



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	v
Abstract	vii
1. Introducción	1
2. Conceptos generales	5
2.1. Cómputo Móvil	6
2.2. Cómputo Ubicuo	6
2.3. Comunicación Móvil	7
2.3.1. El sistema GSM	7
2.3.2. GPRS	8
2.3.3. Redes inalámbricas 802.11	9
2.4. GPS	10
2.5. Sistemas Operativos para dispositivos móviles	11
2.5.1. Symbian OS	12
2.5.2. iPhone OS	13
2.5.3. Android	14
2.5.3.1. Arquitectura	15
2.5.3.2. Componentes de una aplicación	18
2.5.3.3. Ciclo de vida de una aplicación	19

3. Diseño de la propuesta	23
3.1. Arquitectura del sistema	23
3.1.1. Casos de uso	24
3.1.2. Diagramas de Secuencias	25
3.1.3. Modulos del Sistema	29
3.1.3.1. Módulo de persistencia	29
3.1.3.2. Módulo de conexión	32
3.1.3.3. Módulo de seguridad	32
3.2. Flujo del sistema	33
3.2.1. Cliente	33
3.2.1.1. Consulta de Información	34
3.2.1.2. Envío de multas	37
3.2.1.3. Rastreo de usuarios	37
3.2.2. Servidor	38
3.2.2.1. Servidor web-móvil de infracciones	38
3.2.3. Sistema web de infracciones	38
4. Pruebas y Resultados	41
4.1. Desempeño del sistema	42
4.1.1. Tiempos de respuesta sobre redes WiFi y GPRS	42
4.1.1.1. Sockets TCP	42
4.1.1.2. Sockets UDP	43
4.1.1.3. Peticiones GET	43
4.1.1.4. Peticiones POST	43
4.1.2. Duración de la batería del cliente móvil	43
4.1.3. TTFF	43
5. Conclusiones y trabajo futuro	45
Bibliografía	47

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*, los cuales poseen varias características desde el *hardware* como el *GPS*, acelerómetros y pantallas táctiles 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*[16], la tendencia principal del uso del Internet esta 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 de las computadoras de escritorio 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 implicaciones del cómputo móvil, esta basado en servicios de geolocalización, ofertas basadas en tiempo, y notificaciones, que aportan un uso más

personalizado del dispositivo, al contrario de la computadora de escritorio de la cual se puede obtener una menor cantidad de información para hacer una experiencia menos personal.

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 SO *Android* ofrece a los desarrolladores la posibilidad de crear aplicaciones libres[11] que toman una completa ventaja de las características del teléfono incluso modificar el SO según se requiera a las necesidades del desarrollo.

El trabajo de campo[13] se define como el conjunto de acciones encaminadas a obtener en forma directa datos de las fuentes primarias de la información, es decir, de las personas y en el lugar y tiempo en que se suscita el conjunto de hechos o acontecimientos de interés para la investigación.

Mucho del trabajo de campo realizado actualmente, es recolectado en libretas y posteriormente capturado sobre un sistema central. Sin embargo, al tener un doble reingreso de los datos es muy frecuente la ocurrencia de error de captura, asimismo no se posee una vista de los datos en tiempo real creando ambigüedades cada vez que se captura la información.

De esta forma se plantea el desarrollo de una aplicación para la captura de datos para los trabajadores de campo sobre dispositivos móviles, evitando errores humanos de captura así como tener actualizada la información en todo momento, para tener una mejor percepción del trabajo realizado, y ofrecer decisiones más acertadas.

Con el fin de demostrar el desarrollo propuesto se presenta como caso de uso o caso práctico una aplicación en el área gubernamental, 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 *Web* el cual responde las solicitudes por parte del móvil y muestra un portal *Web* para monitorear en tiempo real el desarrollo del sistema.

Objetivo General

Desarrollar un framework basado en tecnologías móviles 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 móvil
- 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 determina la forma en cada uno de estos procesos opera para su funcionamiento, limitantes y su aplicación. De esta forma se determina la forma de emplear estos procesos.

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. El desarrollo del caso de estudio para la aplicación propuesta, de tal forma que se obtenga un sistema funcional operando.

Definición de métricas: Para evaluar el desempeño de la aplicación se definen algunas métricas tales como el consumo de energía, tiempo consumido por operación y precisión del módulo de geolocalización, etcétera.

Realización de pruebas y análisis de resultados. Se evalúa la aplicación de acuerdo a las métricas establecidas y se estudian 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 presentan 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 es la creación y reproducción de archivos multimedia, el envío de información a tasas de transferencia cada vez mayores, localización de dispositivos en tiempo real, así como 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. En primer lugar se encuentran las limitantes propias del dispositivo móvil, como la duración de la pila, el tamaño de la pantalla, la disponibilidad de vías de comunicación, etcétera. En segundo plano se encuentra la necesidad de conocer una serie de tecnologías involucradas, como son la telefonía celular, los sistemas operativos propietarios, lenguajes de programación y ambientes de desarrollo especiales. En este capítulo se presentan los conceptos necesarios que dan pie al desarrollo del presente trabajo.

2.1. Cómputo Móvil

En la década de los noventa surge el campo de la computación móvil, basado en los principios de los sistemas distribuidos y la necesidad de integrar en un tipo de arquitectura a los clientes móviles, los cuales poseen características de funcionalidad y desempeño diferentes a las computadoras de uso general.

