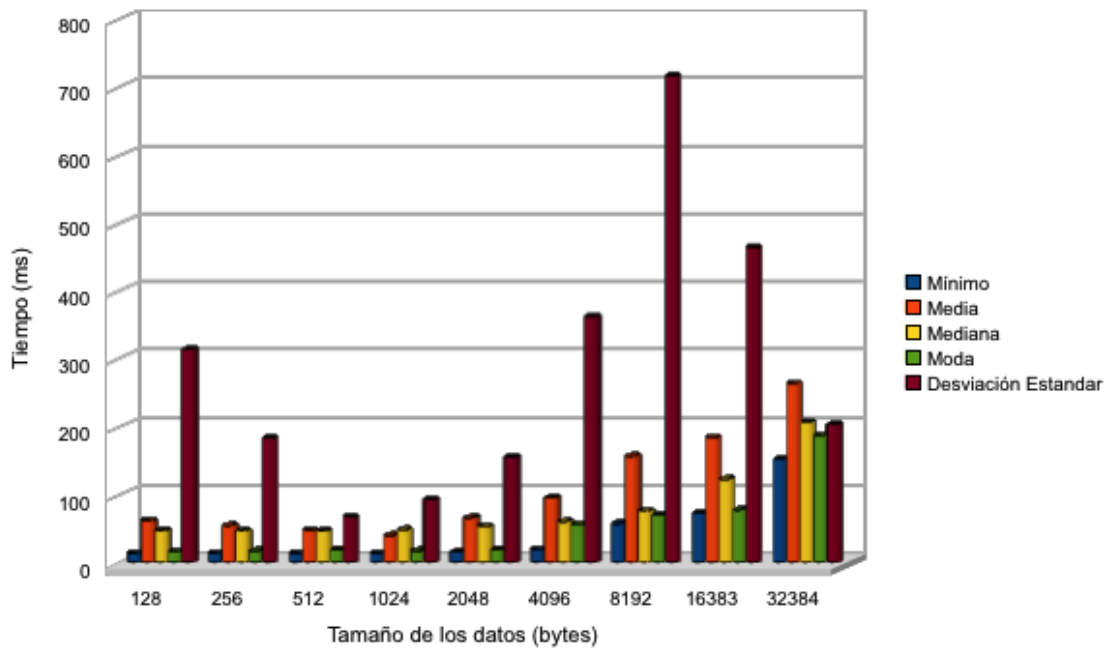


Figura 4.5: Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión WiFi

indica unos valores bastante bajos proporcionando una respuesta casi inmediata para un usuario.

Las tabla 4.6 y la figura 4.8 muestran los tiempos de respuesta sobre la una red celular GPRS.

En la figura ?? se muestran los tiempos de respuesta de una petición get sobre una red GPRS. Es posible observar que hasta los 2 KBytes de información se obtiene un tiempo de respuesta por debajo de los 500 milisegundos.

4.1.1.4. Peticiones POST

En esta prueba la aplicación cliente ejecuta una petición Http tipo *Post* con el servidor, enviando datos de un tamaño constante. La prueba consiste en la medición del tiempo en la transmisión de información desde el cliente hasta el momento de la

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	166	441.66	242.5	191	558.66	3418
256	196	480.54	258	264	1014.63	4955
512	177	489.69	321	433	613.99	3933
1024	240	535.24	336	280	867.24	5104
2048	330	699.07	452.5	444	601.22	9782
4096	518	1084.99	922.5	735	894.01	10451
8192	823	1436.58	1191.5	1008	863.44	5717
16383	916	1527.55	1251.6	1179	734.78	23783
32384	1037	2607.43	2018	2113	1525.74	36784

