

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL

CARRERA:

INGENIERÍA EN SISTEMAS

**Tesis previa a la obtención del título de:
Ingeniero en Sistemas con Mención en Telemática**

TEMA:

**“Análisis y solución al congestionamiento vehicular en horas pico
utilizando una aplicación móvil con GPS”**

AUTORES:

Patricio David Mendoza Solórzano

Cristhian Ángel Villacis Vargas

DIRECTOR:

Ing. Miguel Quiroz Martínez

Guayaquil – 2014

DEDICATORIA

A Dios por que ha sido quien me ha ayudado y me ha guiado a concluir esta etapa de mi vida.

A las personas que día a día han luchado conmigo para que sea un gran profesional mi familia, en primer lugar mis padres Ángel Ramiro Villacis Albán y Reina Isabel Vargas Fariño quienes me siento orgullosamente y felizmente que están siempre a mi lado, ante las adversidades de mi vida, mis hermanos Alex Villacis y Paul Villacis quienes han sido de ejemplo y me han dado la mano cuando más lo necesitaba.

A mis compañeros con quien egrese, con quienes pude contar cuando no entendía algún tema en particular o cuando teníamos alguna tarea pendiente.

A mis amigos, quienes siempre me dieron sus palabras de apoyo para que siga adelante. Y mis colegas que con sus opiniones permitieron que ambos terminemos este proyecto.

Cristhian Ángel Villacis Vargas

DEDICATORIA

A Dios quien me ha guiado durante toda mi carrera universitaria y me seguirá guiando durante el camino que todavía falta.

A mis abuelos Blanca Castro Barsola, Francisco Solórzano Govea y Dora Chukry quienes son un pilar más y me siento orgulloso de ser su nieto, de recibir esas enseñanzas y experiencias que solo ellos pueden transmitir.

A mi padre Jaime Armando Mendoza López y mi madre Rosa Narcisa Solórzano Castro los cuales son un ejemplo a seguir de su constancia, dedicación y trabajo para mi formación académica y personal.

A mis compañeros con quien estuve compartiendo las enseñanzas de los profesores, dudas y tareas durante nuestra carrera universitaria.

A mis profesores a quienes tendré el gusto de llamarlos colegas quienes estuvieron apoyándome durante la ejecución de este proyecto.

Patricio David Mendoza Solórzano

AGRADECIMIENTO

Agradezco a Dios por siempre apoyarme en los momentos difíciles y haberme permitido no derrumbarme ante las dificultades durante la carrera universitaria y poder haber llegado con bendición a culminar mi carrera y mi proyecto de tesis.

A la Universidad Politécnica Salesiana sede Guayaquil por brindarme los conocimientos necesarios para poder afrontar diferentes situaciones en el plano laboral o personal.

A mi mamá por ser quien más se preocupó y estuvo pendiente de mí ante cualquier problema presentado en la universidad, además de darme sus buenos consejos para poder afrontarlos.

Al Ing. Miguel Quiroz por su inmensa ayuda desinteresada en el desarrollo de este proyecto fomentando siempre cosas innovadoras.

A todas las personas que me han ayudado y apoyado de manera desinteresada e incondicional muchas gracias.

Cristhian Ángel Villacis Vargas

AGRADECIMIENTO

La presente tesis es un esfuerzo en el cual, directa o indirectamente, participaron varias personas leyendo, opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañando en los momentos de crisis y en los momentos de felicidad.

Agradezco a Dios por permitirme llegar a estas instancias de mi vida por ser la guía y el que siempre me ha estado levantando de mis caídas.

A mis padres y abuelos quienes me han estado apoyando en cada situación que se me ha presentado durante toda mi vida.

A mis amigos y compañeros con quienes he compartido toda mi carrera universitaria, apoyándonos los unos a los otros para conseguir un objetivo común.

A todos mis profesores que me han enseñado muchas cosas que me servirán en mi vida profesional y personal.

Patricio David Mendoza Solórzano

CERTIFICADO

Certifico que el presente trabajo fue realizado por los señores Mendoza Solórzano Patricio David y Villacis Vargas Cristhian Ángel, bajo mi supervisión.

Guayaquil, Enero del 2014.

Ing. Miguel Quiroz Martínez
DIRECTOR DE TESIS

DECLARACIÓN DE RESPONSABILIDAD

Nosotros Mendoza Solórzano Patricio David y Villacis Vargas Cristhian Ángel declaramos que el trabajo realizado dentro de este tema de tesis es netamente de nuestra autoría, en lo que corresponde al planteamiento y al desarrollo y los derechos correspondientes le pertenecen a la Universidad Politécnica Salesiana.

Guayaquil, Enero del 2014.

Mendoza Solórzano Patricio David

Villacis Vargas Cristhian Ángel

RESUMEN

La congestión vehicular en nuestro país ha venido en crecimiento en estos últimos años por motivo del aumento del parque automotor, razón por la cual se optó por la inclusión como medida el sistema de restricción vehicular al que se le denominó “Pico y Placa”, desde el día 3 de mayo del 2010 en la ciudad de Quito.

El cual consiste en la restricción en la circulación vehicular de lunes a viernes, tomando en cuenta el último dígito de la placa del vehículo, de acuerdo al cronograma propuesto por la Comisión Nacional de Tránsito del Ecuador (CTE), en las horas picos donde se identificó que existe mayor tráfico vehicular, como son en la mañana desde las 7h00 a 9h30 y en la tarde desde las 16h00 a 19h30.

Por motivo de este problema ha aumentado la tasa de accidentes de tránsito y la contaminación ambiental, pensando en las consecuencias de esta problemática se impulsó el desarrollo de una aplicación móvil que permita compartir nuestro medio de transporte.

Nuestra aplicación móvil llamada “Llévame” nos ayuda a mejorar los siguientes aspectos en la sociedad:

- Reducción de tráfico vehicular.
- Menos contaminación ambiental.
- Promueve la cooperación.
- Ahorro de Tiempo al trasladarse de un sitio a otro
- Incentivar en el uso de tecnologías móvil
- Permite trazar ruta y escoger la mejor.

ABSTRACT

Traffic congestion in our country has been growing in recent years on account of the increase in vehicles, which is why we opted for inclusion as a measure restricting vehicular system to which he call "Pico y Placa" from the day May 3, 2010 in the city of Quito.

Which is the restriction on vehicular movement Monday through Friday, taking into account the last digit of the license plate, according to the schedule proposed by the National Commission Transit Ecuador (CTE) in the peak hours where identified that there is more traffic, as they are in the morning from 7:00 am to 9:30 am and in the afternoon from 16h00 to 19h30.

By reason of this problem that has increased the rate of traffic accidents and environmental pollution and thinking about the consequences of this problem prompted the development of a mobile application for sharing our means of transportation.

Our mobile application called "Take Me" helps us to improve the following aspects in society:

- Reduced traffic.
- Less pollution.
- Promotes cooperation.
- Saving Time to move from one place to another
- Encourages the use of mobile technologies
- It can chart and choose the best route.

Tabla de contenido

DEDICATORIA CRISTHIAN ANGEL VILLACIS VARGAS	II
DEDICATORIA PATRICIO DAVID MENDOZA SOLORZANO.....	III
AGRADECIMIENTO CRISTHIAN ANGEL VILLACIS VARGAS.....	IV
AGRADECIMIENTO PATRICIO DAVID MENDOZA SOLORZANO	V
CERTIFICADO.....	VI
DECLARACIÓN DE RESPONSABILIDAD	VII
RESUMEN	VIII
ABSTRACT	IX
INTRODUCCIÓN.....	1
CAPITULO 1	2
DISEÑO DE LA INVESTIGACIÓN.....	2
1.1 Antecedentes de la investigación.....	2
1.2 Problema de investigación.....	3
1.2.1 Planteamiento del problema de investigación	3
1.2.2 Formulación del problema de investigación.....	3
1.2.3 Sistematización del problema de investigación.....	4
1.3 Objetivos de la investigación.....	4
1.3.1 Objetivos generales.....	4
1.3.2 Objetivos específicos.....	4
1.4 Justificación de la investigación	5
CAPITULO 2	7
MARCO DE REFERENCIA DE LA INVESTIGACIÓN.....	7
2.1 Marco Teórico	7
2.1.1 Plataforma Android	7
2.1.2 Arquitectura Cliente – Servidor.....	12
2.1.3 Modelo Vista Controlador (MVC)	15
2.1.4 Arquitectura de 3 capas	17
2.1.5 Lenguaje de programación PHP.....	18
2.1.6 Motor de Base de Datos MySQL	20
2.1.7 Kit de Desarrollo de Software de Android (SDK)	20
2.1.8 Interfaz de Programación de Aplicaciones (API) y API Google Maps.....	22
2.1.9 Servicios Web y sus componentes	23

2.1.10	Fórmula Haversine	29
2.1.11	Sistema de posicionamiento global (GPS)	30
2.1.12	Contaminación Ambiental por emisión de gases de los automóviles	32
2.2	Marco Conceptual.....	34
2.3	Formulación de la hipótesis y variables	39
2.3.1	Hipótesis general	39
2.3.2	Hipótesis particulares	39
2.4	Matriz Causa-Efecto	40
2.5	Variables e indicadores.....	42
CAPITULO 3	43
MARCO METODOLOGICO DE LA INVESTIGACIÓN	43
3.1	Tipo de estudio	43
3.2	Métodos de Investigación.....	44
3.3	Fuentes y Técnicas para la recopilación de información.....	44
3.4	Población y Muestreo	44
3.5	Tratamiento de la Información	45
3.6	Resultado Esperado	46
CAPITULO 4	47
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS	47
4.1	Resultados de las encuestas	47
4.2	Arquitectura del Sistema	56
4.3	Descripción general de la Arquitectura	57
4.3.1	Diagrama de bloques	57
4.3.2	Diagrama de Actividades	58
4.3.3	Fórmula de Haversine.....	58
4.3.4	Trazos de rutas en Google Maps	59
4.3.5	Búsqueda de lugares	61
4.3.6	Diagrama de Casos de Uso.....	62
4.3.7	Diagramas de Eventos	64
4.3.8	Diagrama de Interacción.....	67
4.4	Diseño de la arquitectura	69
4.4.1	Capa de Base de Datos	69
4.4.2	Definición de Tablas del Sistema	70

4.4.3 Definición de Procedimientos Almacenados.....	75
4.5. Capa de Reglas de Negocio	77
4.6 Especificación de Librerías	77
4.7 Capa de presentación (Interfaz Gráfica).....	79
4.7.1 Descripción General de Ventanas y menú de la Aplicación móvil.....	79
4.7.2 Instalación de la Aplicación móvil Llévame	96
4.8 Pruebas y Resultados	96
4.8.1 Resultados de las pruebas	96
4.8.2 Evaluación Final	97
4.9 Requerimientos de Hardware y Software	98
CAPITULO 5	99
CONCLUSIONES Y RECOMENDACIONES	99
5.1 Conclusiones.....	99
5.2 Recomendaciones	100
BIBIOGRAFIA.....	101
ANEXOS	104
Anexo 1. Código de procedimiento almacenado para registrar un usuario.....	105
Anexo 2. Código del procedimiento almacenado para actualizar la posición del usuario	106
Anexo 3. Código del procedimiento almacenado para enviar solicitud de llevar ...	107
Anexo 4. Código de procedimiento almacenado para los tipos de publicaciones....	108
Anexo 5. Código de procedimiento almacenado para los tipos de notificaciones ...	110
Anexo 6. Código del WebServices WSDDataQueries.....	111
Anexo 7. Código del web Services WSDDataOperations.....	113
Anexo 8. Código del web Services WSCCommonQueries	114

INDICE DE TABLAS

Tabla 2.1 Matriz Causa - Efecto	40
Tabla 4.2 Tabla con resultados de opinión de la pregunta N° 1	47
Tabla 4.3 Tabla con resultados de opinión de la pregunta N° 2.....	48
Tabla 4.4 Tabla con resultados de opinión de la pregunta N° 3.....	49
Tabla 4.5 Tabla con resultados de opinión de la pregunta N° 4.....	50
Tabla 4.6 Tabla con resultados de opinión de la pregunta N° 5.....	51
Tabla 4.7 Tabla con resultados de opinión de la pregunta N° 6.....	52
Tabla 4.8 Tabla con resultados de opinión de la pregunta N° 7.....	53
Tabla 4.9 Tabla con resultados de opinión de la pregunta N° 8.....	54
Tabla 4.10 Tabla con resultados de opinión de la pregunta N° 9.....	55
Tabla 4.11 Diagrama de Eventos.....	64
Tabla 4.12 Diccionario de Datos de la tabla usuarios	70
Tabla 4.13 Diccionario de Datos de la tabla datos personales	70
Tabla 4.14 Diccionario de Datos de la tabla coordenadas de usuarios.....	71
Tabla 4.15 Diccionario de Datos de la tabla coordenadas de visitas.....	72
Tabla 4.16 Diccionario de Datos de la tabla historial de mensajes de estado	72
Tabla 4.17 Diccionario de Datos de la tabla amigos	73
Tabla 4.18 Diccionario de Datos de la tabla solicitudes de llévame	74
Tabla 4.19 Diccionario de Datos de la tabla estados.....	74
Tabla 4.20 Diccionario de Datos de la tabla parámetros.....	75
Tabla 4.21 Tabla de resultados de pruebas iniciales	96
Tabla 4.22 Parámetros para funcionamiento de la aplicación	97