El cómputo móvil es definido como el cómputo distribuido en el cual la posición de los elementos puede cambiar durante el tiempo de ejecución, debido a que posee su fuente de energía integrada y una comunicación inalámbrica con otros equipos [9]. Los dispositivos que realizan este tipo de cómputo son vistos como computadoras que incluyen aplicaciones y programas en ejecución, accediendo código y datos almacenados de forma local y remota.

2.2. Cómputo Ubicuo

El computo ubicuo es definido como la integración de la informática en el entorno de la persona, de tal forma que los ordenadores no se perciban como objetos diferenciados [14]. Aunque la idea de tal entorno surgió en la década de los ochentas, 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 se considera que 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 individualizar el cómputo, 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, en el primero los dispositivos siguen prestando sus servicios de manera independiente y en la última se conjugan de

tal manera forma 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 un interfaz para servicio telefónico celular tal como GSM, pero adicionalmente pueden contar con una o más interfaces como Bluetooth o Wi-Fi. A continuación se describen las interfaces de comunicación presentes en 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, alrededor de 212 países [6]. Esta ubicuidad significa que los suscriptores 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 [15]. Con el transcurso del tiempo, CEPT evolucionó en una nueva organización, el Instituto Europeo de Estándares de Comunicación (ETSI por sus siglas en inglés), sin cambiar 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, GSM sus señales de voz y datos son digitales considerandola de segunda generación.

GSM es una red celular, 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. GPRS

GSM fue originalmente diseñado para soportar solamente 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 [15]. 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.

2.3.3. 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 [5].

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

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 [4].

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 conocen las la ubicación de mínimo tres satelites en un momento dado para despues obtener distancias hacia el punto de interes, 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" (TTFF Time to first fix por sus siglas en ingles) el cual es el tiempo requerido por un receptor GPS para adquirir señales satelitales y datos de navegación y e calculo a la solución de la posición llamado ajuste.

2.5. 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 más simples y están orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes formas de introducir información a estos.

La importancia creciente de los dispositivos móviles ha intensificado la competición 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.

2.5.1. 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 [8]. Matsushita se unió a Symbian en 1999, 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 S.O. 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 buen 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.

2.5.2. 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 [10].

La arquitectura subyacente del sistema y muchos módulos, son similares a las 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 [7]:

- **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*.

2.5.3. Android

Android es un nuevo sistema operativo para teléfonos móviles que fue 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 sustituir 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 de 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 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.

2.5.3.1. Arquitectura

La arquitectuta esta Basada en una pila de software que incluye el SO, *middlewa-re* y aplicaciones llave. 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 ejecutables que están en las clases de Java compilado optimizado para minimizar la huella de la memoria [12]. La arquitectura general del sistema se encuentra en la figura 2.1.



Figura 2.1: Arquitectura del Sistema Operativo Android

La pila de software está 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 de 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) que opera como un intérprete entre la aplicación y el sistema operativo. Cada aplicación que es ejecutada sobre Android está 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* serán 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.

2.5.3.2. 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 un 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 acceder a ella.

2.5.3.3. 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 a 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 2.2. 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 desapegada al usuario.
- **onResume()** Es ejecutada cuando la actividad comienza a interactuar con el usuario.
- **onPause()** Es invocada cuando la actividad va a pasar a un segundo plano.