Tabla 4.6: Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión GPRS

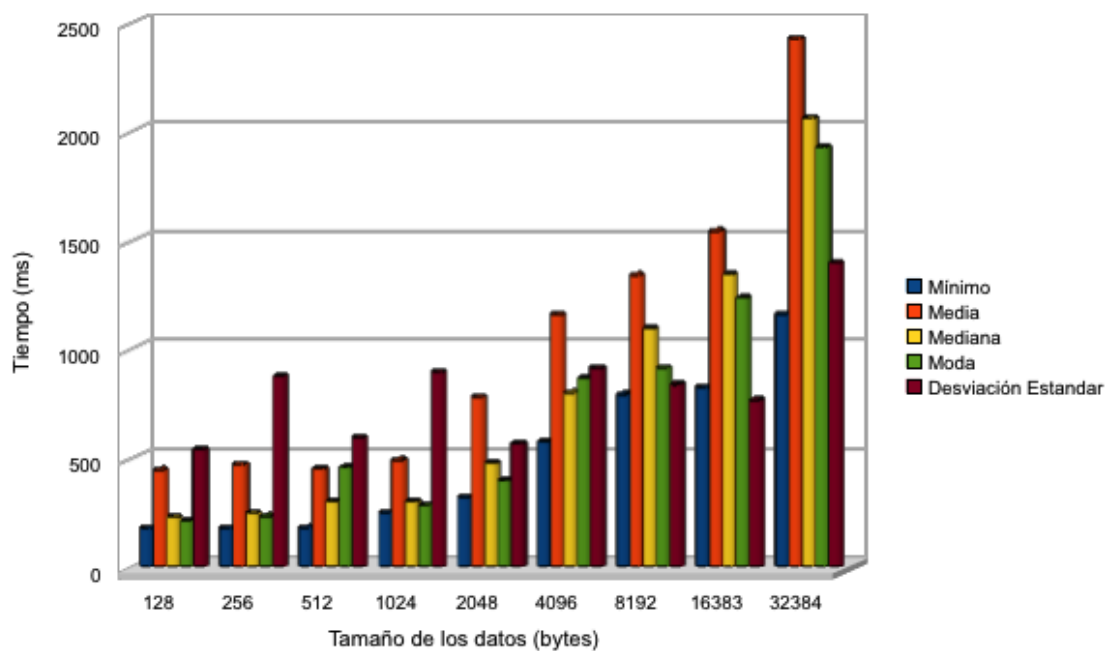


Figura 4.6: Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión GPRS

recepción del *acknowledge* por parte del servidor. Esta prueba se realizó 1,000 veces sobre las redes WiFi y GPRS.

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	14	63.13	45	18	283.33	9163
256	15	51.95	45	19	172.25	4541
512	14	43.91	42	17	61.68	1854
1024	14	43.65	45	19	90.1	1668
2048	16	70.55	56	17	145.92	2356
4096	20	92.29	56	53	334.4	8222
8192	57	140.9	79	64	695.51	11818
16383	76	192.85	114	74	463.21	7108
32384	156	241.54	205	172	209.89	3361

Tabla 4.7: Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión WiFi

En la figura ?? se observan los tiempos de respuesta de la prueba de petición POST sobre una red WiFi. Es posible observar que en esta prueba se obtiene un tiempo de respuesta aceptable contra el que obtiene el servidor, con tiempos promedio menores a los 500 ms.

Las tabla 4.8 y la figura 4.8 muestran los tiempos de respuesta sobre la una red celular GPRS.

En la figura ?? se muestran los tiempo de respuesta de la prueba de petición POST, presentando tiempos por debajo de los 500 ms, hasta los 2 KBytes de información.