INDICE DE FIGURAS

Figura 1.1 Caos vehicular en Guayaquil	2
Figura 1.2 Crecimiento del parque automotor en este año 2013	5
Figura 2.3 Entorno de la Plataforma Android	7
Figura 2.4 Ejemplo básico de una Activity en Android	8
Figura 2.5 Ejemplo de un Service en Android	8
Figura 2.6 ContentProvider como gestor de base de datos.....	9
Figura 2.7 Ejemplo básico de un BroadcastReceiver	10
Figura 2.8 Porcentaje de distribución de las versiones de Android	12
Figura 2.9 Diseño de una arquitectura Cliente - Servidor	13
Figura 2.10 Modelo Vista Controlador en Android	16
Figura 2.11 Diseño de una Arquitectura de 3 capas	17
Figura 2.12 Esquema de funcionamiento de una página en PHP	19
Figura 2.13 Entorno de desarrollo de Android	21
Figura 2.14 Aplicación Móvil usando el API de Google	23
Figura 2.15 Proceso del WSDL.....	24
Figura 2.16 Estructura de un documento WSDL	25
Figura 2.17 Response del SOAP	25
Figura 2.18 Estructura de un mensaje SOAP	26
Figura 2.19 Ecuación de Haversine	29
Figura 2.20 Nomenclatura de la ecuación de Haversine	29
Figura 2.21 Fórmula directa de la distancia	30
Figura 2.22 Funcionamiento del GPS.....	31
Figura 2.23 Integración del GPS en el dispositivo Móvil	32
Figura 2.24 Porcentaje de la contaminación por los automóviles	33
Figura 2.25 Porcentajes de número de vehículos a Nivel Nacional	34
Figura 4.26 Porcentajes de la pregunta N° 1	48
Figura 4.27 Porcentajes de la pregunta N° 2	49
Figura 4.28 Porcentajes de la pregunta N° 3	50
Figura 4.29 Porcentajes de la pregunta N° 4	51
Figura 4.30 Porcentajes de la pregunta N° 5	52
Figura 4.31 Porcentajes de la pregunta N° 6	53

Figura 4.32 Porcentajes de la pregunta N° 7	54
Figura 4.33 Porcentajes de la pregunta N° 8	55
Figura 4.34 Porcentajes de la pregunta N° 9	56
Figura 4.35 Diagrama de bloques de la aplicación.....	57
Figura 4.36 Diagrama de Actividades para postear llevar	58
Figura 4.37 Fórmula de Haversine en MySQL.....	59
Figura 4.38 Solicitud del API de rutas	59
Figura 4.39 Fragmento de código para la obtención de coordenadas	60
Figura 4.40 Fragmento de código para dibujar el trazo de la ruta.....	60
Figura 4.41 Solicitud de búsqueda de lugares API Google.....	61
Figura 4.42 Fragmento de código para obtener lugar seleccionado	62
Figura 4.43 Diagrama de casos de uso	62
Figura 4.44 Diagrama de interacción para postear llevar.....	67
Figura 4.45 Diagrama de Interacción enviar solicitud de amistad	68
Figura 4.46 Modelo Entidad-Relación	69
Figura 4.47 Esquema del procedimiento registrar usuario	76
Figura 4.48 Esquema del procedimiento actualiza ubicación	76
Figura 4.49 Esquema del procedimiento envía solicitud llévame.....	77
Figura 4.50 Descripción de carpetas creadas en la capa de negocio	78
Figura 4.51 Pantalla de inicio de la aplicación.....	79
Figura 4.52 Pantalla de inicio de sesión de la aplicación	80
Figura 4.53 Pantalla de registro de la aplicación.....	81
Figura 4.54 Icono de Usuario de la aplicación móvil.....	81
Figura 4.55 Pantalla de Login de la aplicación	82
Figura 4.56 Pantalla del Home de la aplicación	83
Figura 4.57 Icono de Ubicación del lugar	83
Figura 4.58 Pantalla de Búsqueda de Ruta de la aplicación.....	84
Figura 4.59 Icono de pestaña de retroceso	84
Figura 4.60 Menú de la pantalla del Home	85
Figura 4.61 Icono de la aplicación móvil Llévame	85
Figura 4.62 Pantalla de Notificaciones de la aplicación.....	86
Figura 4.63 Icono de escribir publicaciones.....	86
Figura 4.64 Pantalla de Publicación de la aplicación	87
Figura 4.65 Icono para Seleccionar un lugar.....	87

Figura 4.66 Pantalla de Búsqueda del lugar en la aplicación	88
Figura 4.67 Pantalla de Mapa en la aplicación	89
Figura 4.68 Pantalla de mis amigos en la aplicación.....	90
Figura 4.69 Pantalla de Configuración	91
Figura 4.70 Pantalla de Configuración de la Cuenta de la aplicación.....	92
Figura 4.71 Pantalla de edición de perfil	93
Figura 4.72 Pantalla de cambio de contraseña	93
Figura 4.73 Pantalla para buscar amigos en la aplicación.....	94
Figura 4.74 Pantalla de Solicitudes de Amistad	95
Figura 4.75 Pantalla de Buscar todos los amigos	95

INTRODUCCIÓN

A medida de que pasa el tiempo el crecimiento del parque automotor del país está avanzando, produciendo un desorden vehicular, siendo las provincias más afectadas las provincias con mayor población como son Pichincha, Guayas y Azuay.

Por este motivo, se incentivó en nuestro país la utilización del programa de restricción vehicular Pico y Placa, para racionar el uso del transporte terrestre debido a la demanda de vehículos circulando en las calles, para tratar de esta manera descongestionar el tránsito y tratar de disminuir el alto índice de accidentes.

Esta medida empezó a implementarse en la ciudad de Quito, donde se optó restringir la circulación de ciertos automotores por seis horas, cada día, de acuerdo con el último dígito de su placa. Con esto se pretendió evaluar el comportamiento del flujo vehicular y tratar de formular recomendaciones para mejorar el sistema de transporte vehicular actual y una estrategia de aumento en la productividad de esta ciudad.

En la actualidad, no se logró controlar este problema en su totalidad. De acuerdo a esta problemática se buscó desarrollar una aplicación móvil para tratar de ayudar e incentivar la colaboración de las personas que usan el transporte particular, aprovechando del avance tecnológico como el Internet, y la telefonía celular que hará llegar a todas las personas que dispongan de estos servicios.

Esta aplicación tendrá un entorno social similar a una red social como Twitter, Facebook o Foursquare, con su uso se podrá incentivar la cooperación entre personas como medio para poder transportarse de un lugar a otro y de manera segura, interactuando con personas conocidas, ayudando de esta manera a poder llegar de manera efectiva y eficaz al lugar donde se dirijan las personas.

CAPITULO 1

DISEÑO DE LA INVESTIGACIÓN

1.1 Antecedentes de la investigación

Debido a la gran conglomeración de automotores en la ciudad de Quito, se impulsó el programa “Pico y Placa” desde el 3 de Mayo del 2010, donde se optó restringir la circulación de automóviles livianos por seis horas, cada día, y de acuerdo con el último dígito de su placa.

Actualmente la Secretaría de Movilidad del Distrito Metropolitano de Quito sostuvo que esta medida seguirá vigente este año en esta ciudad, y se ejecutará, con dos dígitos diarios en horarios de 07:00 a 09:30 y desde las 16:00 hasta las 19:30. La Comisión de Tránsito del Ecuador CTE busca que se pueda ampliar en las demás ciudades con mayor tráfico vehicular como Guayas, Cuenca.

Figura 1.1 Caos vehicular en Guayaquil



Nota: Guayaquil, tráfico vehicular en Guayaquil; Av. de las Américas. Obtenido del diario Expreso (Junio 2010)

La inclusión de la Metro vía por parte del Municipio de Guayaquil pudo reducir un porcentaje del tráfico vehicular, debido al cambio de recorridos a diferentes líneas de transporte público en las principales calles, porque según cifras “el 60% del congestionamiento vehicular se concentra en los sectores comerciales, principalmente en el centro, y de acuerdo con datos de la Comisión de Tránsito del Ecuador (CTE) el parque automotor en Guayaquil crece entre 10% y 15% cada año.” (Expreso, 2012)

Actualmente el automóvil de uso particular se considera un inconveniente en las ciudades más grandes del Ecuador, por ser el medio de transporte donde la mayoría de veces su uso es utilizado por solo una persona, lo que se busca en los próximos años es en concientizar a esas personas, a compartir su vehículo con la sociedad.

1.2 Problema de investigación

1.2.1 Planteamiento del problema de investigación

Una de las medidas para reducir el embotellamiento vehicular es el Sistema de Pico y Placa que se incluirá en el país, este nuevo método busca solucionar el congestionamiento vehicular y reducir la tasa de accidentes de tránsito, evitando el uso innecesario de los vehículos particulares y estimular que se utilice el transporte público para disminuir la congestión en las horas de máxima demanda.

De alguna manera el programa Pico y Placa desmotivará a las personas que tienen su propio vehículo particular por el cual se transportan día a día, sin embargo para otras personas resultará un viaje seguro y una reducción del consumo de gasolina ya que se optaría por otros medios de transportes terrestre.

De continuar el descontento de los usuarios al utilizar este programa de restricción vehicular podría ocasionar que el sistema no cumpla con los resultados esperados, por la falta de colaboración de los usuarios.

Como una medida para solucionar esto se presenta una alternativa moderna, accesible, útil y de fácil uso que ayudará a reducir el descontento de los usuarios por el Pico y Placa, y a su vez incentiva la cooperación de los conductores para que se adapten a la medida del programa.

1.2.2 Formulación del problema de investigación

¿Cómo una aplicación móvil podrá beneficiar a los medios de transporte terrestre y colaborar con el sistema de restricción vehicular?

1.2.3 Sistematización del problema de investigación

¿Cómo fomentar la cooperación de las personas al sistema de restricción vehicular y reducir la contaminación?

¿Qué aportes dará el uso de esta red social móvil en el país?

¿Qué influencia positiva tendrá el avance tecnológico móvil ante la sociedad ecuatoriana?

1.3 Objetivos de la investigación

1.3.1 Objetivos generales

Crear una aplicación móvil para dispositivos con sistema operativo Android que nos permita mejorar y facilitar el sistema de transportación terrestre que colabore con la restricción vehicular a través de una red social de amigos para así llegar al lugar que se desee de una manera rápida, segura y colaborativa.

1.3.2 Objetivos específicos

- Se logrará fomentar el uso adecuado del flujo vehicular con la ayuda de la aplicación móvil impulsando con esta la interacción y cooperación de las personas, ayudando a reducir el congestión vehicular y la contaminación ambiental.

- Realizar una aplicación orientada a la última tendencia tecnológica como lo son las aplicaciones móviles y el lenguaje de programación Android, permitiendo incentivar a las personas el desarrollo de nuevos aportes tecnológicos a la sociedad y mediante estos mejorar el estilo de vida de las personas.

- La tecnología cada día nos ayuda a vivir mejor, al contar con teléfonos inteligentes con mayores prestaciones influirá a que nuestra sociedad pueda aprovechar todos los recursos que estos dispositivos poseen de una mejor

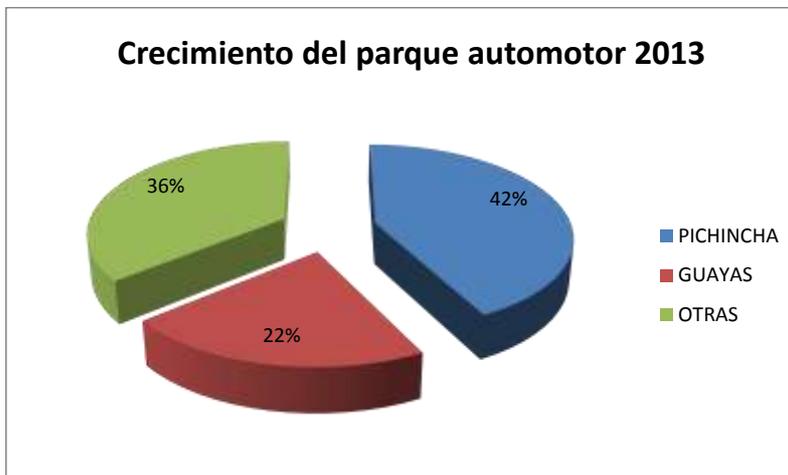
manera, ya que día a día se presentan muchas necesidades en diversos aspectos social, económico, ambiental, entre otros.

1.4 Justificación de la investigación

El uso de redes sociales ha causado un gran impacto en la sociedad actual, el poder compartir intereses, gustos y demás actividades entre conocidos, amigos y familiares nos permite tener mejor análisis de las problemáticas sociales.

En Guayaquil no existe sector donde no se presenten embotellamientos, esta situación en algunos ratos es insostenible, principalmente en los dueños de autos particulares, todo esto debido en su totalidad al aumento del parque automotor en un 60%, registrando 320 mil vehículos durante el año vigente, y otra de las razones como la circulación exclusiva de la metrovía en vías en donde aún van buses de transporte urbano. (García, 2013)

Figura 1.2 Crecimiento del parque automotor en este año 2013



Nota: Incremento del parque automotor en el año 2013. Elaborado por los Autores (Junio 2013)

Según la cifras, el parque automotor está en crecimiento a pesar de las medidas que han optado las entidades encargadas del tránsito vehicular. A su vez otros de los factores que están siendo afectados debido a esto, es el medio ambiente debido de la gran cantidad de dióxido de carbono expulsados por los vehículos.

Lo que se busca es que puedan compartir su vehículo ayudando a evitar congestionamientos en sectores claves de la ciudad, el poder llegar a nuestro destino de una manera eficaz, segura y rápida en una ruta con menor tránsito, con esto se reducirá el gran número de automotores, reduciendo el tráfico de una manera considerable.

El uso de esta aplicación podrá beneficiar a toda la sociedad, no solo a las personas que cuentan con un dispositivo móvil con sistema operativo Android al usar la aplicación, sino también a las personas que no lo posean ya que podrán vivir en un medio ambiente más sano ya que se reducirá de emisión de gases contaminantes por parte de los vehículos.

La tendencia en el futuro es mejorar el ecosistema, y con esta aplicación contribuir en resolver este problema que más preocupa a la sociedad mundial.