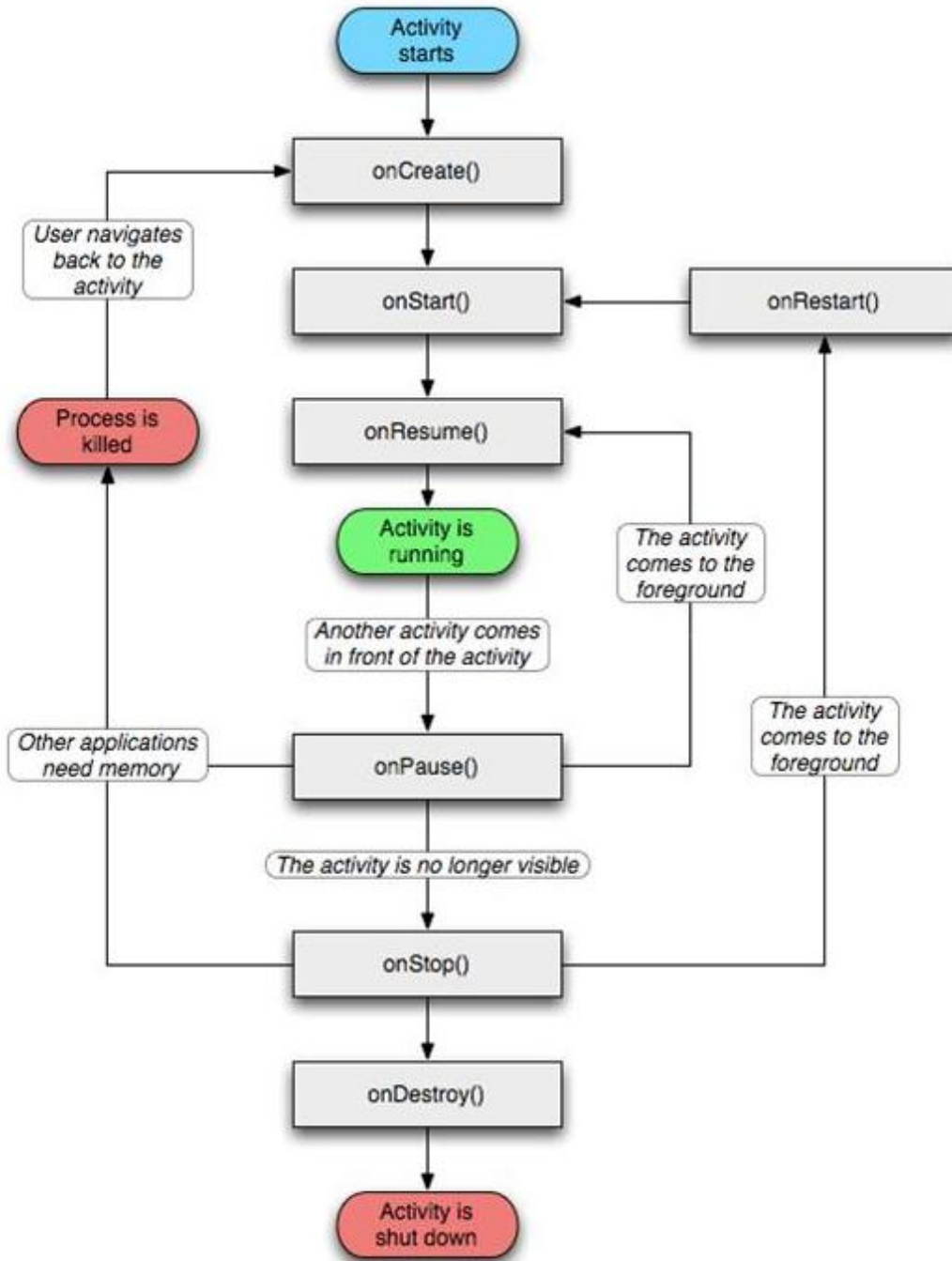


Figura 2.2: Ciclo de vida de un Actividad en Android

- **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 aplicación sea diseñada considerando estos principios.

Capítulo 3

Diseño de la propuesta

3.1. Arquitectura del sistema

En este capítulo se presenta la arquitectura empleada para el desarrollo del caso de estudio, es decir, aplicación del sistema de Infracciones.

El objetivo principal del sistema de infracciones es tener en la información prácticamente en tiempo real suministrada por los usuarios desde el trabajo, de tal forma que aún con posibles fallas en la comunicación se pueda almacenar de forma local y transmitirla tan pronto se restablezca la conexión con el servidor para mantener actualizada la información.

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 con dispositivos móviles los cuales están recolectando información continuamente, se tiene un enlace web mediante WiFi o la red celular con un servidor conectado a la Internet, teniendo una preferencia por el uso del WiFi pues no con lleva un costo inherente el envío de la información. Si no se posee una conexión con el servidor se almacena de forma local la información hasta que se posea de nuevo un enlace.

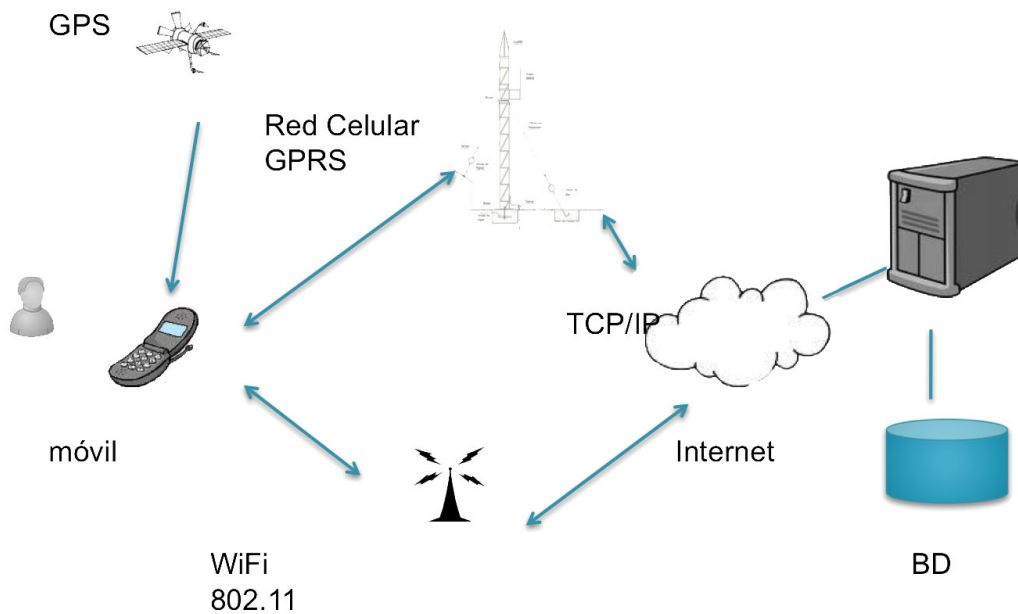


Figura 3.1: Arquitectura básica empleada para el caso de estudio

3.1.1. Casos de uso

En el sistema propuesta se tienen tres actores principales, los trabajadores de campo, administradores y conductores. De los cuales se describen a continuación sus 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 de sistema de infracciones pueden leer el estado de un vehículo o conductor o registrar un nuevo evento (multa) a través del cliente móvil el cual se comunica por Internet con un servidor en Web.