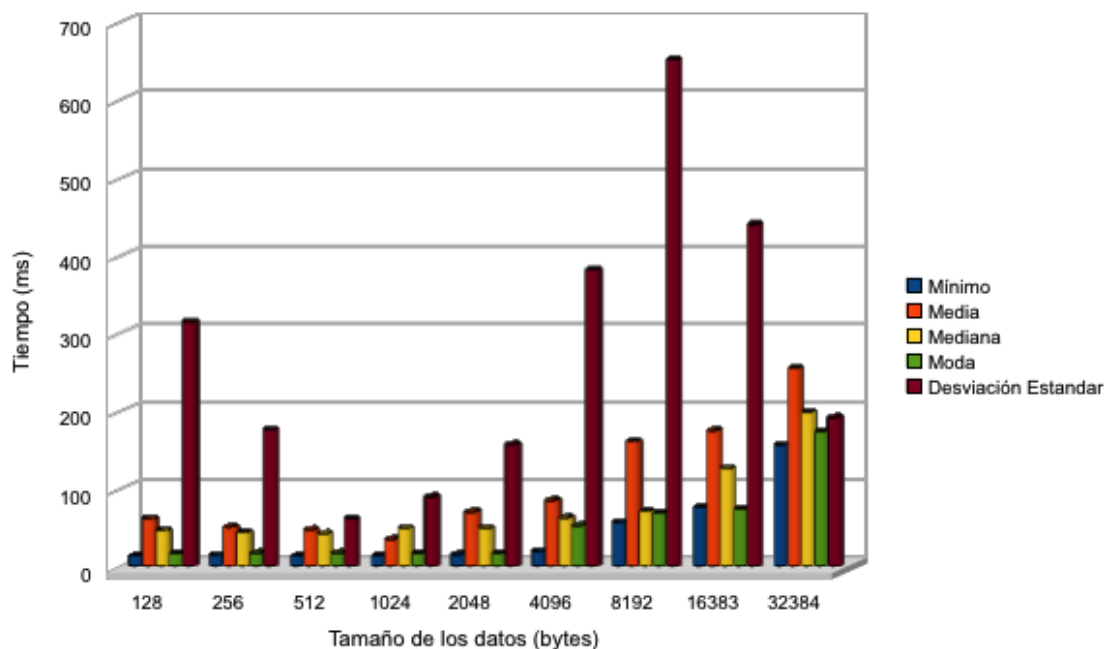


Figura 4.7: Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión WiFi

Tamaño de los Datos (bytes)	Mínimo (ms)	Media (ms)	Mediana (ms)	Moda (ms)	Desviación Estándar (ms)	Máximo (ms)
128	178	445.66	230.5	211	539.66	3230
256	179	471.54	250	233	874.63	4418
512	180	450.69	300	457	594.99	3876
1024	250	490.24	299	284	896.24	4578
2048	321	780.07	475.5	395	566.22	10242
4096	575	1159.99	799.5	865	913.01	10032
8192	792	1339.58	1096.5	914	837.44	6041
16383	824	1543.55	1342.6	1236	765.78	28759
32384	1162	2427.43	2063	1929	1398.74	37692

Tabla 4.8: Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión GPRS

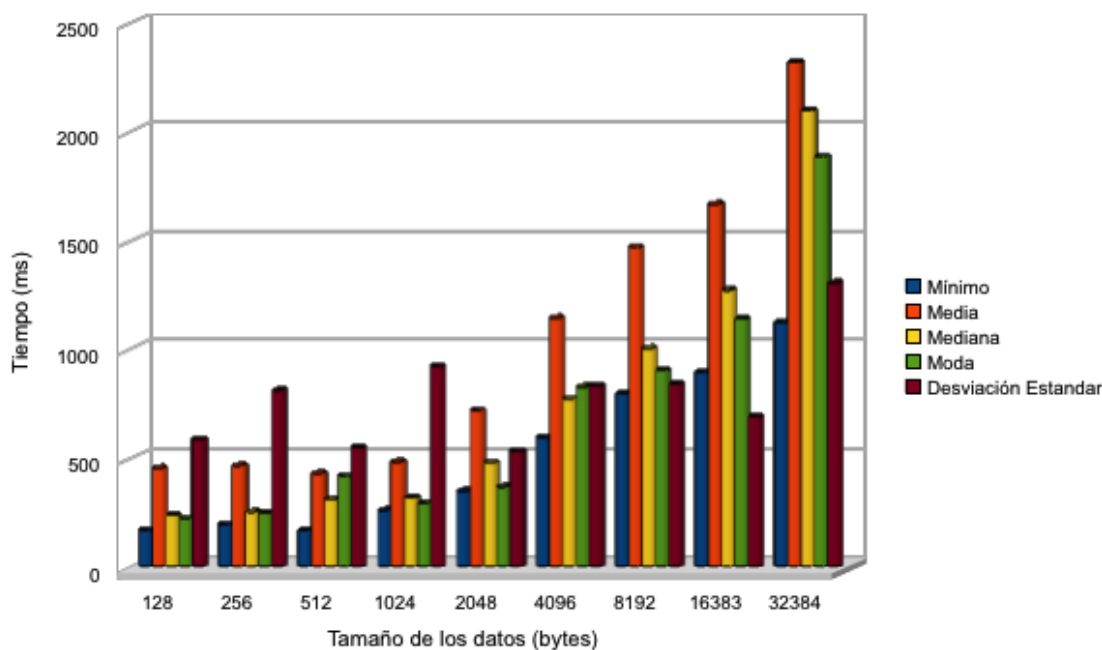


Figura 4.8: Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión GPRS

4.1.1.5. Elección del Protocolo del Comunicaciones

Después de la realización de las pruebas anteriores se tomo la decisión de utilizar Http Post por las siguientes razones:

- Aún cuando los sockets UDP presentan tiempos de respuesta bajos, tienen un alta probabilidad de pérdida de información, lo cual no puede ser considerado, tratandose de una aplicación con información sensible.
- Se optó por una mayor normalización al tener tiempos de respuesta similares entre una petición con un socket TCP y un petición Http.
- Una petición Http Get mayor a los 2 KBytes suele tener problemas. Cuando se corta la información o en HTTP 1.1 da un error 414 indicando que hay una URL demasiado larga.

4.1.2. Duración de la Batería del Cliente Móvil

Para hacer uso de este sistema en una aplicación real es necesario conocer la duración de la batería ejecutando operaciones propias del sistemas desarrollado (por ejemplo consulta y registro de nuevos eventos), con el fin de asegurar que el móvil soporta toda una jornada de trabajo.

En esta prueba se dejó ejecutando de forma continua un cliente móvil el cual enviaba eventos de multas al servidor sobre la red GPRS. Los resultados se muestran en la tabla 4.9.