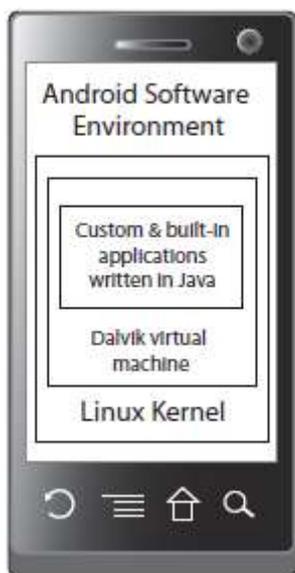
CAPITULO 2

MARCO DE REFERENCIA DE LA INVESTIGACIÓN

2.1 Marco Teórico

2.1.1 Plataforma Android

Figura 2.3 Entorno de la Plataforma Android



Nota: Android es un software que aprovecha el kernel de Linux para interactuar con el hardware. Obtenida del libro Android in Action (Enero 2010)

“Es un entorno de software integrado para dispositivos móviles, que incluye un kernel basado en sistema operativo Linux. Sus desarrolladores escriben sus aplicaciones en el lenguaje de programación Java y en su entorno de ejecución conocido como Dalvik virtual Machine (VM). Los cuatro componentes principales para construir una aplicación son: *Activity*, *Service*, *ContentProvider*, y *BroadcastReceiver*.” (W.Frank Ableson, 2012)

Activity.- Es una parte del ciclo de vida de la aplicación, ya que se trata de la interfaz que interactúa con el usuario. Su componente principal es la vista en donde se manejan los elementos del diseño como formularios, botones, gráficos. A menudo las aplicaciones Android contienen más de una actividad la cual responde a los eventos del sistema iniciadas por el usuario. Su relación entre una *Activity* y la interfaz gráfica es similar al paradigma MVC (Modelo Vista Controlador).

Figura 2.4 Ejemplo básico de una Activity en Android

```
package com.example.formulario;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class Activity1 extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_activity1);
    }
}
```

Nota: La principal tarea de una actividad es mostrar los elementos de la interfaz del usuario. Elaborado por los Autores (Junio 2013)

Service.- Si una aplicación va a tener un largo ciclo de vida, es necesario crear un servicio. Al igual que una actividad es una clase que permite extender el tiempo en ejecución en Android, y puede incorporar servicios de manera periódica.

Figura 2.5 Ejemplo de un Service en Android

```
package com.example.formulario;

import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.util.Log;
public class Service1 extends Service implements Runnable { ← 1 Extend Service class
    public static final String tag = "service1";
    private int counter = 0;
    @Override
    public void onCreate() { ← 2 Initialization
        super.onCreate();
        Thread aThread = new Thread (this);
        aThread.start();
    }
    public void run() {
        while (true) {
            try {
                Log.i(tag,"service1 firing : # " + counter++);
                Thread.sleep(1000);
            } catch(Exception ee) {
                Log.e(tag,ee.getMessage());
            }
        }
    }
    @Override
    public IBinder onBind(Intent arg0) { ← 3 Handle Binding request
        // 1000 Auto-generated method stub
        return null;
    }
}
```

Nota: En la figura se observa que en el literal (1) la clase Service1 extiende de la clase Service que implementa la interfaz Runnable, en el (2) el método onCreate permite realizar la inicialización del servicio, y en el (3) el onBind permite devolver el canal de comunicación. Elaborado por los Autores. (Junio 2013)

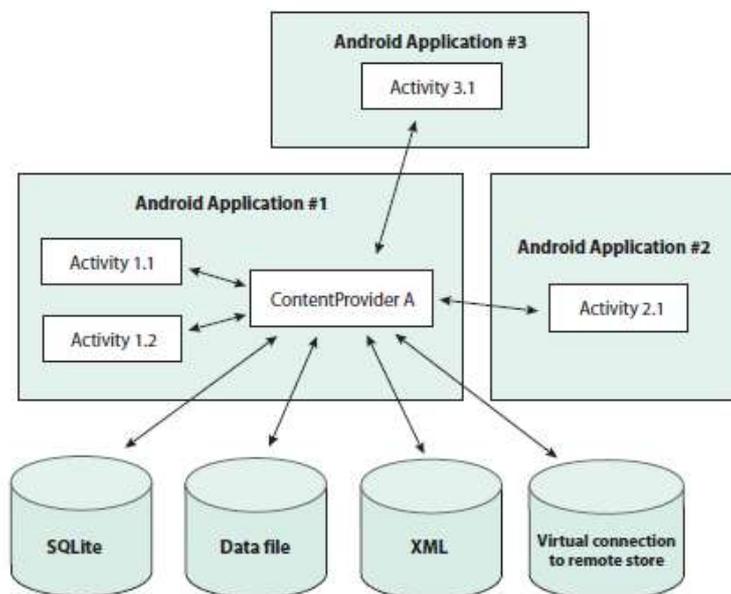
ContentProvider.- Permite gestionar datos entre otras aplicaciones que se ejecuten en el entorno Android, implementa un conjunto de métodos para acceder a una base de datos ya sea para leer o escribir operaciones.

Un *ContentProvider* puede proporcionar datos a una actividad o servicio de la misma aplicación, así como también de otras aplicaciones. Utiliza cualquier tipo de mecanismo de almacenamiento como pueden ser base de datos SQLite, archivo de datos, XML, e incluso mapear archivos basados en la memoria.

No se aconseja compartir archivos o base de datos entre aplicaciones por motivo de seguridad que considera el sistema operativo Linux.

El acceso a un *ContentProvider* es similar al uso de consultas de lenguaje estructurado SQL.

Figura 2.6 *ContentProvider* como gestor de base de datos



Nota: El *ContentProvider* es el nivel de datos que permite acceder y compartir información entre aplicaciones. Obtenida del libro *Android in Action* (Enero 2010)

BroadcastReceiver.-Permite recibir intentos de un *sendBroadcast ()* o responder a un evento global, como puede ser un mensaje de texto o un timbre de teléfono. Puede recibir intentos de las siguientes maneras:

- Puede implementar un elemento `<receiver>` en el archivo *AndroidManifest.xml* en el cual describe el nombre de la clase *BroadcastReceiver* los enumera con *IntentFilters*. Un *IntentFilter* es un descriptor de intentos que la aplicación desea procesar. Cuando el receiver está registrado no necesita estar en ejecución para

ser activado, porque cuando se produce el evento la aplicación se inicia automáticamente después de la notificación.

- Se puede registrar en tiempo de ejecución a través de la clase *Context* con el método *registerReceiver*.

Figura 2.7 Ejemplo básico de un **BroadcastReceiver**

```
package com.example.formulario;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.util.Log;

public class miCorreo extends BroadcastReceiver {
    public static final String tag = "miCorreo";
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.i(tag, "onReceive");
        if (intent.getAction().equals
            ("android.provider.Telephony.SMS_RECEIVED")) {
            Log.i(tag, "Found our Event!");
        }
    }
}
```

Nota: La figura se trata de la activación del BroadcastReceiver en la recepción de mensajes de textos entrantes. Elaborado por los Autores. (Junio 2013)

Algunas de sus características de este sistema operativo se pueden mencionar:

- De código abierto
- No hay diferencias entre aplicaciones incorporadas y las aplicaciones que se crean con el SDK.
- Sus aplicaciones corren en múltiples dispositivos móviles, debido a su compatibilidad de versiones.
- Soporte de multimedia como son archivos de audio, video e imágenes.
- Permite a las aplicaciones usar las características del teléfono.
- Soporte para dispositivos móviles táctiles.

Versiones de la plataforma Android

Desde que se creó el sistema operativo Android hasta la actualidad ha lanzado algunas versiones, con el objetivo de mejorar el rendimiento y la interfaz para el usuario, entre las cuales podemos mencionar las más importantes:

- **Eclair.-** Se aumentó la resolución en la pantalla en cuanto a la versión anterior Donut, debido a los nuevos dispositivos móviles. Incluyo ser multitouch y el uso de gestos para interactuar con el usuario.

- **Froyo.-** Se mejoró el rendimiento, soporte a Adobe Flash. Permitió a los desarrolladores el uso de la memoria externa SD para instalar aplicaciones.

- **Gingerbread.-** Se mejoró la velocidad en la interfaz con el usuario. Se presentó el NFC (*Nearfield Communication*) en los dispositivos móviles Android, para permitir el intercambio de datos a través de la comunicación inalámbrica.

- **Honeycomb.-** Pensada y diseñada para tablets. Donde cabe destacar la conectividad USB y el soporte de teclados.

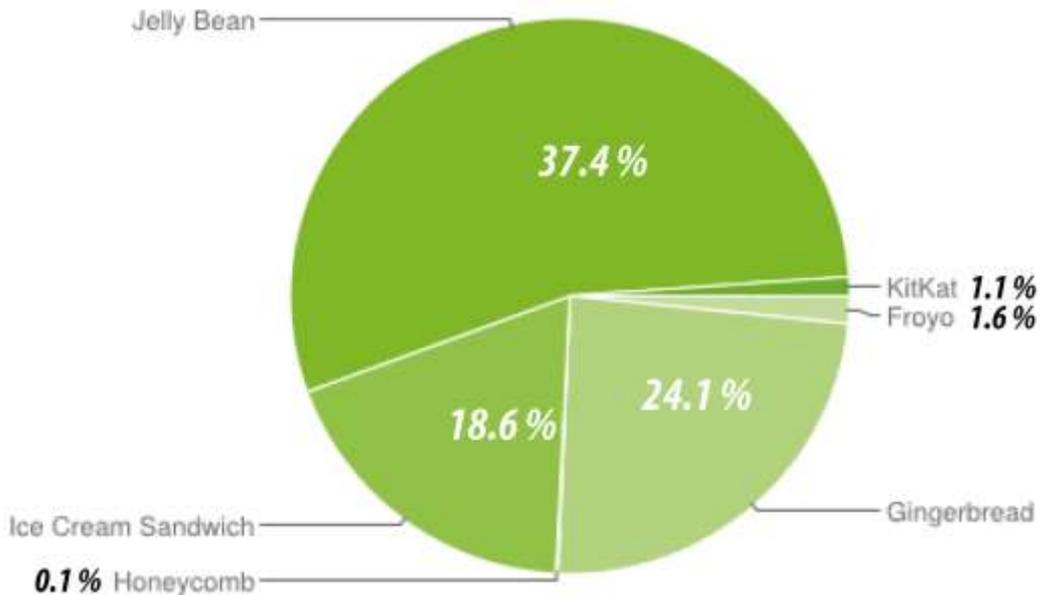
- **Ice Cream Sandwich.-** Trajo novedades en la interfaz y nuevas carpetas para accesos directos. Reunificación entre las versiones para Smartphone y tablets.

- **Jelly Bean.-** generar fotos de 360 grados y la inclusión de Google Now. Además permite poseer varias cuentas en un dispositivo pero solo en tablets. En el teclado se añade escritura por gestos. (Condesa, Androideity, 2012)

- **Kit Kat.-** Da prioridad a contactos en función de las personas que más te llaman. Al tiempo que permite realizar búsquedas de lugares cercanos, negocios y contactos.

Cabe mencionar que todas las versiones de Android oficiales al ser código abierto puedes ser modificadas por terceros agregando así múltiples funciones y características que no poseen.

Figura 2.8 Porcentaje de distribución de las versiones de Android



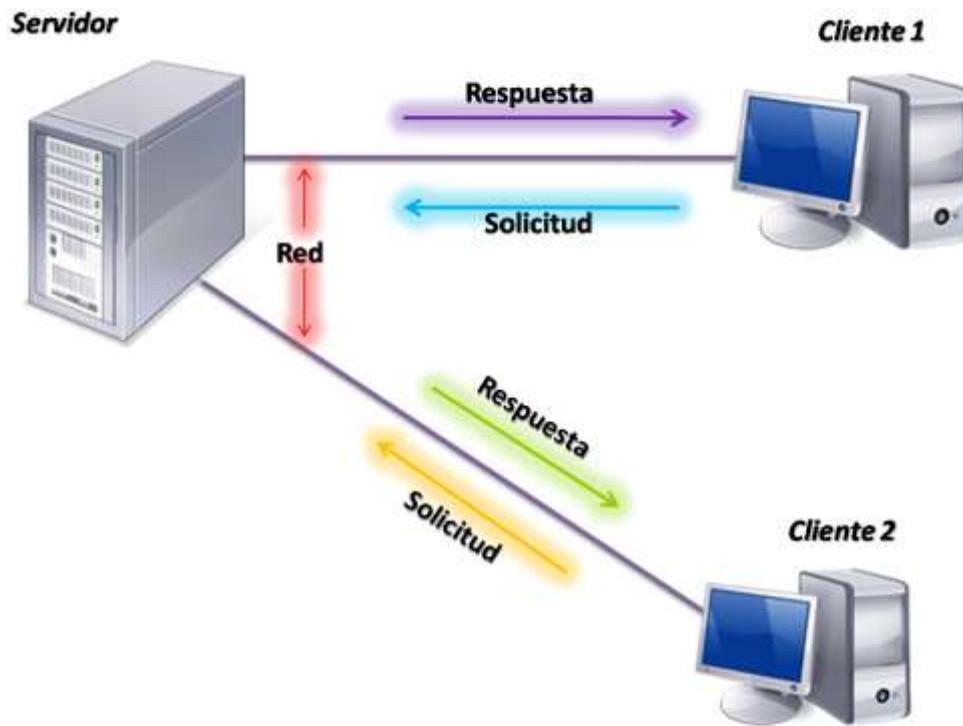
Nota: Datos recogidos durante 7 días el cual refleja la distribución de sistema operativo a nivel mundial. Obtenido de <http://developer.android.com/> (Enero 2014).

2.1.2 Arquitectura Cliente – Servidor

La arquitectura cliente-servidor consiste básicamente en un cliente que realiza peticiones a otro programa el cual da una respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora, es mucho más ventajosa ya que es un sistema operativo multiusuario (distribuido) a través de una red de computadoras.