- **Administradores.**

Los administradores como es mostrado en la figura 3.3 básicamente supervisan que la operación del sistema se efectua de la forma adecuada, es decir, verifican

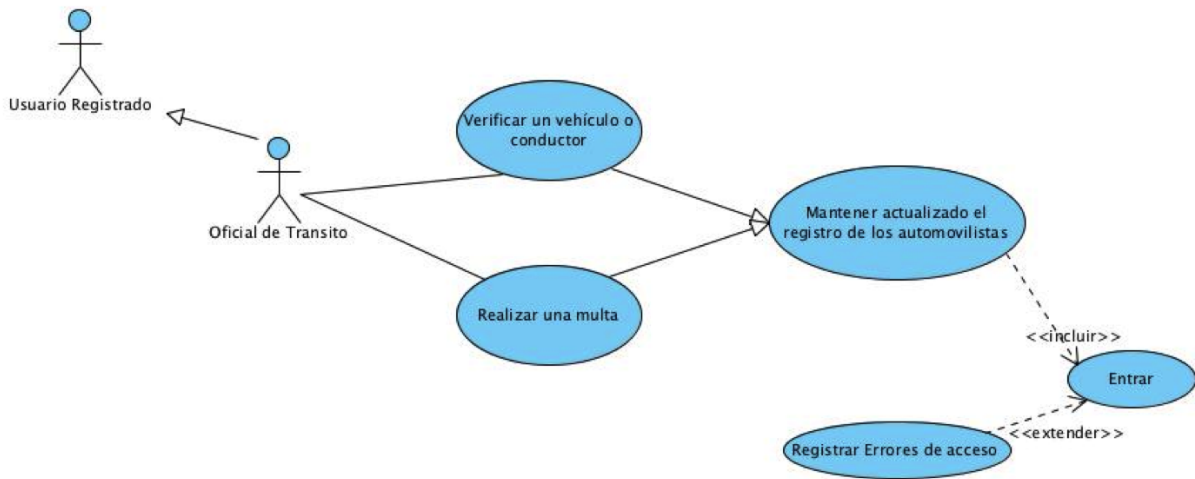


Figura 3.2: Caso de estudio para los trabajadores de campo

que los trabajadores de campo estén llevando sus actividades de forma cotidiana, examinar el registro corroborando que el número de eventos registrados y solicitud de información se mantenga en un rango considerado normal y con el paso del tiempo la obtención de estadísticas para checar tendencias.

- **Conductores.**

EL tercer actor en el sistema es el conductor de un vehículo. El cuál su función es verificar si tiene alguna multa en sistema, figura 3.4.

3.1.2. Diagramas de Secuencias

El cliente móvil 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 funcionalidades son explicadas de una forma más amplia a continuación:

- **Consulta de información.**

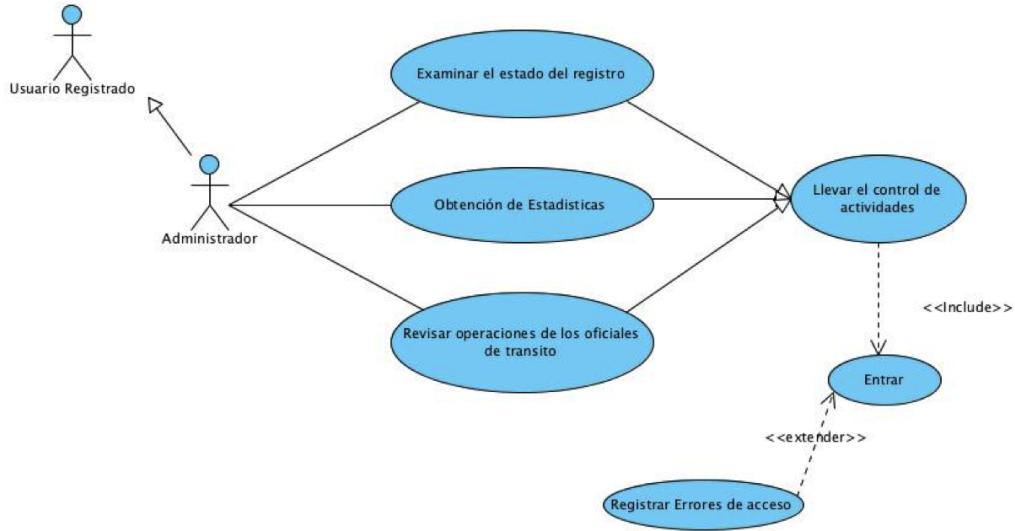


Figura 3.3: Caso de estudio para los administradores del sistema

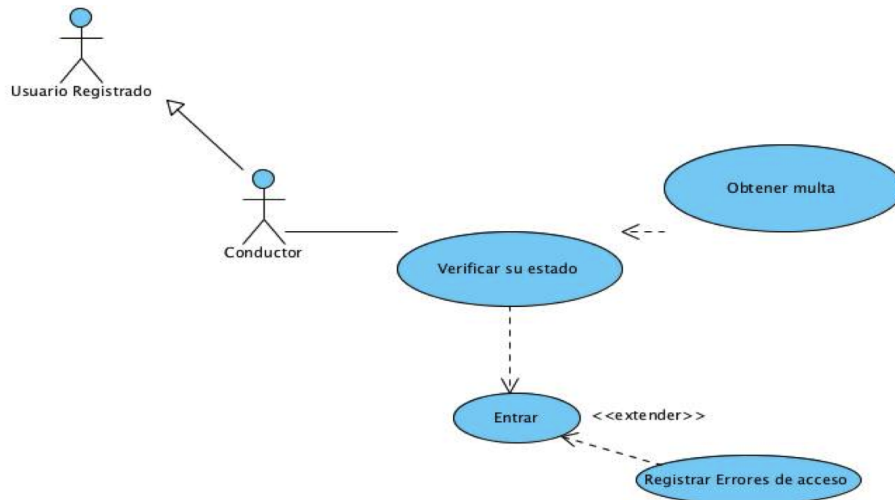


Figura 3.4: Caso del estudio para los conductores

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 credenciales para autentificarlo, la aplicación corroborara estos datos para permitir el acceso, rellena un formulario para efectuar una petición, la información pasa de la actividad actual a un servicio parte del proceso de la aplicación para administrar la solicitud web, si el móvil detecta que posee conexión a la red, el servidor es visible y alcanzable, se efectua una petición consulta en caso contrario la aplicación móvil envia un mensaje de error al usuario.

Después de 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 BD local hasta que exista una conexión con el servidor.

El caso de uso de registrar un evento, este involucra al cliente como es mostrado en el diagrama 3.6, al móvil de una forma más activa y el servidor almacenando la información generada. A diferencia de una consulta, al registrar un evento se debe poseer un mecanismo de 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 la cual puede tener un retraso considerable hasta obtener la primera coordenada dependiendo del

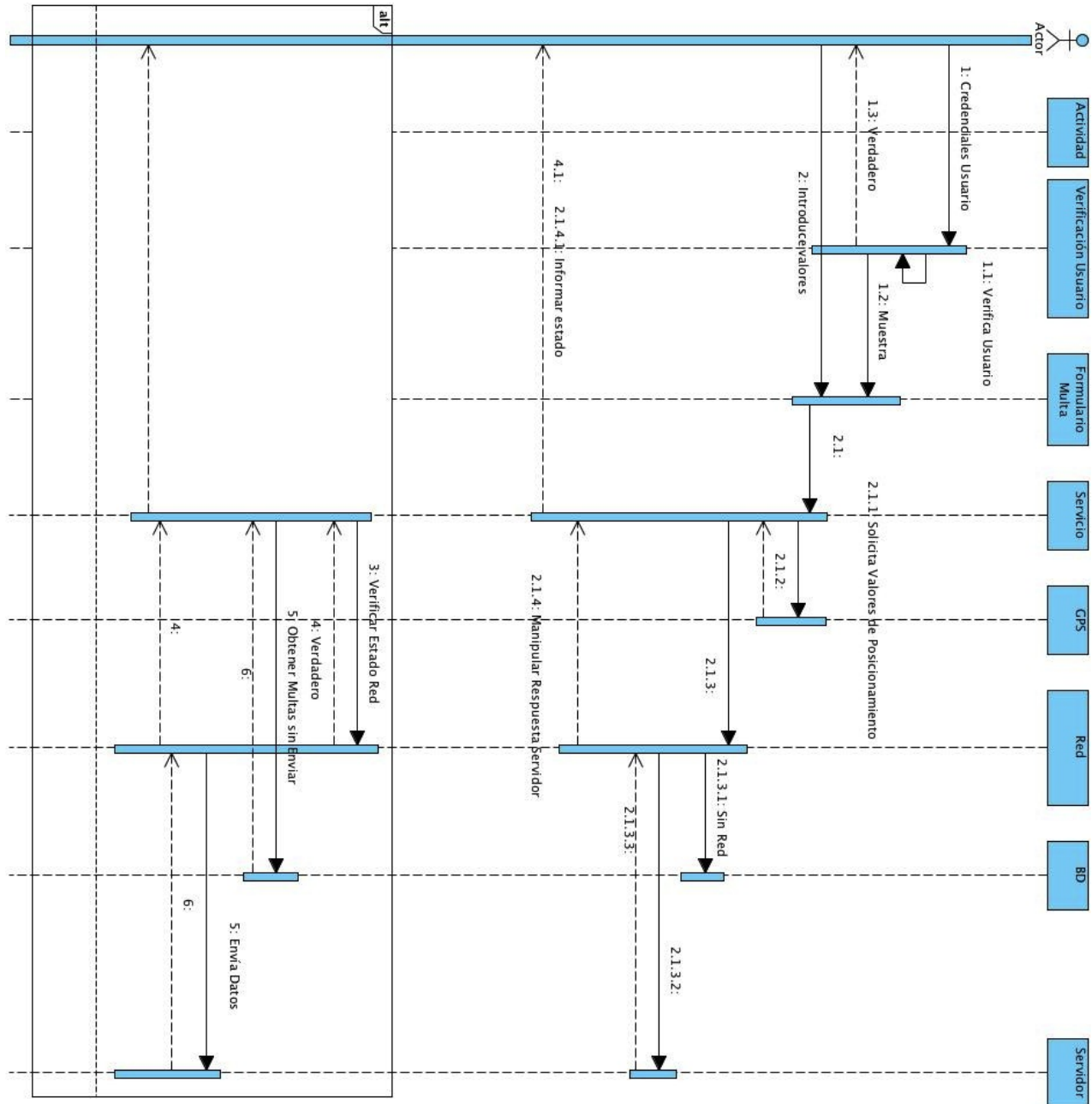


Figura 3.5: Diagrama de secuencias para la consulta de información

TTF, una vez obtenida la coordenada verifica si hay conexión a la red, si no es el caso se guarda en una BD SQLITE y cada intervalo de tiempo se verifica la conexión para enviar los registros. Por el contrario si existe una conexión a la red, envía la información y la guarda en la BD hasta que se confirme por parte del servidor que ha sido recibida.