Mínimo	Media	Mediana	Moda	Desviación Estandar	Máximo
02:12:00	03:48:00	03:43:00	03:34:00	00:27:00	04:24:00

Tabla 4.9: Tiempo de vida de la batería efectuando registro del eventos en el eventos (multas)

Como se muestra en la figura 4.9 el teléfono llega a permanecer en operación en promedio 3 horas y 30 minutos , lo cual es bueno si se tiene en cuenta que se lograron realizar en aproximadamente 200 eventos o multas, lo cual es considerado un rango mayor al de una jornada de trabajo.

4.1.3. Primer Tiempo de Ajuste del GPS

En esta prueba se realiza el cálculo del tiempo de ajuste de la señal del GPS. Al solicitar la primera señal del GPS, se tiene un tiempo de espera para ajustar el GPS conocido como primer tiempo de ajuste. Si estos tiempo de ajuste inicial son muy largos, ocurrirá que no tienen un *retardo* extra en el sistema por parte de la espera del GPS.

En base a los resultados mostrados en la gráfica 4.9 y la Tabla 4.10 podemos observar que el tiempo necesario para obtener la primera señal del satélite, es casi un

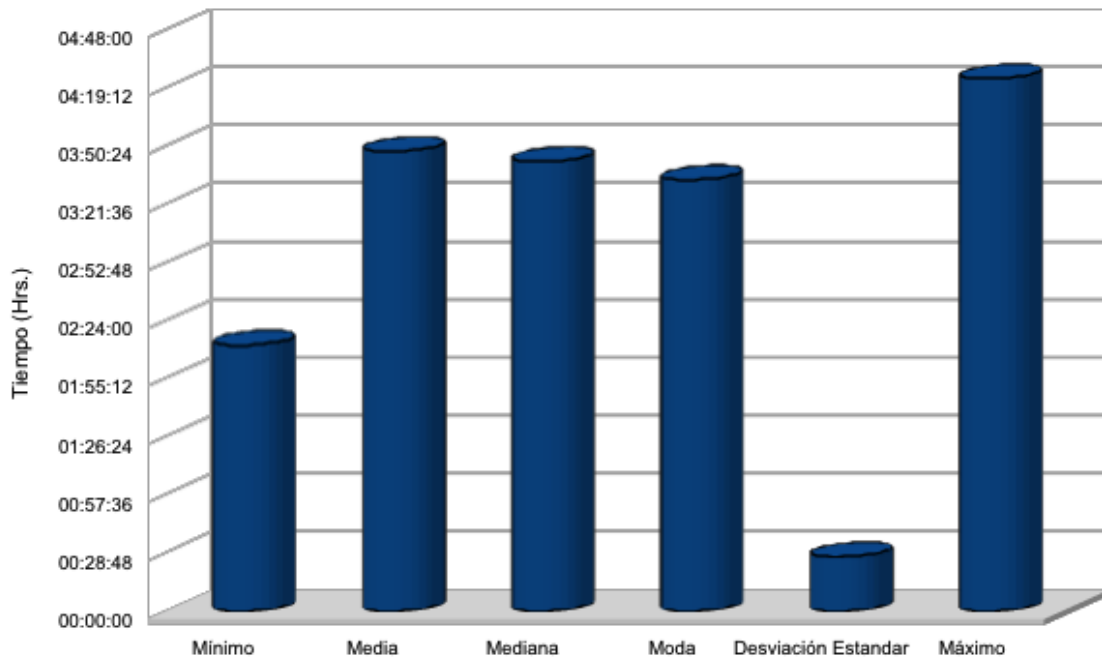


Figura 4.9: Tiempo de vida de la batería efectuando registro del eventos en el eventos (multas)