Este tipo de arquitectura facilita bastante el manejo de información ya que al ser distribuido si alguna maquina sufre algún tipo de percance los datos no se pierden, el mantenimiento del sistema se puede hacer en versiones e ir actualizando de acuerdo a los cambios de las versiones.

Figura 2.9 Diseño de una arquitectura Cliente - Servidor



Nota: Se observa en esta arquitectura como el cliente 1 se comunica con el cliente a través del servidor. Obtenido en <http://eddieramos.wordpress.com/2010/02/01/35/> por Eduardo Ramos. (Febrero 2010)

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc.

“Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.” (Jose Guillermo Valle, 2005)

La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta y que los pone a disposición de los clientes cada vez que estos son solicitados.

Otra parte importante es el middleware que se encarga del transporte de los mensajes del cliente y servidor, facilitando el desarrollo de las aplicaciones en sistemas heterogéneos sin usar tecnologías propietarias.

Las características del cliente son las siguientes:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo maestro o amo).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Las características del servidor son las siguientes:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Ventajas y Desventajas

Entre sus principales ventajas:

- **Escalabilidad.**- Se puede añadir nuevos nodos a la red.
- **Fácil Mantenimiento.**- Posee facilidad de reparar, actualizar, reemplazar un servidor.
- **Diversidad en Tecnologías.**

Y se puede mencionar las siguientes desventajas:

- El cliente no dispone de los recursos que puedan existir en el servidor.
- Se requiere de un gran servidor para poder comunicarse con grandes cantidades de clientes.
- Cuando un nodo que interviene en la comunicación se cae puede provocar un incorrecto funcionamiento en el sistema, e incluso deshabilitarlo

2.1.3 Modelo Vista Controlador (MVC)

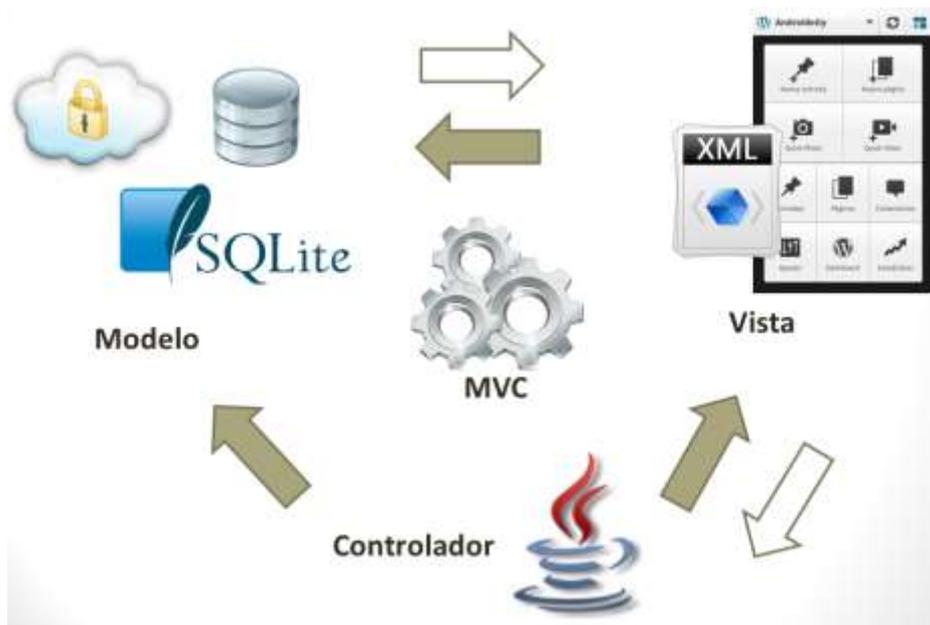
“La arquitectura MVC consiste en separar los datos de la aplicación, la interfaz del usuario, y la lógica de negocios en tres componentes distintos que se relacionarán para al final tener como resultado nuestra aplicación.” (Condesa, Androideity, 2012)

De esta forma podemos seccionar de forma más fácil nuestro equipo de trabajo y dedicarnos a desarrollar nuestros componentes de tal forma que construyamos módulos o librerías con funcionalidades específicas que incluso podríamos reutilizar en proyectos posteriores y no simplemente en el proyecto actual.

Para ello MVC propone la construcción de tres componentes como son: el modelo, la vista, y el controlador. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones para después tener un buen mantenimiento.

Este modelo facilita el desarrollo de un sistema, todas las actualizaciones sean estas visuales o de programación podrán ser por separado y podrán también ponerse en ejecución de diferentes maneras; los cambios visuales para mejoras de los sistemas al usar este tipo de arquitectura se pueden hacer independientemente sin afectar el sistema actual ni la información que este manejando el sistema.

Figura 2.10 Modelo Vista Controlador en Android



Nota: Establece comunicación e interactúa por medio del modelo MVC (Modelo Vista Controlador). Obtenido en <http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/> por Condesa. (Mayo 2012)

Modelo.- Son representaciones que construimos basándonos en la información con la que operará nuestra aplicación. En Java el modelo viene siendo análogo a los beans que pueden ser reutilizados haciendo las aplicaciones más escalables. Además gestiona todos los accesos de la información tanto consultas como actualizaciones. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

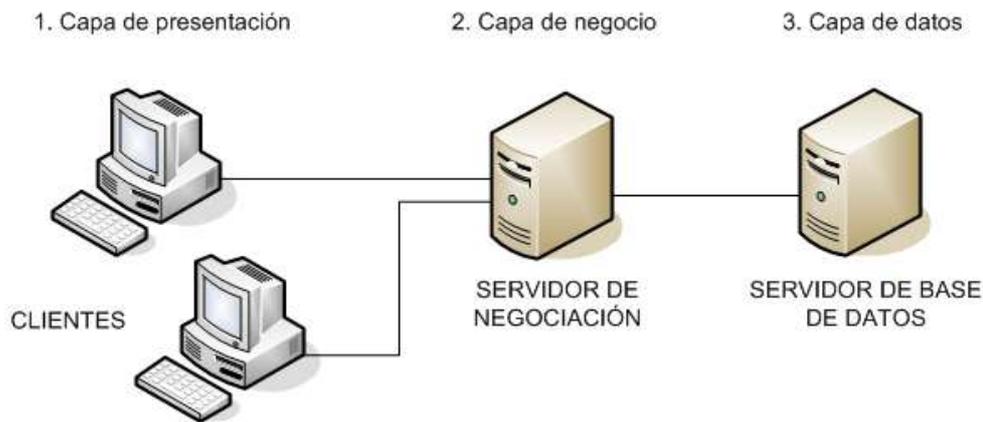
Vista.- Es la interfaz gráfica que interactúa con el usuario. En Android las interfaces se construyen con XML.

Controlador.- Son todas las clases que ayudan a darle vida a las interfaces, para que permitan desplegarse y consumir información entre el usuario. Responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. Estos controladores se programan en lenguaje Java que son el core de la aplicación.

2.1.4 Arquitectura de 3 capas

Es un estilo de programación que se basa en la arquitectura cliente – servidor donde se encarga de separar la lógica de negocio con la interfaz visual que ve el usuario. Este modelo se divide en: capa de presentación, negocio, y datos.

Figura 2.11 Diseño de una Arquitectura de 3 capas



Nota: En esta arquitectura se describe como un usuario final en la capa de presentación procesa información y realiza transacciones a través de la capa de negocio, y esta almacena en la base de datos. Obtenido en <http://arquitecturaencapas.blogspot.com/2011/08/arquitectura-3-capas-programacion-por.html> y publicado por Milagro Paredes. (Agosto 2011)

- **Capa de Presentación:** Proporciona la interfaz visual que los usuarios utilizan para ver la información y los datos. En este nivel, los componentes son responsables de enviar y recibir servicios de otros componentes. Podemos mencionar que las funciones de negocio son transparentes para el usuario.
- **Capa de Negocio:** Como el usuario es este modelo no puede comunicarse directamente con el servicio de datos, es necesidad de la capa de negocio para poder transmitir información de las peticiones o respuestas que espera el usuario desde la base de datos. Los objetos de negocios proporcionan mecanismos o servicios para proporcionar mayor seguridad a la base de datos, como es la verificación de los datos enviados por el usuario.
- **Capa de Datos:** Se encarga de realizar las tareas u operaciones que se encarga la base de datos como son inserción, actualización, eliminación y consulta, en donde es responsable de la gestión de peticiones realizadas por un objeto de negocio.

Entre sus principales ventajas tenemos:

- Pueden ser desarrollados en cualquier lenguaje.
- Los componentes son independientes y pueden alojarse en múltiples servidores.
- Los datos de la base de datos no son vistas por el usuario.
- Permite la reutilización real del software.

2.1.5 Lenguaje de programación PHP

Es un lenguaje de código abierto interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor con las siglas “Hypertext Preprocessor”.

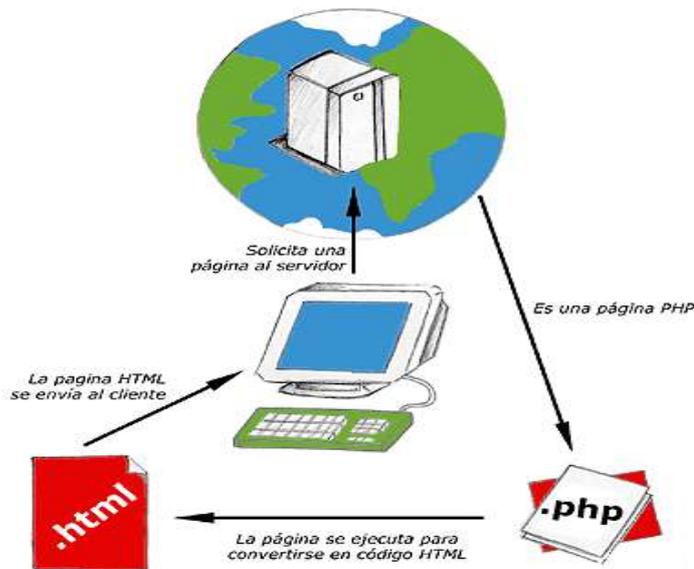
El código PHP se incluye entre etiquetas especiales de comienzo y final, que nos permite entrar o salir del modo PHP.

“Lo que distingue a PHP de la tecnología JavaScript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido.” (Jorge, 2011)

El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP. Comparando con otro tipo de tecnologías similares resulta más rápido, independiente de la plataforma y más sencillo de aprender y utilizar, además de poder interactuar con más de 20 tipo de bases datos.

El intérprete de PHP solo ejecuta el código que se encuentra entre sus delimitadores. Los delimitadores más comunes son `<?php` para abrir una sección PHP y `?>` para cerrarla. El propósito de estos delimitadores es separar el código PHP del resto de código, como por ejemplo el HTML.

Figura 2.12 Esquema de funcionamiento de una página en PHP



Nota: Se describe en este esquema como se solicita procesar una página html a través del servidor el cual responde con otra página html. Obtenido en <http://www.desarrolloweb.com/articulos/392.php> por Miguel Álvarez (Mayo 2001)

Entre sus características más importantes podemos mencionar:

- PHP es un lenguaje interpretado, solo se necesita un navegador web para ejecutarlo.
- Se ejecuta en del servidor, por lo que los script se ejecutan remotamente y el resultado aparece en la máquina cliente (local).
- Tiene soporte para muchos tipos de bases de datos, entre las principales están MySQL, PostgreSQL, SQLite, entre otras.
- Es embebido en código HTML.
- Soporte de orientación a objetos

Es un lenguaje Open-Source implementarlo no tiene ningún costo.

Ventajas

- Es de código abierto.
- Simple y muy fácil de aprender.
- PHP puede ser utilizado en todos los principales sistemas operativos, incluyendo Linux, muchas variantes Unix, Microsoft Windows, Mac OS X, RISC OS, y probablemente otros.
- Una de las mejores características de PHP es que puede ser compatible con la mayoría de las bases de datos.

- PHP también cuenta con servicios de apoyo que utilizan protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows).
- Fácil despliegue y alojamiento rentables.

Desventajas

- Los defectos de seguridad debido a las vulnerabilidades desconocidas.
- Como es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado.
- La ofuscación es una técnica que puede dificultar la lectura del código pero no lo impide, en ciertos casos, representa un costo en tiempos de ejecución.

2.1.6 Motor de Base de Datos MySQL

MySQL es un interpretador de SQL, es un servidor de base de datos, que permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas otras operaciones.

Sus características más importantes:

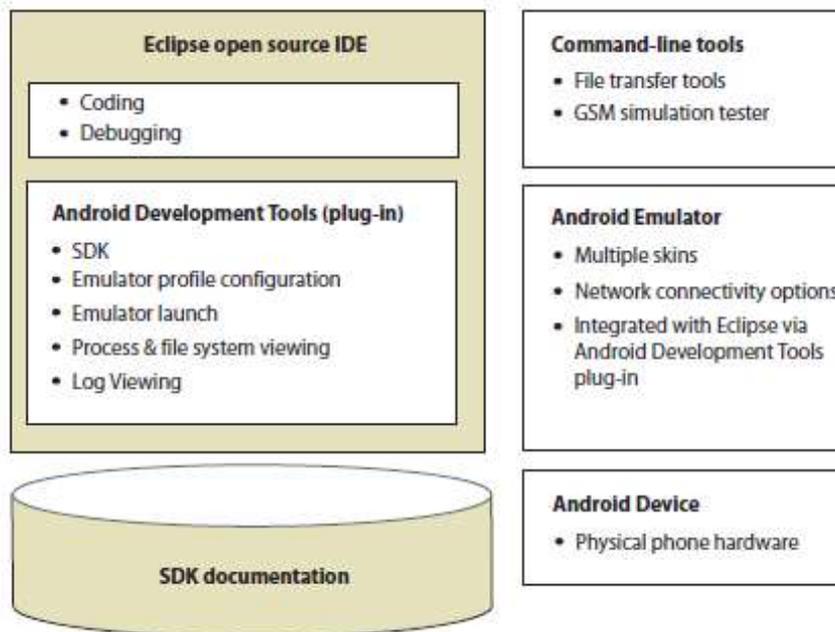
- Uso de transacciones ACID para construir aplicaciones más seguras mediante commit, rollback, crashrecovery y bloqueo por registro.
- Mejora la programación mediante el uso de procedimientos almacenados.
- Mejora las reglas de negocio a través de los triggers.
- Permite usar vistas para que la información sensible sea más segura.
- Fácil acceso a los metadatos.
- Soporta transacciones entre múltiples ambientes de bases de datos.