- **Seguimiento de posición.**

La tarea de seguimiento de posición de los usuarios siempre se ejecuta en un segundo plano sin una UI presente, cada intervalo de tiempo se activa un evento que ejecuta el pedido de posición al servidor, el mecanismo de operación es muy similar al registro de un evento en la parte del servicio al del registro de un evento, cada dato generado es guardado en la BD local hasta que se confirma su recepción por parte del servidor.

3.1.3. 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.

3.1.3.1. Módulo de persistencia

Se entiende por persistencia de modo genérico, a la propiedad de los datos para que estos sobrevivan de alguna forma a 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 referencias por llave valor.

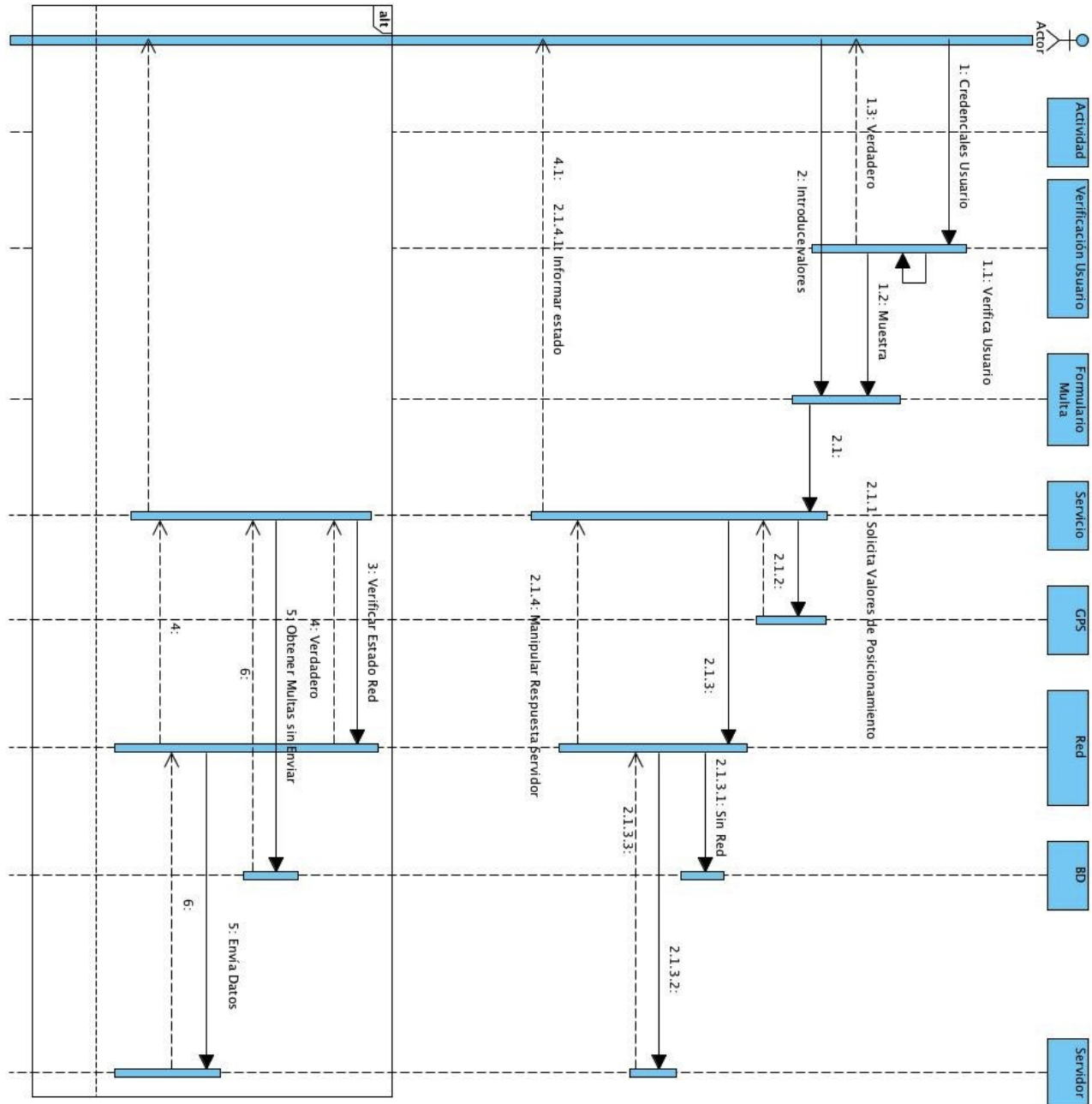


Figura 3.6: Diagrama de secuencias para la consulta de información

Del lado del cliente la persistencia es implementada en el sistema con dos mecanismos, "preferencias" llave-valor, el cual puede entenderse como una tabla con dos renglones la llave que es un identificador, y un dato asociado. Las valor "preferencias" llave-valor son empleadas para para guardar las configuraciones de ejecución de la aplicación móvil, no se recomienda su uso para guardar datos genericos del uso del programa. 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 ejecutandose en segundo plano el cual verifica cada cierto intervalo de tiempo si existen eventos por enviar, de ser así las reenvía nuevamente al servidor.