Mínimo	Media	Mediana	Desviación Estándar	Máximo
41	59.24	63.5	14.39	82

Tabla 4.10: Primer Ajuste para obtención de señal del GPS

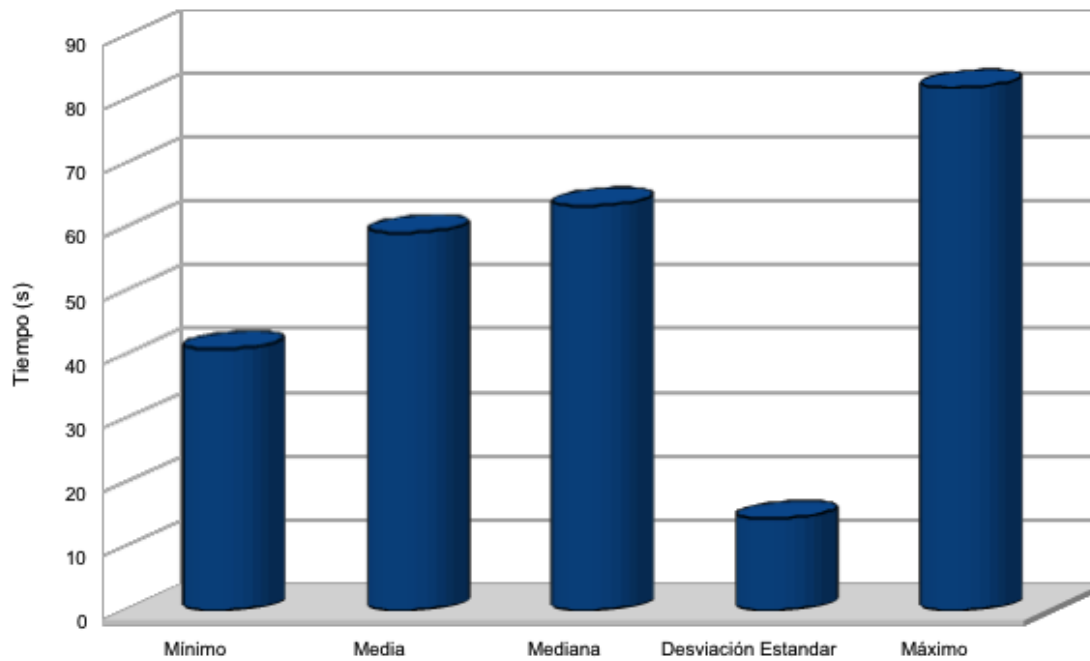


Figura 4.10: Primer Ajuste para obtención de señal del GPS

minuto lo cual se sabe que es un rango normal para móviles.

4.1.4. Uso del Sistema

Esta prueba se realizó para comprobar la funcionalidad de la aplicación en un caso real. Como demostración del uso del sistema, se realizó una prueba de uso sobre una avenida de la ciudad realizando diversos eventos desde un cliente móvil.

La primera prueba fue la medición del tiempo de respuesta sobre las operaciones típicas del sistema: consultas y multas, sobre la red GPRS en el entorno de la ciudad, mostrados en la tabla 4.11 y en la figura 4.11

Operación	Mínimo	Media	Mediana	Moda	Desviación Estándar	Máximo
Consulta	78	201.17	116.54	88	273.54	3713
Multa	171	436.45	237	192	521.34	4691

Tabla 4.11: Tiempo de respuesta para operaciones típicas del sistema de infracciones

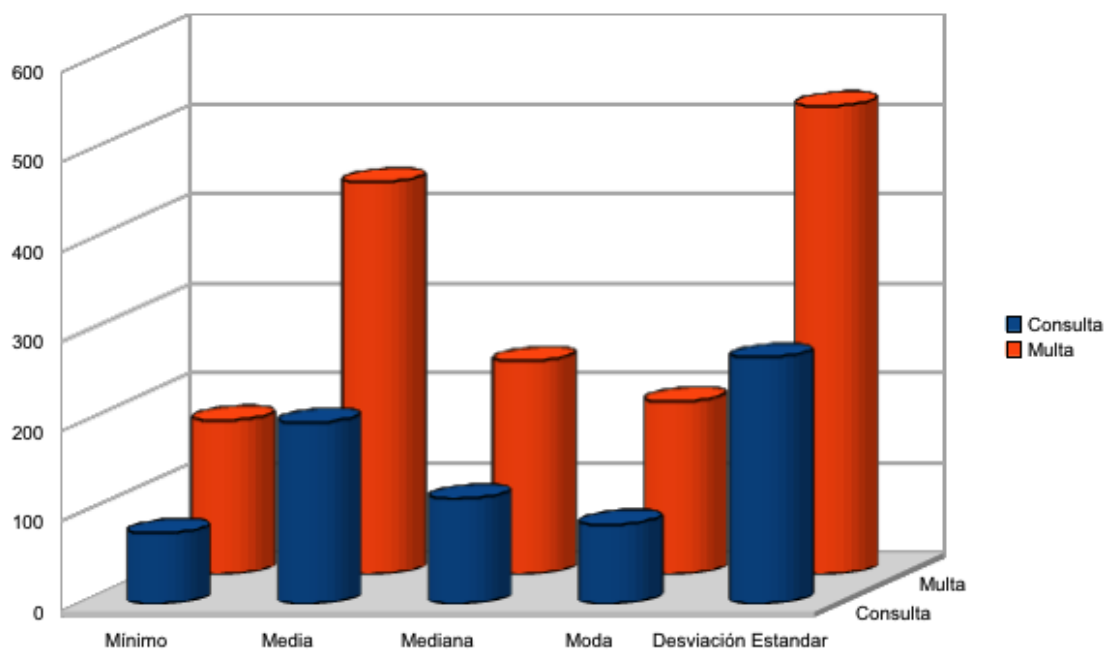


Figura 4.11: Tiempo de respuesta para operaciones típicas del sistema de infracciones