2.1.7 Kit de Desarrollo de Software de Android (SDK)

El SDK de Android es un conjunto de herramientas para el desarrollo de una aplicación Android, y comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales para facilitar crear aplicaciones.

Se puede descargar gratuitamente del sitio web de Android. Es necesario para la creación de aplicaciones el software Eclipse el cual es el entorno de desarrollo mas usado.

Figura 2.13 Entorno de desarrollo de Android



Nota: Se describe el entorno de desarrollo para crear aplicaciones Android, con su IDE o herramienta de desarrollo como es Eclipse. Obtenida del libro Android in Action. (Enero 2010)

“Aunque Eclipse no es una herramienta exclusiva para el desarrollo de aplicaciones, pero al ser un software open Source es el más utilizado por los desarrolladores. Provee compilación y depuración de código en Java y permite gestionar y controlar en modo de pruebas.” (W.Frank Ableson, 2012)

El entorno de desarrollo en Eclipse presenta las siguientes cualidades:

- Depuración a nivel de fuente
- Ejecuta y gestiona a través del emulador AVD (Android Virtual Devices).
- Gestiona sistema de archivos en el emulador.
- Control del emulador.
- Ingreso en el sistema y la aplicación.

Eclipse cuenta con dos perspectivas importantes, donde el diseño de su pantalla posee de un conjunto de ventanas y herramientas relacionadas. Estas son: perspectiva Java y perspectiva DDMS.

2.1.8 Interfaz de Programación de Aplicaciones (API) y API Google Maps

Se define un API como el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece ciertas bibliotecas para ser utilizado por otro software como una capa de abstracción. Son generalmente usadas en las bibliotecas por que incluye un conjunto de funciones de uso general.

API de Google Maps

El API ofrece clases para realizar descargas, procesamiento y almacenamiento en caché integrados de mosaicos de Google Maps, así como numerosos controles y opciones de visualización.

La clase clave de la biblioteca de Google Maps es `MapView`, una subclase de `ViewGroup` de la biblioteca estándar de Android. Una clase `MapView` muestra un mapa con datos obtenidos del servicio de Google Maps. También proporciona todos los elementos de interfaz de usuario que necesitan los usuarios para controlar el mapa.

Los desarrolladores solo pueden utilizar los métodos de clase `MapView` para controlar mediante programación y dibujar algunos tipos de superposiciones (`Overlay`) en la parte superior del mapa.

Conviene destacar que a diferencia de Android, Google Maps no es un software libre, por lo que está limitado a una serie de condiciones de servicio. Podemos usarlo de forma gratuita siempre que nuestra aplicación no solicite más de 15.000 codificaciones geográficas al día.

Figura 2.14 Aplicación Móvil usando el API de Google



Nota: En la figura se observa el uso del api de Google Maps para el trazo de rutas de un sitio a otro en un teléfono Iphone. Obtenida en <http://moviltoday.com/navegacion-gratuita-con-google-maps-para-iphone/> y publicado por Juan Carlos Magno. (Abril 2010)

Para poder utilizar este servicio es necesario obtener una clave que Google facilita y la cual nos permite también acceder a los demás servicios que Google ofrece los cuales son necesarios para este tipo de aplicaciones y los nombraremos a continuación:

- Google Maps
- Places API.
- Google Maps Tracks API.
- Static Maps API.
- Google Cloud Messaging for Android.

2.1.9 Servicios Web y sus componentes

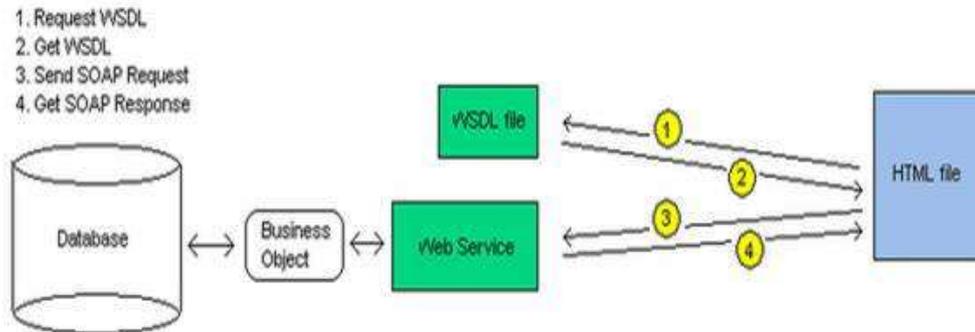
Es un estándar de comunicación entre procesos y o componentes, diseñado para ser multiplataforma y multilinguaje, las cuales pueden ser accesibles y utilizables por otras aplicaciones desarrolladas en otras plataformas o lenguajes de programación.

La gran ventaja es que trae el protocolo HTTP para el transporte de mensaje, especialmente diseñado y optimizado para ser utilizado en redes como Internet.

Entre sus componentes que conforman parte de los WebServices podemos destacar:

WSDL

Figura 2.15 Proceso del WSDL



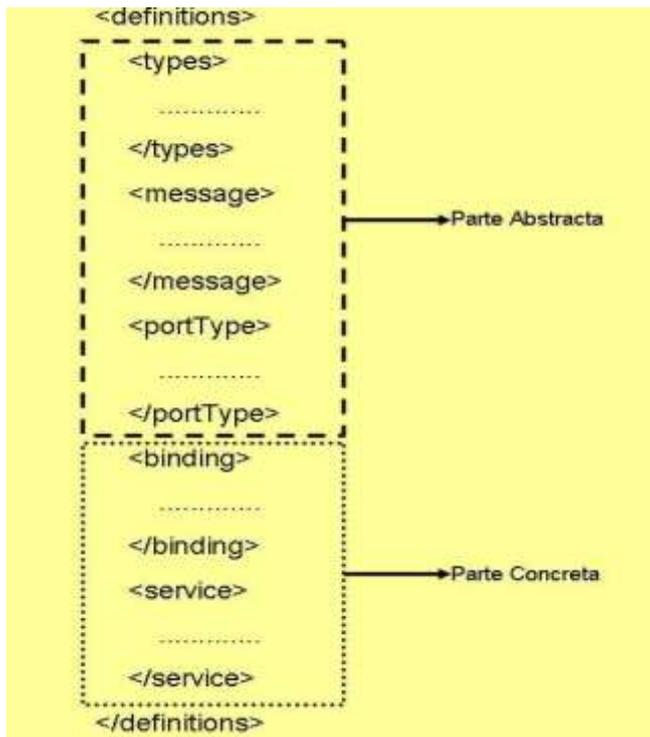
Nota: En la figura se observa como consumir el webservice en el paso (1) realiza la solicitud, en el paso (2) el servidor entrega el archivo WSDL, en el paso (3) el cliente realiza la petición en el formato especificado, y en el paso (4) entrega el resultado. Obtenido en <http://programacion.jias.es/2012/01/web-service-definicion-utilizacion-estructura-del-wsdl/>. (Enero 2012)

El WSDL (*Web Services Description Languages*) es un documento XML que se utiliza para describir mensajes SOAP. Permite describir los métodos y los tipos de parámetros del servicio web publicado.

Todo documento WSDL está compuesto por un elemento raíz llamado **definiciones** que a su vez está compuesto de los siguientes parámetros:

- **Types.**-Define el tipo de esquema a ser utilizado, usualmente **XML**.
- **Message.**-Define los mensajes de entrada y salida de forma abstracta entre el cliente y servidor. Normalmente habrá varios elementos de este tipo para cada protocolo (**HTTP, SOAP**) y para cada **Web Method**.
- **PortType.**- Define los tipos de mensajes a intercambiar entre el cliente y el servidor. Este puede ser: En el cual por cada requerimiento se envía una respuesta, o solo recibe requerimientos pero no envía respuesta, o viceversa.
- **Binding.**- Se establecen las definiciones de los vínculos de los protocolos como SOAP a un tipo de vínculo en particular.
- **Service.**-Se informa el punto de acceso a los servicios por cada uno de los protocolos.

Figura 2.16 Estructura de un documento WSDL

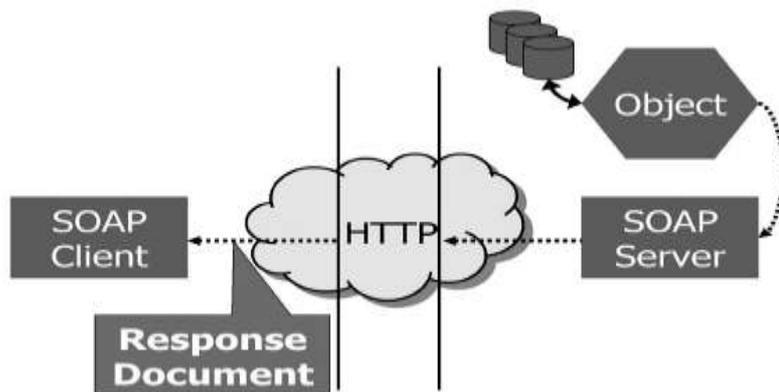


Nota: El documento WSDL se divide en dos componentes en donde la parte concreta define el cómo o el donde. Y la parte abstracta define que hace el servicio con los mensajes. Obtenido en <http://arquitecturaorientadaaservicios.blogspot.com/2006/12/articulo-tecnolgico-wsdl-el-contrato-de.html> y publicado por Antonio Barco. (Diciembre 2006)

SOAP

SOAP (Simple Object Access Protocol) es el protocolo base de los WebServices basado en XML, que se encarga de la comunicación entre mensajes por medio del HTTP.

Figura 2.17 Response del SOAP



Nota: Es la respuesta del servidor el cual es enviado por el HTTP para que pueda ser consumido de un cliente

El SOAP simplifica el intercambio de información y proporciona interoperabilidad a través de una variedad de plataformas, ya que no está ligado a cualquier modelo de objeto específico.

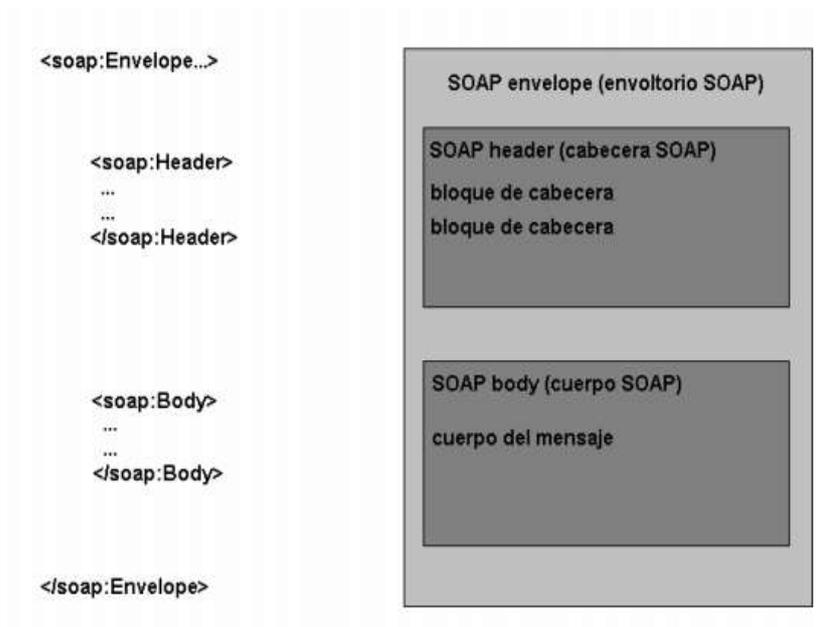
Los mensajes SOAP están compuestos por un tag llamado **Envelope**, que está dividido en cabecera **Header** y un cuerpo **Body**.

Entre sus principales características del SOAP tenemos:

- Muchas operaciones con muchos recursos.
- Se centra en el diseño de aplicaciones distribuidas.
- Protocolo de transporte HTTP, usa XML Schema para su comunicación.
- El servidor puede mantener el estado de la conversación.
- Comunicación de origen a destino segura.
- Se puede implementar en cualquier lenguaje de programación.

Especificaciones del SOAP

Figura 2.18 Estructura de un mensaje SOAP



Nota: Es la respuesta del servidor el cual es enviado por el HTTP para que pueda ser consumido de un cliente SOAP. Obtenido en <http://www.ehu.es/mrodriguez/archivos/csharpdf/ServiciosWeb/WebServices.pdf>. (Septiembre 2013)

- El elemento **Envelope** SOAP es el elemento superior del documento XML que representa el mensaje SOAP. Espacios de nombres XML se utilizan para eliminar la ambigüedad de los identificadores de SOAP a partir de

identificadores específicos de la aplicación. SOAP proporciona un mecanismo flexible para extender un mensaje de una manera descentralizada y modular sin el conocimiento previo entre las partes que se comunican.

- El elemento **Header** se codifica como el primer elemento secundario inmediato del elemento Envelope XML SOAP. Todos los elementos secundarios inmediatos del elemento Header se llaman entradas de cabecera.
- El elemento **Body** proporciona un mecanismo sencillo para el intercambio de información obligatoria prevista para el destinatario final del mensaje. Los usos típicos del elemento Body de maniobras incluyen llamadas RPC y los informes de error. El elemento Body se codifica como un elemento secundario inmediato del elemento Envelope XML SOAP.

Si un elemento de encabezado está presente, entonces el elemento de cuerpo debe seguir inmediatamente el elemento Header, de lo contrario, debe ser el primer elemento secundario inmediato del elemento Envelope.