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.

3.1.3.2. 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, 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 su funcionamiento. 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 (eg. fotografías que demuetran el hecho registrado).

3.1.3.3. Módulo de seguridad

Teniendo datos sensibles en la operación de la aplicación es necesario contar con un mecanismo de protección de los datos, en otro caso la información podría ser leida 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 traves de su función hash o digesto. Una función hash es una función de una sola vía que genera un resultado teorico único con una entrada unico, 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 codigo fuente.

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

3.2. Flujo 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 los cuales deben de ser capaces de soportar la posible intermitencia en el canal de comunicación. A continuación se describe con mayor profundidad cada una de las funciones de ambos.

3.2.1. 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 representan varias ventajas los cuales incluyen: servicios basados en geolocalización, imágenes de la cámara y acelerómetros, los cuales se pueden combinar con datos disponibles en linea para ofrecer una mejor experiencia para el usuario.

En el 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 se debera logear para autenticarlo como se muestra en la figura 3.8, la primera contraseña es para permitir el acceso a la alicación, las credenciales siguientes son empeadas al identificarse



Figura 3.7: Pantalla para ingresar al sistema de Infracciones

ante el servidor para realizar cualquier operación.

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

Al ingresar a las opciones de la aplicación se presenta con una serie de parametros configurables, figura 3.9 , como son la URL del servidor, el intervalo de espera para el envío de la información en caso de error, un opción de debug que muestra opciones adicionales y mensajes extras, redefinir la contraseña del administrador para entrar a este menu asi com la opción de desabilitarla.

3.2.1.1. Consulta de Información

Durante la jornada de un usuario, 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.

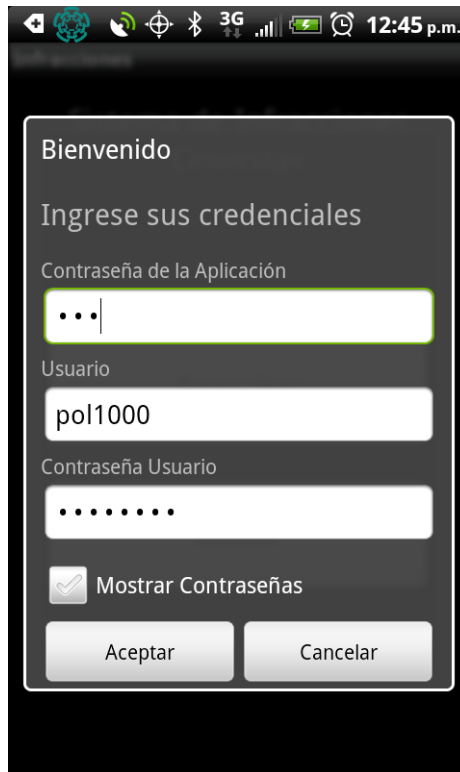


Figura 3.8: Pantalla para de opciones del sistema de infracciones

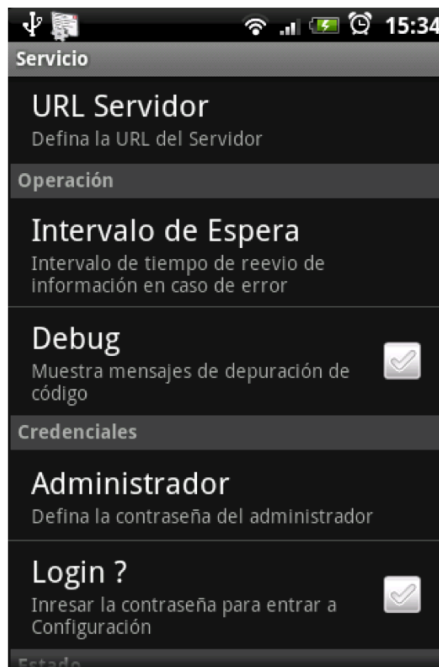


Figura 3.9: Pantalla menu del sistema de infracciones

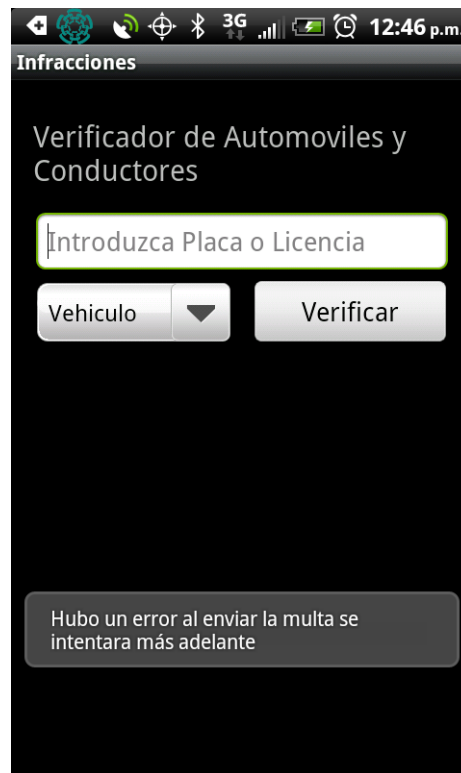


Figura 3.10: Pantalla para de consulta de un vehiculo o conductor del sistema de infracciones

Para obtener la información deseada el oficial de transito, 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 3.10 primeramente verificando el estado de la red, si hay una conexión se envía la petición y se espera por su respuesta, en caso contrario simplemente manda un mensaje de error.