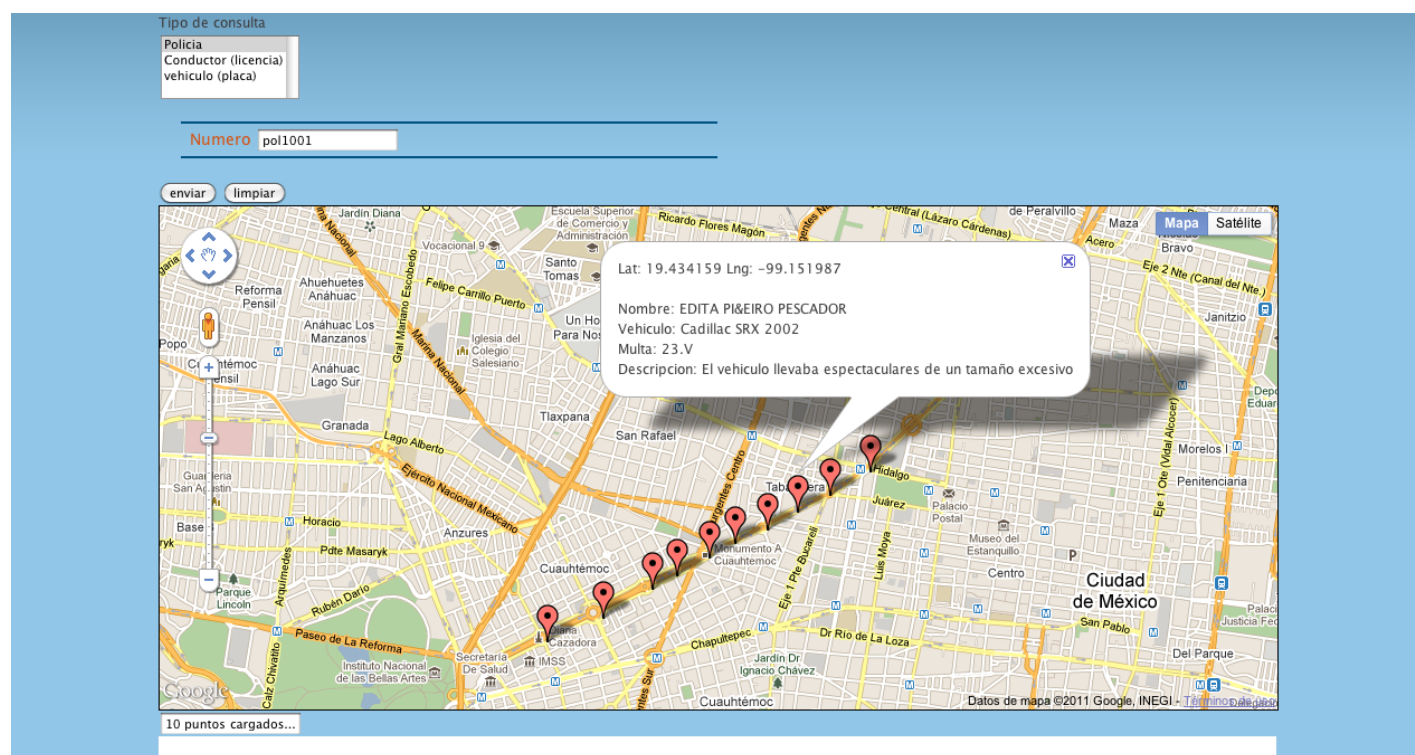


Figura 4.12: Despliegue en el servidor de una prueba de uso del sistema sobre la ciudad

4.1.4.1. Registro de Eventos

El registro de eventos de esta prueba se realizó en cada cuadra de una avenida del DF, en donde se realizo un evento(multa) para dar un total de 10 registros.

En el cliente se guardó un bitácora, con todos los detalles de la multa y el tiempo en el que se realizó dicho evento. Estos eventos con todos los datos recolectados se envían al servidor para ser visualizados, el resultado de la captura de dichos eventos se muestra en la figura 4.12.

Capítulo 5

Conclusiones y trabajo futuro

El uso de teléfonos celulares se ha incrementado de forma considerable en los últimos años. Existiendo más personas que usan un celular a una computadora y convirtiéndose en un elemento multifuncional que ha ido reuniendo características propias de varios dispositivos. Tal es el caso de las agendas electrónicas, reproductores multimedia, video juegos portátiles, navegadores web, etcétera. De tal forma que da la impresión que funciones básicas como hacer una llamada telefónica están en un segundo plano, gracias a las características actuales de los móviles. Esto da lugar al lugar al desarrollo de aplicaciones complejas haciendo tareas desde cualquier lugar.

El uso de las comunicaciones se ha vuelto una parte fundamental en la parte de decisiones de cualquier entorno, es por ello que el sistema mostrado en el presente trabajo de tesis ayuda en parte a poseer la información prácticamente al momento de ser generada en un ambiente donde se tenga una serie de trabajadores por toda una región geográfica recolectando datos.