Cuando se trata del envío de mensajes, se cuenta con una amplia variedad de opciones que van, desde el envío de una solicitud y la espera de una respuesta, al envío de una solicitud y olvidarse de ella, hasta el envío de una solicitud y hacerla pasar a través de intermediarios múltiples antes de que llegue a su destino final.

No obstante, al final, sólo tiene dos opciones:

- **Solicitud/Respuesta:** Según el patrón Solicitud/Respuesta, se envía una solicitud con forma de mensaje SOAP y simplemente espera la llegada de una respuesta. La solicitud puede ser sincrónica o asincrónica.
- **Mensajería unidireccional:** Este caso, también conocido como método “dispare y olvídense”, implica el envío de la solicitud para después olvidarse directamente de ella. Esto lo puede usar en situaciones en las que simplemente

imparte información, o en alguna otra en la que, en realidad, no le importa lo que el destinatario pueda decir de ella.

Las reglas de codificación para las entradas de cabecera son de la siguiente manera:

1. Una entrada de la cabecera se identifica por su nombre de elemento completo, que consiste en la URI de espacio de nombres y el nombre local. Todos los elementos secundarios inmediatos del elemento Header SOAP debe ser namespace.
2. El atributo de SOAP encoding Style puede ser usado para indicar el estilo de codificación utilizado para las entradas de cabecera.
3. El atributo agente de SOAP puede utilizarse para indicar la forma de procesar la entrada.

Ventajas

- Es sencillo de implementar, probar y usar.
- Atraviesa firewalls y routers, ya que lo consideran como una comunicación HTTP.
- Usa XML para describir los datos y funciones.
- Es independiente al sistema operativo y procesador.
- Facilidad para utilizar en cualquier lenguaje.
- No se encuentra asociado a ningún protocolo de transporte.

Desventajas

- La utilización de SOAP recaen en la dificultad para entender las especificaciones del protocolo, puesto que es un complejo esquema de codificación en el cual es necesario precisar que todos los mensajes se incluyan en un sobre, con el contenido del mensaje dentro de un elemento de cuerpo para que puedan ser entendidos por cada una de las aplicaciones Web que procesan el mensaje.

- SOAP convierte en opcionales elementos como encabezados y ofrece un amplio margen con respecto a lo que se puede incluir en el elemento de cuerpo y además cambia los nombres de métodos en etiquetas secundarias del cuerpo y los argumentos en etiquetas secundarias del nombre del método, lo que puede generar ciertos problemas de interoperabilidad.

2.1.10 Fórmula Haversine

Es una ecuación que nos permite obtener la distancia entre dos puntos geográficos donde interviene la latitud y longitud. La función se expresa en términos de funciones Haversine o múltiplos de esta y se define así:

Figura 2.19 Ecuación de Haversine

$$\text{haversine}\left(\frac{D}{R}\right) = \text{haversine}(\phi_1 - \phi_2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \text{haversine}(\Delta\lambda)$$

Nota: Formula para el calculo de puntos geográficos de una circunferencia.

Dónde:

Figura 2.20 Nomenclatura de la ecuación de Haversine

D = distancia entre los 2 puntos
R = radio de la esfera
φ = latitud
λ = longitud
Δφ = diferencia de latitud
Δλ = diferencia de longitud

Nota: Nomenclatura basada en la fórmula de Haversine. Obtenido del trabajo de presentación por Ariel Iván Díaz. (Enero 2012)

Al despejar la distancia de la fórmula de haversine se obtiene una nueva ecuación, que presenta algunos problemas para ciertos hemisferios, para solucionar estos se propone la fórmula directa de la distancia de haversine:

Figura 2.21 Fórmula directa de la distancia

$$\begin{aligned}\Delta\phi &= \phi_1 - \phi_2 \\ \Delta\lambda &= \lambda_2 - \lambda_1 \\ A &= \sin^2\left(\frac{\Delta\phi}{2}\right) \cdot \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \\ C &= 2 \cdot \text{atan2}(\sqrt{A}, \sqrt{1-A}) \\ D &= R \cdot C\end{aligned}$$

Nota: Fórmula directa de la distancia basada en la fórmula de Haversine. Obtenido del trabajo de presentación por Ariel Iván Díaz. (Enero 2012)

2.1.11 Sistema de posicionamiento global (GPS)

Las GPS (Sistema de Posicionamiento Global), podemos definir como un Sistema Global de Navegación por Satélite (GNSS) que nos permite fijar a escala mundial la posición de un objeto, una persona, un vehículo o una nave, etc.

La precisión del GPS puede llegar a determinar los puntos de posición con errores mínimos en centímetros, aunque en la práctica hablamos de metros. El sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

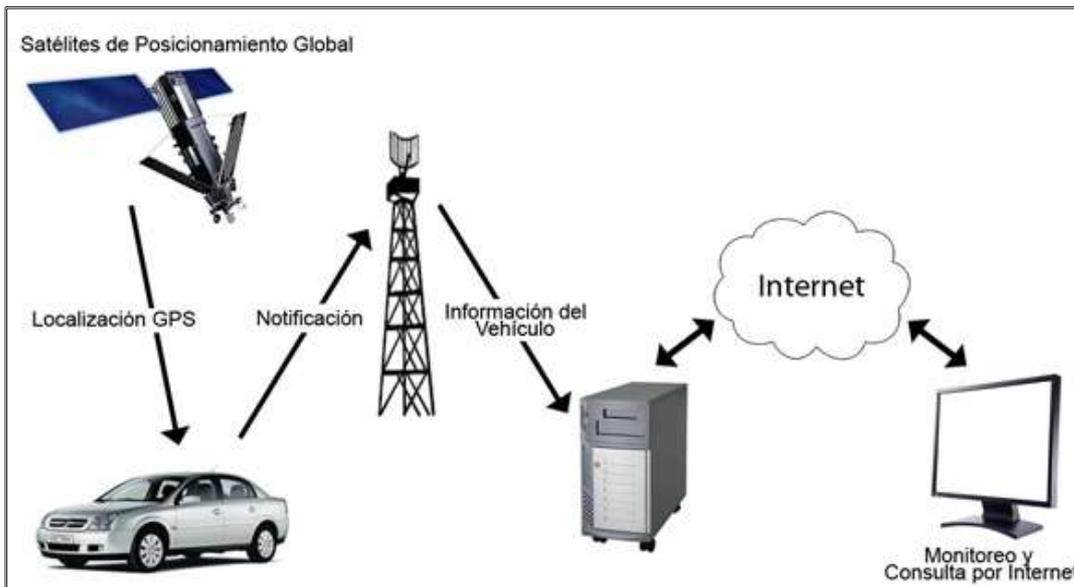
Determinar una posición en la Tierra significa proporcionar la latitud y longitud del punto en el que nos encontramos sobre la superficie terrestre. Por tanto, la mayoría de receptores proporcionan los valores de estas coordenadas en unidades de grados (°) y minutos ('). Tanto la latitud como la longitud son ángulos y por tanto deben medirse con respecto a un 0° de referencia bien definido.

Funcionamiento del Sistema GPS

Para fijar una posición, el navegador GPS localiza automáticamente como mínimo 4 satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada satélite.

El navegador GPS sincroniza su reloj y calcula el retraso de las señales (que viene dado por distancia al satélite), calculando la posición en que éste se halla. (Vargas, 2011)

Figura 2.22 Funcionamiento del GPS



Nota: Muestra un dispositivo móvil que puede ser localizado a través del GPS. Publicado desde <http://www.gps-i.com.mx/como-funciona-un-gps.php>. (Septiembre 2013)

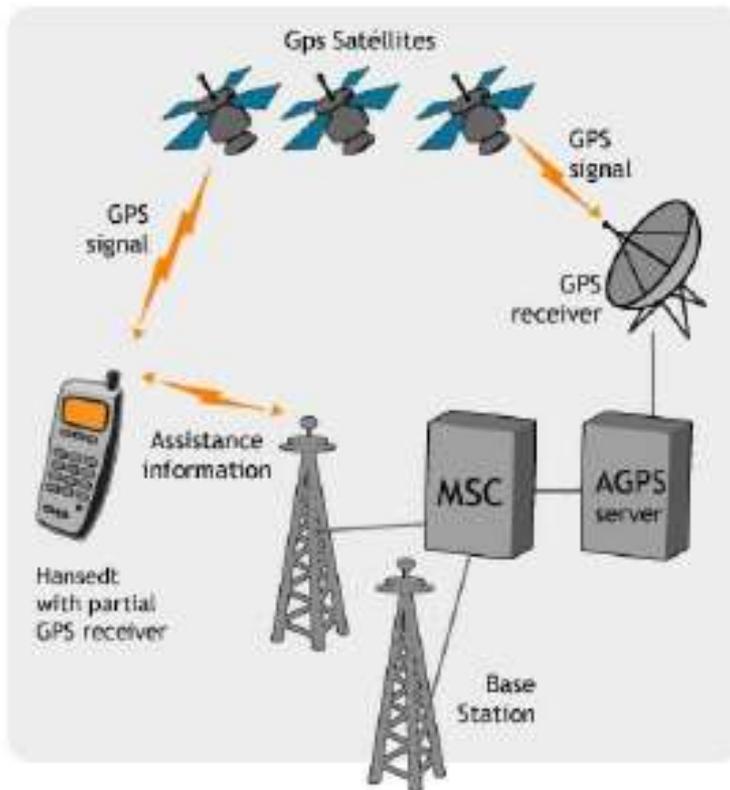
Las distancias estimadas se fijan con facilidad la propia posición relativa del GPS respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene las posiciones absolutas o coordenadas reales del punto de medición.

Integración con la telefonía móvil del GPS

Actualmente dentro del mercado de la telefonía móvil la tendencia es la de integrar, por parte de los fabricante, la tecnología GPS dentro de sus dispositivos.

El uso y masificación del GPS está particularmente extendido en los teléfonos móviles Smartphone, lo que ha hecho surgir todo un ecosistema de software para este tipo de dispositivos, así como nuevos modelos de negocios que van desde el uso del terminal móvil para la navegación tradicional punto-a-punto hasta la prestación de los llamados Servicios Basados en la Localización (LBS).

Figura 2.23 Integración del GPS en el dispositivo Móvil



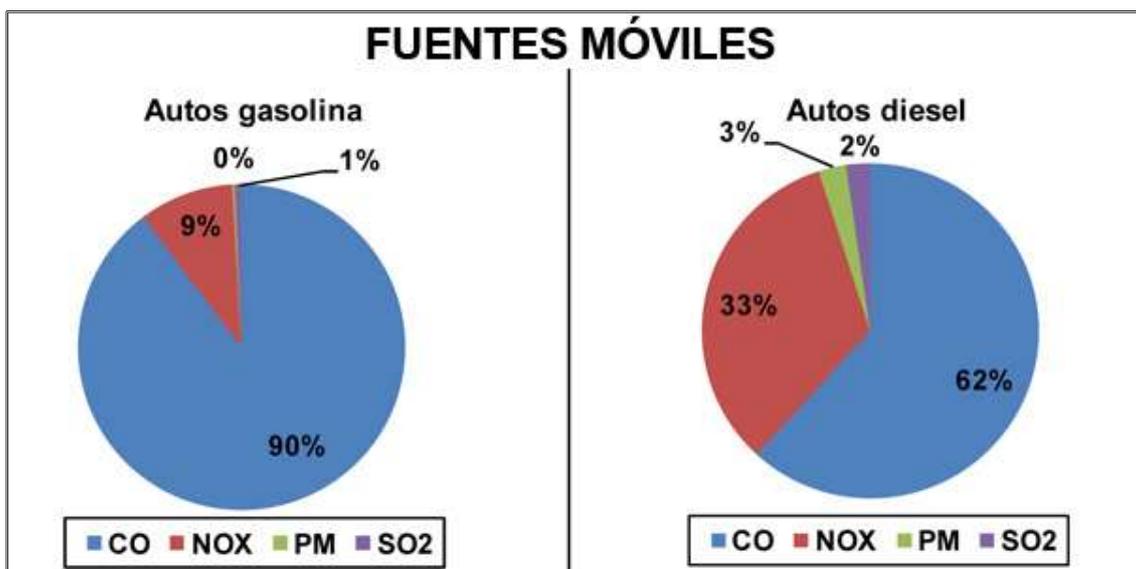
Nota: Muestra como un terminal móvil puede ser localizado mediante un dispositivo receptor de GPS. Obtenido en <http://pacorel.blogspot.com/2009/11/que-es-el-gps.html> (Noviembre 2012)

2.1.12 Contaminación Ambiental por emisión de gases de los automóviles

En la actualidad en todo el mundo, se liberan millones de toneladas contaminantes en la atmósfera por los vehículos de motor. En muchas zonas urbanas, los vehículos son los principales responsables de la presencia de ozono a nivel del suelo.

Las emanaciones de los automóviles son causantes de casos de cáncer en el país, además de problemas medioambientales, tales como la lluvia ácida y el calentamiento global del planeta.