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

3.2.1.2. Envío de multas

Al realizar una multa se rellena un formulario como se muestra en la figura 3.11, la cual consta del id del conductor, número de serie 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, pasa a segundo plano para obtener la coordenada del GPS así como el envío seguro de la información.

3.2.1.3. Rastreo de usuarios

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

un seguimiento a través del tiempo de su posición.

3.2.2. 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.

FIGURA

3.2.2.1. 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 una portal web el cual está a la escucha de peticiones https mediante el método POST, al recibir una petición verifica las credenciales de esta 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. Respondiendo al cliente con la información solicitada en caso de pedirlo o solo un *acknowledge* indicando que la operación se llevó correctamente

.

3.2.3. 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 el cual muestra todos los eventos generados en un mapa para tener su fácil ubicación. Dentro del portal se pueden hacer una serie de tareas 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 se puede obtener una serie de estadísticas que permiten tener un mayor control. (eg. En el sistema de multas viales se pueden detectar los puntos con mayor



Figura 3.12: Portal Web del sistema de infracciones

incidencia, el tipo de de multa que más ferquentre, en otras).

Capítulo 4

Pruebas y Resultados

Con el fin de demostrar la funcionalidad de sistema propuesto, se presentan una serie de pruebas a fin de conocer la respuesta sobre diversos factores. Entre ellos tiempo de respuesta, duración de la pila batería, tiempo para obtener la primera posición del satélite.

Para probar el sistema se utilizó el siguiente escenario: La aplicación cliente se instaló sobre un teléfono celular HTC Hero con SO Android 1.6, contando con interfaces de comunicación Bluetooth, GPRS y 802.11 (WIFI), la aplicación servidora se instaló en una computadora de escritorio con sistema operativo Linux, con el framework Django 1.2.4 para responder las peticiones Web y como BD 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. TTFF

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 una comunicación empleando diferentes medios para acceder al internet empleando diferentes protocolos.

Para llevar a cabo las pruebas del tiempo por parte del cliente se desarrollo una aplicación demostrativa que ejecuta el numero solicitado de veces, previo al desarrollo del sistema de infracciones. Para determinar el protocolo más conveniente sobre el cual de desarrollar la aplicación.

4.1.1.1. Sockets TCP

El aplicación cliente creaba un socket TCP a el servidor y el enviaba información variando el tamaño de esta, cerraba el socket y comenzaba de nuevo, midiendo el tiempo desde el cliente desde el cliente al momento de enviar la información hasta el momento de su recepción, esta prueba se repitio 1,000 veces sobre cada uno sobre WiFi y GPRS.

4.1.1.2. Sockets UDP

4.1.1.3. Peticiones GET

4.1.1.4. Peticiones POST

4.1.2. Duración de la batería del cliente móvil

§

pp

4.1.3. TTFB

pp

Capítulo 5

Conclusiones y trabajo futuro

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] Ahmed El-Rabbany. *Introduction to GPS: The Global Positioning System*. Artech house, 2002.
- [5] Matthew Gast. *802.11 Wireless Networks The Definitive Guide*. O’Reilly, 2005.
- [6] *GSM Overview*. 2010. URL: <http://www.gsmworld.com>.
- [7] Apple inc. *iPhone OS Overview*. 2000. URL: http://developer.apple.com/iphone/library/referencelibrary/GettingStarted/URL_iPhone_OS_Overview/index.html.
- [8] Digia Inc. *Programming for the Series 60 Platform and Symbian OS*. John Wiley & Sons Ltd, 2003, págs. 1,6.
- [9] I. Mahgoub M. Ilyas. *Mobile Computing Handbook. 1st. ed.* CRC Press, 2005.
- [10] Dave Mark y Jeff LaMarche. *Beginning iPhone 3 Development*. Apress, 2006, pág. 11.
- [11] Reto Meier. *Android 2 Application Development*. Wronx, 2010, pág. 543.

- [12] Abraham Omonte. *Arquitectura Android*. 2011. URL: <http://developer.android.com/guide/basics/what-is-android.html>.
- [13] Abraham Omonte. *Trabajo de Campo, concepto y diseño*. 2009. URL: <http://www.mailxmail.com/curso-ciencias-sociales-investigacion-administrativas-academica/trabajo-campo-concepto-diseno>.
- [14] Saúl Eduardo Pomares Hernández. *Computación Ubicua; un gran desafío*. INAOE, 2009.
- [15] Chin-Chun Lee Raymond Steele y Peter Gould. *cdmaOne and 3G Systems*. John Wiley & Sons Ltd, 2001, pág. 65.
- [16] Morgan Stanley. *Internet Trends*. 2010. URL: http://www.morganstanley.com/institutional/techresearch/internet_trends042010.html.