5.1. Conclusiones

En esta tesis se presento una revisión sobre las diferentes arquitecturas móviles predominantes en el mercado, destacando sus ventajas y desventajas y haciendo no-

tar la razón por la cual se utilizó la arquitectura en Android para el desarrollo del presente trabajo. Entre los factores que fueron fundamentales para esta elección fue el hecho de ser una arquitectura *Open Source* basado en un Linux 2.6 lo cual ayuda, en caso de requerirlo acceder al *kernel* y modificarlo. La codificación del aplicaciones para Android se lleva sobre Java 5 lo cual reduce la curva de aprendizaje.

Se desarrollaron una serie de módulos sobre Android los cuales proporcionan diferentes características como son: localización, persistencia local y comunicación a la red. Todos estos módulos fueron desarrollados de forma independiente y se unieron posteriormente al desarrollo del sistema de infracciones.

Se desarrolló una arquitectura para el presente sistema, descrito en el Capítulo 3, el cual permite la implantación de los módulos antes mencionados. La arquitectura está compuesta de dos elementos principalmente: la parte de los clientes móviles recolectando datos y la parte de un servidor capaz de atender todas las peticiones de los móviles. El elemento que más interactúa en el sistema es el cliente, pues en el se realiza la carga de los datos para realizar un evento, la recolección de datos de localización, la verificación del estado de la conexión a la red, el envío de la información y el respaldo de la información.

Para verificar el funcionamiento del sistema de infracciones, se desarrollaron una serie de pruebas como son:

- **Tiempo de respuesta.** Se desarrollo esta prueba para conocer el retardo presente sobre los diferentes protocolos comunes y sobre diferentes tipos de accesos a la red. Se optó por realizar comunicaciones por Http POST debido permite enviar una mayor cantidad de información respecto a otros protocolos como es Http GET, presenta poca perdida de información.

En segunda lugar se dejo observar que realizar una comunicación sobre una red celular es considerablemente más lento que un enlace por WiFi, de tal forma que

enviar mas de 2 KBytes de información implica un tiempo de retardo cercano a los 500 ms, lo cual es el máximo permitido para una tener una respuesta sin mayores retrasos para el usuario.

- **Duración de la batería del cliente móvil.** Al encontrarse desarrollando trabajo de campo sin la posibilidad de recarga el móvil durante un jornada, es necesario que este tenga una duración de la pila lo suficientemente largo. La prueba desarrollada en la sección 4.1.2 muestra una duración aproximada de 3:30 horas ejecutando más de 200 eventos en promedio. Lo cual es considerado aceptable para el caso de uso propuesto, en el cual en un día un usuario no genera más de 100 eventos.
- **Primer Tiempo de Ajuste del GPS.** El primer tiempo que para que pueda proporcionar datos el GPS, según la prueba descrita en la sección 4.1.3 se requiere un tiempo promedio de un minuto, lo cual es un tiempo relativamente bueno considerando que en la aplicación desde que se registra un evento, se solicita información de la posición.
- **Uso del sistema.** La funcionalidad de la aplicación descrita en la sección 4.1.4 se verificaron los los tiempos promedios para realizar una consulta con 100 ms y una multa con 200 ms.

5.2. Trabajo Futuro

Los trabajos futuros que se proponen como complemento a este trabajo de tesis son los siguientes:

- Una forma de optimizar la duración de la batería, sería poder desactivar la interfaz de red de forma automática cuando no se este ocupando y activarlo cuando sea necesario. Asimismo que el cliente pueda estar a la espera por si llegara un mensaje de texto en caso que el servidor desee iniciar una comunicación con el

cliente.

- Otro trabajo futuro que podría hacerse a partir de esta tesis es el uso de criptografía de llave pública, esto permitiría entre otras cosas el uso de firmas digitales para aquellas aplicaciones que lo requieran.
- Un tercer trabajo futuro sería mejorar al servicio de transferencia de archivos, un mecanismo de compresión de datos, con el objetivo de reducir los tiempos de transmisión, costos y aprovechando de una forma más eficiente el ancho de banda.

Índice de Tablas

4.1. Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión WiFi	63
4.2. Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión GPRS	64
4.3. Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión WiFi	66
4.4. Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión GPRS	67
4.5. Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión WiFi	68
4.6. Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión GPRS	70
4.7. Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión WiFi	71
4.8. Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión GPRS	72
4.9. Tiempo de vida de la batería efectuando registro del eventos en el eventos (multas)	74
4.10. Primer Ajuste para obtención de señal del GPS	75
4.11. Tiempo de respuesta para operaciones típicas del sistema de infracciones	77