Figura 2.24 Porcentaje de la contaminación por los automóviles



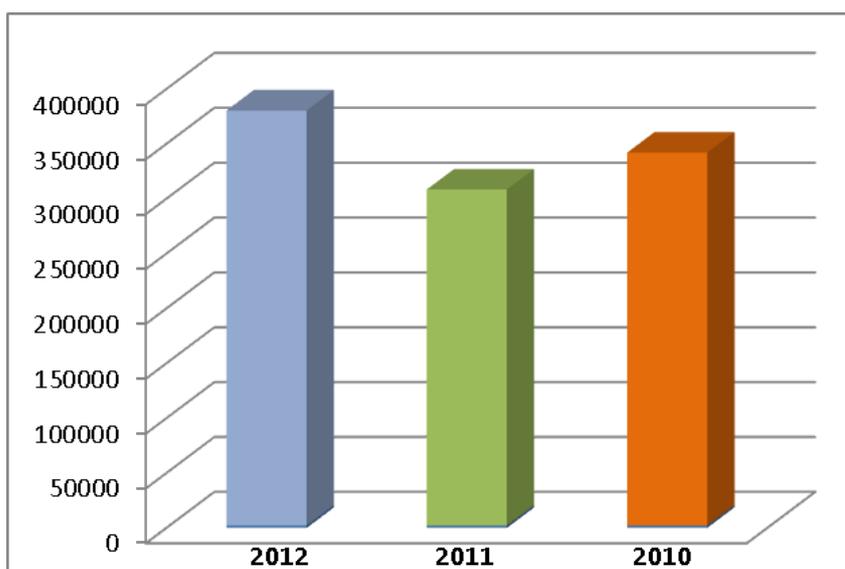
Nota: Muestra el porcentaje de emisión de gases por parte de los automotores como son CO (Monóxido de Carbono), NOX (Óxidos de Nitrógeno), PM (Contaminación Electromagnética), SO2 (Dióxido de Azufre). Obtenido en <http://especiales.elcomercio.com/2012/06/contaminacion/>. (Noviembre 2012)

Los vehículos emiten tres contaminantes principales: hidrocarburos, óxidos de nitrógeno y monóxido de carbono.

- Los hidrocarburos reaccionan con los óxidos de nitrógeno por acción de la luz solar y a temperaturas elevadas, formando ozono a nivel del suelo, el cual puede causar irritación de los ojos, tos, insuficiencia respiratoria y trastornos pulmonares permanentes.
- Los óxidos de nitrógeno (NOX) contribuyen a la formación de ozono y de lluvia ácida, además de afectar la calidad del agua.
- El monóxido de carbono es un gas incoloro letal que reduce el flujo de oxígeno en el torrente sanguíneo y puede afectar las funciones cerebrales y la visión. En las zonas urbanas, los vehículos son responsables de nada menos que el 90% del monóxido de carbono del aire. (Suarez, 2007)

El incremento del número de vehículos es muy importante en este problema pero el gobierno está vigilando la calidad de los combustibles que nos llega para su distribución, tomando medidas como gasolina sin plomo, gasolina de buen octanaje y diésel Premium, para mejorar sustancialmente el medio ambiente.

Figura 2.25 Porcentajes de número de vehículos a Nivel Nacional



Nota: Muestra el porcentaje a nivel nacional de los números de vehículos matriculados en los últimos tres años. Elaborado por los Autores. (Septiembre 2013)

2.2 Marco Conceptual

ARM. - (*Advanced RISC Machine*). Es una arquitectura de procesadores RISC de 32 bits desarrollada por ARM Limited, que es ampliamente empleada en sistemas integrados, por sus características de ahorro energético, las CPU ARM dominan en el mercado de dispositivos móviles, donde el bajo consumo de energía es un objetivo crítico de sus diseños.

Abstracción. - Consideración aislada de las cualidades esenciales de un objeto, o del mismo objeto en su pura esencia o noción.

ACID. - (*Atomicity Consistency Isolation Durability*). Es un conjunto de características o propiedades que garantizan que las transacciones en una base de datos son fiables.

API. - (*Application Programming Interface*). Grupo de rutinas que provee un sistema operativo, una aplicación o una biblioteca, que definen cómo invocar desde un programa un servicio que éstos prestan. En otras palabras, una API representa un interfaz de comunicación entre componentes software.

AVD. - (Android Virtual Devices). Es una configuración del emulador que permite modelar un dispositivo real mediante la definición de las opciones de software que emula el emulador de Android y hardware.

Bean. - Es un componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno.

COM. - (Component Object Model). Es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993.

Commit. - Comando SQL para ejecutar una operación en la base de datos.

Compatibilidad. - Es la condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente como indirectamente.

Crashrecovery. - Permite interrumpir las transacciones en la base de datos cuando se produce algún error.

DDMS. - (Dalvik Debug Monitor Server). Es un monitor de depuración de la máquina virtual Dalvik.

Emulador. - Es un software que permite ejecutar programas o videojuegos en una plataforma (sea una arquitectura de hardware o un sistema operativo) diferente de aquella para la cual fueron escritos originalmente.

Evento. - Es un conjunto de resultados posibles que se pueden dar en un experimento aleatorio.

Firewall. - Es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

GPS. - (Global Positioning System). Se trata de un sistema global de navegación por satélite (GNSS) que permite localizar con precisión un dispositivo GPS en cualquier lugar del mundo.

HTTP. - (Hypertext Transfer Protocol). Es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

IMAP. - (Internet Message Access Protocol). Es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

Interfaz. - Hace referencia al conjunto de métodos para lograr interactividad entre un usuario y una aplicación.

Interoperabilidad. - Característica de los ordenadores que les permite su interconexión y funcionamiento conjunto de manera compatible.

Java. - Es un lenguaje de programación por objetos creado por Sun Microsystems, Inc. que permite crear programas que funcionan en cualquier tipo de ordenador y sistema operativo.

LBS. - (Location Based Services). Hacen referencia a Servicios Basados en Localización o para algunos autores simplemente Servicios de Localización.

LDAP. - (Lightweight Directory Access Protocol). Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Linux. - Es un sistema operativo de código abierto es decir que cualquier persona pueda estudiarlo, usarlo, modificarlo y redistribuirlo.

Log. - Es un registro oficial de eventos durante un rango de tiempo en particular.

Mapear. - Se trata de investigar, explorar, localizar o analizar algún objeto.

MVC. - (Model View Controller). Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Namespace. - Son espacios de nombres el cual permite que se analice un elemento dado.

NFC. - (Near Field Communication). Permite realizar pagos con el móvil o compartir información acercando la tapa del teléfono a un dispositivo destinado a tal efecto, como con las tarjetas sin contacto típicas de los autobuses urbanos.

NNTP. - (Network News Transport Protocol). Es el Protocolo de red utilizado por el Usenet internet Service. Es un Protocolo de red basado en tiras de textos enviados sobre canales TCP de 7 bit ASCII. Es usado para transferir artículos entre servidores.

Nodo. - Es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar.

Paradigma. - Provee la visión y métodos de un programador en la construcción de un programa o subprograma.

Patrón. - Se trata de un modelo específico a seguir.

PHP. - (Hypertext Pre-Processor). Es un lenguaje de programación usado generalmente en la creación de contenidos para sitios web.

PostgreSQL. - Es un sistema de gestión de base de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Protocolo. - Es un método establecido de intercambiar datos en Internet. Un protocolo es un método por el cual dos ordenadores acuerdan comunicarse, una especificación que describe cómo los ordenadores hablan el uno al otro en una red.

Rollback. - Es un comando SQL para realizar un reverso a las transacciones realizadas en la base de datos.

RPC. -(Remote Procedure Call). Protocolo que permite a una aplicación en una computadora ejecutar código en otra remota, sin tener que preocuparse por la comunicación entre ambas.

SD. - (Secure Digital). Es un formato de tarjeta de memoria flash desarrollado por Matsushita, SanDisk y Toshiba

SDK. -(Software Development Kit). Es un conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consulta de videojuego, sistema operativo o similar.

Servidor. - Un servidor es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.

SNMP. - (Simple Network Management Protocol). Permite supervisar, analizar y comunicar información de estado entre una gran variedad de hosts, pudiendo detectar problemas y proporcionar mensajes de estados.

SOAP. - (Simple Object Access Protocol). Es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP.

SQL. -(Structured Query Language). Lenguaje utilizado para base de datos desarrollado entre 1974 y 1975 en IBM Research.

SQLITE. - Es una base de datos liviana, que es usada para propósitos de almacenamiento de datos.

Terminal. - Es un dispositivo electrónico o electromecánico de hardware usado para introducir o mostrar datos de una computadora o de un sistema de computación.

Triggers. - Es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación.

Unix. - Sistema operativo multiplataforma, multitarea, y multiusuario desarrollado originalmente por empleados de Bell de AT&T.

URI. - (Uniform Resource Identifier). Identificador de recursos se suele utilizar como sinónimo de URL.

USB. - (Universal Serial Bus). Se trata de un concepto de la informática para nombrar al puerto que permite conectar periféricos a una computadora.

VM. - (Virtual Machine). Es un software que simula a una computadora y puede ejecutar programas como si fuese una computadora real.

WSDL. - (Web Services Description Language). Lenguaje basado en XML para describir servicios web.

XML. - (Extensible Markup Language). Su objetivo es conseguir páginas web más semántica. XML separa la estructura del contenido y permite el desarrollo de vocabularios modulares. Se trata de un formato abierto.

2.3 Formulación de la hipótesis y variables

2.3.1 Hipótesis general

La utilización de una red social móvil aplicada al sistema de restricción vehicular contribuirá a mejorar los problemas de transportación terrestre.

2.3.2 Hipótesis particulares

- La utilización de una red social móvil ayudará a la cooperación entre las personas.

- El uso de la tecnología móvil mejora la portabilidad y la comunicación a través de la nube.

- La utilización de servicios de datos de alta velocidad mejorara el tiempo de respuesta en el consumo de datos, y ayudara a promover el desarrollo de aplicaciones más complejas y robustas a diferentes aspectos de la sociedad.

2.4 Matriz Causa-Efecto

Tabla 2.1 Matriz Causa - Efecto

Problema general	Objetivo general	Hipótesis general
¿Cómo una aplicación móvil podrá beneficiar a los medios de transporte terrestre y colaborar con el sistema de restricción vehicular?	Crear una aplicación móvil para dispositivos con sistema operativo Android que nos permita mejorar y facilitar el sistema de transportación terrestre que colabore con la restricción vehicular a través de una red social de amigos para así llegar al lugar que se desee de una manera rápida, segura y colaborativa.	La utilización de una red social móvil aplicado al sistema de restricción vehicular ayudará a disminuir la cantidad de vehículos en las horas picos, ya que las personas podrán compartir su vehículo.
Problemas específicos	Objetivos específicos	Hipótesis específicas
¿Cómo fomentar la cooperación de las personas al sistema de restricción vehicular y reducir la contaminación?	Se logrará fomentar el uso adecuado del flujo vehicular con la ayuda de la aplicación móvil impulsando con esta la interacción y cooperación	La utilización de una red social móvil ayudará a la cooperación entre las personas.

<p>¿Qué aportes dará el uso de esta red social móvil en el país?</p>	<p>de las personas, ayudando a reducir el congestión vehicular y la contaminación ambiental.</p> <p>Realizar una aplicación orientada a la última tendencia tecnológica como lo son las aplicaciones móviles y el lenguaje de programación Android, permitiendo incentivar a las personas el desarrollo de nuevos aportes tecnológicos a la sociedad y mediante estos mejorar el estilo de vida de las personas.</p>	<p>El uso de la tecnología móvil mejora la portabilidad y la comunicación a través de la nube.</p>
<p>¿Qué influencia positiva tendrá el avance tecnológico móvil ante la sociedad ecuatoriana?</p>	<p>La tecnología cada día nos ayuda a vivir mejor, al contar con teléfonos inteligentes con mayores prestaciones influirá a que nuestra sociedad pueda aprovechar todos los recursos que estos dispositivos poseen de una mejor manera, ya que día a día se presentan muchas necesidades en diversos aspectos social, económico, ambiental, entre otros.</p>	<p>La utilización de servicios de datos de alta velocidad mejorará el tiempo de respuesta en el consumo de datos, y ayudara a promover el desarrollo de aplicaciones más complejas y robustas a diferentes aspectos de la sociedad.</p>

2.5 Variables e indicadores

Variables Dependientes

- Acceso a internet
- Ser móviles con sistema Android

Variables Independientes

- Nivel socioeconómico del usuario (El usuario necesitará tener un teléfono celular inteligente con acceso a internet para poder usar el aplicativo).
- Exigencia de calidad (El aplicativo debe de brindar información confiable a los usuarios).

CAPITULO 3

MARCO METODOLOGICO DE LA INVESTIGACIÓN

3.1 Tipo de estudio

Tipo de Investigación

La investigación nos ayuda a descubrir nuevos acontecimientos, hechos o leyes sobre algún campo del conocimiento.

En donde el investigador debe elegir y describir el diseño que le permita responder a las diferentes situaciones presentadas en su investigación.

Estos se clasifican según el objetivo del estudio, tipo de intervención en relación del tiempo, y la forma en que se recogen los datos:

- Según el objetivo de estudio puede ser: descriptivo, analítico, predictivo.
- Según el tiempo pueden ser: transversales, longitudinales.
- Según la intervención: observacionales, experimentales.

A partir de lo mencionado podemos definir que el tipo de investigación que estamos aplicando es predictiva, porque se analizará el estudio de la situación actual del caos vehicular que se presenta día a día en el Ecuador, y que se está incrementando debido al mayor uso de vehículos particulares, además de saber el impacto positivo que tendrá sobre el medio ambiente.

Lo que se trata es de beneficiar a las personas que usan este medio de transporte debido a la medida aplicada de parte de la Comisión Nacional de Tránsito para controlar este tráfico vehicular, en el cual se propone mediante esta aplicación que estas personas se puedan adaptar a esta medida impuesta.

3.2 Métodos de Investigación

Para nuestro estudio de investigación se realizó un análisis acerca del problema ocasionado por la gran influencia de vehículos livianos, razón por la cual se tomó las medidas correctivas por parte de la Comisión Nacional de Tránsito

Previo a este análisis podemos definir que su estado es crítico, utilizando el **método deductivo**, debido a que se realizó las debidas indagaciones para resolver esta problemática y poder ayudar a adaptarse a esta medida de prevención. La cual pudo resolver en un gran porcentaje la congestión vehicular por medio de esta aplicación móvil.

3.3 Fuentes y Técnicas para la recopilación de información

Las técnicas para recopilar los datos son los siguientes:

- Recolección de datos a través de encuesta digital sobre la aplicación instalada y su aceptación ante la sociedad.

- Recolección de datos a través de pequeñas muestras del porcentaje de accidentes de tránsito y tráfico en la Ciudad.

3.4 Población y Muestreo

Se entiende como **población** al conjunto finito o infinito de elementos con características comunes, que nos ayudará como parte de la investigación para sacar una conclusión.