Índice de Figuras

2.1. Dimensiones extras del cómputo móvil	8
2.2. Estructura celular de red GSM	13
2.3. Arquitectura del Sistema GSM	15
2.4. Arquitectura del sistema GPRS	19
2.5. Componentes básicos de un sistema operativo	23
2.6. Arquitectura del Sistema Operativo Symbian	30
2.7. Arquitectura del Sistema Operativo iOS	33
2.8. Arquitectura del Sistema Operativo Android	35
2.9. Ciclo de vida de un Actividad en Android	39
3.1. Arquitectura básica empleada para el caso de estudio	44
3.2. Caso de estudio para los trabajadores de campo	45
3.3. Caso de estudio para los administradores del sistema	45
3.4. Caso del estudio para los conductores	46
3.5. Diagrama de secuencias para la consulta de información	48
3.6. Diagrama de secuencias para la consulta de información	50
3.7. Pantalla para ingresar al sistema de Infracciones	54
3.8. Pantalla para de opciones del sistema de infracciones	55
3.9. Pantalla menu del sistema de infracciones	56
3.10. Pantalla de consulta de un vehículo o conductor del sistema de infracciones	57
3.11. Pantalla para efectuar una multa del sistema de infracciones	58

3.12. Portal Web del sistema de infracciones	60
4.1. Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión WiFi	64
4.2. Tiempos de respuesta utilizando del servidor empleando sockets TCP con una conexión GPRS	65
4.3. Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión WiFi	66
4.4. Tiempos de respuesta utilizando del servidor empleando sockets UDP con una conexión GPRS	67
4.5. Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión WiFi	69
4.6. Tiempos de respuesta utilizando del servidor empleando un servicio Http Get con una conexión GPRS	70
4.7. Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión WiFi	72
4.8. Tiempos de respuesta utilizando del servidor empleando un servicio Http Post con una conexión GPRS	73
4.9. Tiempo de vida de la batería efectuando registro del eventos en el eventos (multas)	75
4.10. Primer Ajuste para obtención de señal del GPS	76
4.11. Tiempo de respuesta para operaciones típicas del sistema de infracciones	77
4.12. Despliegue en el servidor de una prueba de uso del sistema sobre la ciudad	78

Bibliografía

- [1] Christian Bettstetter et al. “GSM Phase 2 General Packet Radio Service GPRS: Architecture, Protocols and Air Interface”. En: *IEEE Communications Surveys* (1999).
- [2] Steve Babin y Richard Harrison. *Developing Software for Symbian OS*. John Wiley & Sons Ltd, 2006, pág. 11.
- [3] ED Burnette. *Hello, Android.: Introducing Google’s Mobile Development Platform*. Pragmatic Bookshelf, 2009.
- [4] Android Dev. *Arquitectura Android*. 2011. URL: <http://developer.android.com/guide/basics/what-is-android.html>.
- [5] Ahmed El-Rabbany. *Introduction to GPS: The Global Positioning System*. Artech house, 2002.
- [6] Matthew Gast. *802.11 Wireless Networks The Definitive Guide*. O’Reilly, 2005.
- [7] *GSM Overview*. 2010. URL: <http://www.gsmworld.com>.
- [8] Ann McIver McHoes Ida M. Flynn. *Understanding Operating Systems*. Brooks/Cole Publishing Company, 2000, pág. 349.
- [9] Apple inc. *iOS Architecture*. 2010. URL: <http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSOverview/iPhoneOSOverview.html>.

-
- [10] Apple inc. *iPhone OS Overview*. 2000. URL: http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/URL_iPhone_OS_Overview/index.html.
- [11] Digia Inc. *Programming for the Series 60 Platform and Symbian OS*. John Wiley & Sons Ltd, 2003, págs. 1,6.
- [12] I. Mahgoub M. Ilyas. *Mobile Computing Handbook. 1st. ed.* Cambridge University Press, 2005.
- [13] Dave Mark y Jeff LaMarche. *Beginning iPhone 3 Development*. Apress, 2006, pág. 11.
- [14] Reto Meier. *Android 2 Application Development*. Wronx, 2010, pág. 543.
- [15] Saúl Eduardo Pomares Hernández. *Computación Ubicua; un gran desafío*. INAOE, 2009.
- [16] Chin-Chun Lee Raymond Steele y Peter Gould. *cdmaOne and 3G Systems*. John Wiley & Sons Ltd, 2001, pág. 65.
- [17] Morgan Stanley. *Internet Trends*. 2010. URL: http://www.morganstanley.com/institutional/techresearch/internet_trends042010.html.

Los abajo firmantes, integrantes del jurado para el examen de grado que sustantará el **Sr. Carlos Jonathan Ferreyra Rodríguez**, declaramos que hemos revisado la tesis titulada:

TITULO DE LA TESIS

Y consideramos que cumple con los requisitos para obtener el Grado de Maestría en Ciencias de la Computación.

Atentamente,

Dra. Xiaou Li

Dr. César Torres Huitzil