Se entiende como **muestreo** el obtener una o varias muestras que se extraen de una población. Es decir que representa una parte de la población de la cual es objeto de estudio.

Para esta investigación la población que se tomara en cuenta es la de la ciudad de Guayaquil en donde se estima que el 60% de la congestión vehicular se origina por el aumento del parque automotor, motivo por el cual se satura las calles principales día a día.

Para determinar la muestra aplicaremos la siguiente fórmula:

$$n = \frac{t^2 \times p(1-p)}{m^2}$$

En donde:

n= tamaño de la muestra requerida

t= nivel de confiabilidad (95% valor estándar de 1,96)

p= variabilidad estimada

m= margen de error (6,47% valor estándar de 0,0647)

Considerando la formula y realizando obtenemos la siguiente información:

$$n = \frac{1,96^2 \times 0,6(1-0,6)}{0,0647^2}$$

$$n = \frac{3,84 \times 0,24}{0,00419}$$

$$n = \frac{0,922}{0,00419} \Rightarrow \mathbf{220,4 \text{ encuestas}}$$

3.5 Tratamiento de la Información

La realización de estas encuestas se hará a un grupo de personas una vez que hayan probado todas las bondades de la aplicación móvil. El cual constará de preguntas objetivas acerca de la aceptación de la aplicación como es la interfaz, la navegabilidad, y el tiempo de respuesta.

Además para conocer la opinión de las personas acerca del beneficio que se obtendrá con respecto a la medida de prevención de caos vehicular llamado “Pico y Placa”.

3.6 Resultado Esperado

El resultado que se desea obtener a través de esta investigación es que las personas conozcan el uso de esta aplicación, y el beneficio que dará a la sociedad y al medio ambiente.

Con los resultados obtenidos de las encuestas se podrá determinar la aceptación favorable por parte de las personas ante la medida propuesta por la Comisión Nacional de Tránsito del Ecuador ocurridos en la ciudad de Guayaquil, y dar conocer que con solo el compartir su vehículo podrá reducir la contaminación ambiental, el embotellamiento vehicular y el número de accidentes de tránsito de las horas pico.

CAPITULO 4

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1 Resultados de las encuestas

Esta encuesta fue realizada mediante un formulario web, divulgado a través de las redes sociales y por medio de correos electrónicos enviado a personas de diferente índole, como amigos, compañeros, entre otros.

Las preguntas que se realizaron en la encuesta fueron redactadas tomando en cuenta todos los factores de la problemática, tanto los factores económicos del usuario y los medios de transporte que los usuarios poseen para transportarse de un lugar a otro dentro de sus ciudades.

El resultado de esta encuesta nos ayudara a identificar la problemática, y obtener una información oportuna al evaluar, el beneficio de nuestra aplicación móvil ante la congestión vehicular en nuestro ciudad Guayaquil, a continuación se presentará los resultados de las encuestas.

Tabla 4.2 Tabla con resultados de opinión de la pregunta N° 1

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE	
Usualmente, ¿cómo se traslada hacia su lugar de destino?	A pie	29	13,18	%
	Transporte público	28	12,73	%
	Taxi	29	13,18	%
	Vehículo propio	57	25,91	%
	Vehículo de un familiar/amigo	30	13,64	%
	Bicicleta	9	4,09	%
	Metro Vía	38	17,27	%
	TOTAL		220	100,00

Nota: En esta tabla se muestra el resultado de la encuesta de la primera pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.26 Porcentajes de la pregunta N° 1



Nota: Este gráfico muestra el porcentaje de la primera pregunta de la encuesta consultada a las personas en internet. (Agosto2013)

De todas las personas encuestadas sobre la pregunta propuesta, encontramos que el 26% tienen vehículo propio, el 17% usan la metro vía, el 13% usan taxi, van a pie o transporte público, el 14% usan un vehículo de un familiar/amigo y un 4% van a pie.

Tabla 4.3 Tabla con resultados de opinión de la pregunta N° 2

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE
Si se moviliza en auto, ¿encuentra parqueadero fácilmente?	SI	28	12,73 %
	No	96	43,64 %
	No aplica	96	43,64 %
	TOTAL	220	100,00

Nota: En esta tabla se muestra el resultado de la encuesta de la segunda pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.27 Porcentajes de la pregunta N° 2



Nota: Este gráfico muestra el porcentaje de la segunda pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

Esta pregunta va relacionada con la pregunta anterior, sabiendo que porcentaje y qué tipo de movilización usan podremos determinar qué porcentaje encuentra parqueo o no.

Dentro de las personas encuestadas el 44% no encuentra parqueo, 13% si encuentra y el otro 43% no aplica según los datos de la pregunta anterior usan un medio alternativo, taxi, autobuses, metrovía entre otros.

Tabla 4.4 Tabla con resultados de opinión de la pregunta N° 3

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE	
¿Cuánto tiempo le toma movilizarse desde su lugar actual hacia otro destino?	30 - 45 MINUTOS	120	54,55	%
	45 - 60 MINUTOS	100	45,45	%
	TOTAL	220	100,00	

Nota: En esta tabla se muestra el resultado de la encuesta de la tercera pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.28 Porcentajes de la pregunta N° 3



Nota: Este gráfico muestra el porcentaje de la tercera pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

En esta pregunta nos muestra, que tiempo les toma a las personas encuestadas llegar a su lugar de destino desde su ubicación actual, podemos ver que el 55% de personas se demoran entre 30 – 45 minutos mientras que el 45% se demoran entre 45 – 60 minutos, estos tiempos dependen del sector y el tiempo en que se trasladan las personas, así mismo depende de la ciudad donde se encuentran las personas (Quito, Guayaquil, entre otras).

Tabla 4.5 Tabla con resultados de opinión de la pregunta N° 4

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE
Usualmente, ¿cómo se traslada hacia su lugar de origen?	A pie	29	13,18 %
	Transporte público	57	25,91 %
	Taxi	20	9,09 %
	Vehículo propio	29	13,18 %
	Vehículo de un familiar/amigo	38	17,27 %
	Bicicleta	9	4,09 %
	Metro Vía	38	17,27 %
	TOTAL	220	100,00

Nota: En esta tabla se muestra el resultado de la encuesta de la cuarta pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.29 Porcentajes de la pregunta N° 4



Nota: Este gráfico muestra el porcentaje de la cuarta pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

Dentro de las personas encuestadas el 26% usan un transporte público para poder trasladarse a su lugar de origen, el 18% usa un vehículo familiar/amigo, 17% usa metrovía, 13% a pie, 9% en taxi y un 4% en bicicleta.

Tabla 4.6 Tabla con resultados de opinión de la pregunta N° 5

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE
¿Cuánto tiempo le toma movilizarse desde su lugar de destino hacia su lugar de origen?	30 - 45 MINUTOS	100	45,45 %
	45 - 60 MINUTOS	120	54,55 %
	TOTAL	220	100,00

Nota: En esta tabla se muestra el resultado de la encuesta de la quinta pregunta consultada a las personas en internet. (Agosto2013)

Figura 4.30 Porcentajes de la pregunta N° 5



Nota: Este gráfico muestra el porcentaje de la quinta pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

En esta pregunta el 55% de las personas encuestadas se demoran entre 30 – 45 minutos en movilizarse entre su lugar de destino hacia su lugar de origen, mientras que el 45% se demora entre 45 – 60 minutos, esto puede significar movilizarse desde su lugar de trabajo hacia su hogar o viceversa.

Tabla 4.7 Tabla con resultados de opinión de la pregunta N° 6

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE	
¿Se encuentra con mucho tráfico al desplazarse hacia o desde su lugar de destino?	SI	170	77,27	%
	NO	50	22,73	%
	TOTAL	220	100,00	

Nota: En esta tabla se muestra el resultado de la encuesta de la sexta pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.31 Porcentajes de la pregunta N° 6



Nota: Este gráfico muestra el porcentaje de la sexta pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

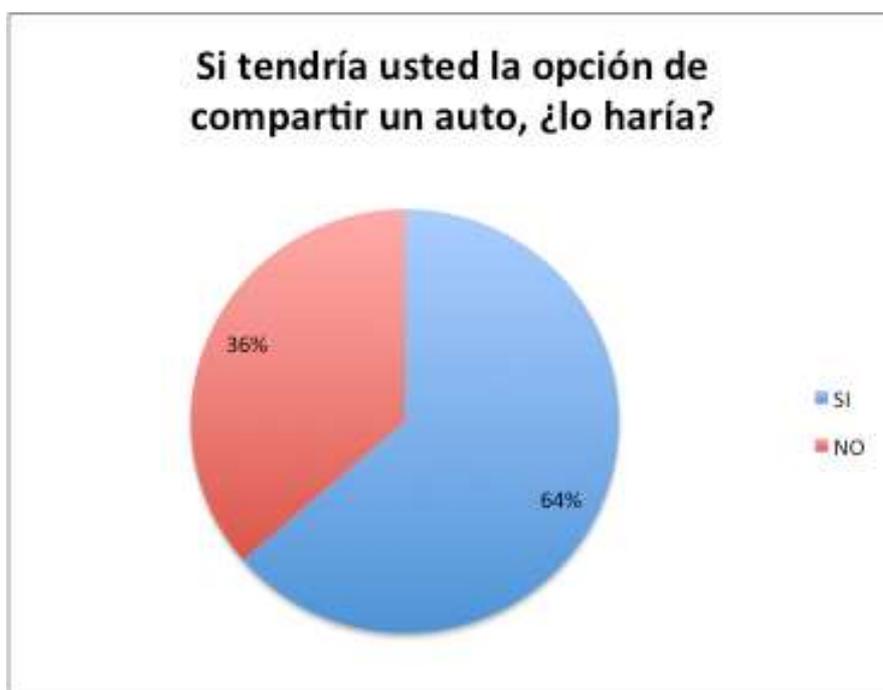
El 77% de las personas encuestadas encuentran mucho tráfico al momento de desplazarse desde o hacia su lugar de destino, mientras que un 23% no tienen inconveniente, esto significa que un gran sector de las ciudades más pobladas como Quito y Guayaquil sufre que un congestionamiento vehicular.

Tabla 4.8 Tabla con resultados de opinión de la pregunta N° 7

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE	
Si tendría usted la opción de compartir un auto, ¿lo haría?	SI	140	63,64	%
	NO	80	36,36	%
	TOTAL	220	100,00	

Nota: En esta tabla se muestra el resultado de la encuesta de la séptima pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.32 Porcentajes de la pregunta N° 7



Nota: Este gráfico muestra el porcentaje de la séptima pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

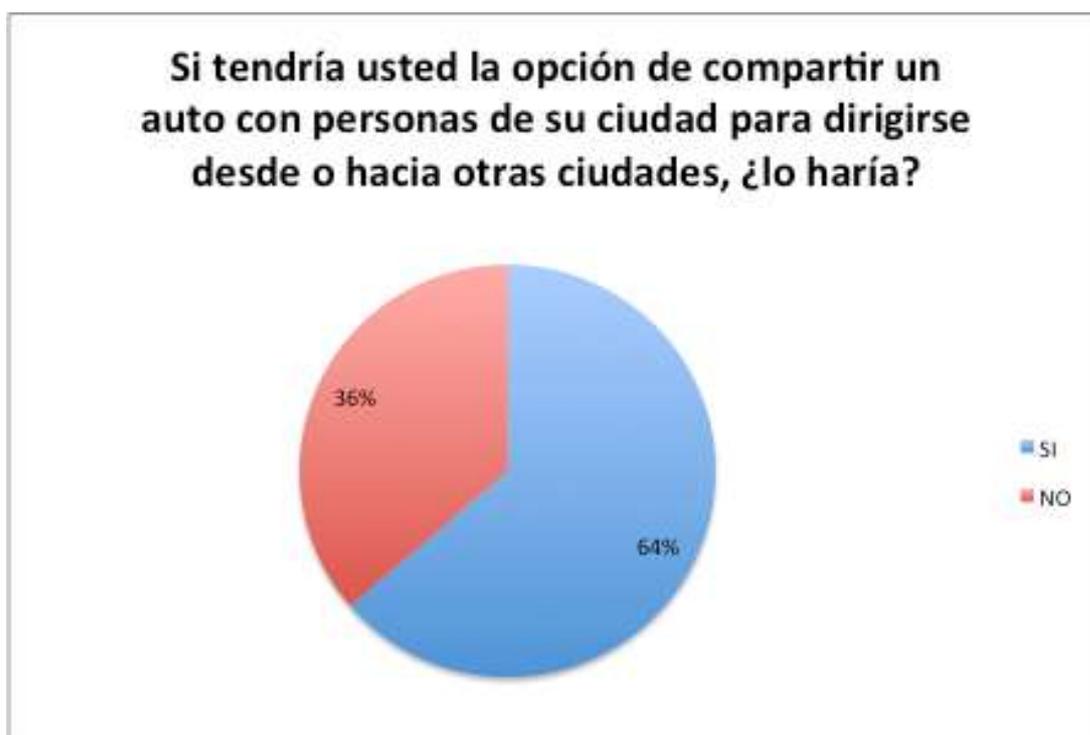
Una vez analizada la situación que se presenta mediante las seis primeras preguntas encontramos que porcentaje de aceptación podría tener nuestra solución a la problemática del tráfico vehicular en las diferentes ciudades, podemos ver que el 64% si compartiría un auto, mientras que un porcentaje del 36% no lo haría.

Tabla 4.9 Tabla con resultados de opinión de la pregunta N° 8

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE
Si tendría usted la opción de compartir un auto con personas de su ciudad para dirigirse desde o hacia otras ciudades, ¿lo haría?	SI	140	63,64 %
	NO	80	36,36 %
	TOTAL	220	100,00

Nota: En esta tabla se muestra el resultado de la encuesta de la octava pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.33 Porcentajes de la pregunta N° 8



Nota: Este gráfico muestra el porcentaje de la octava pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

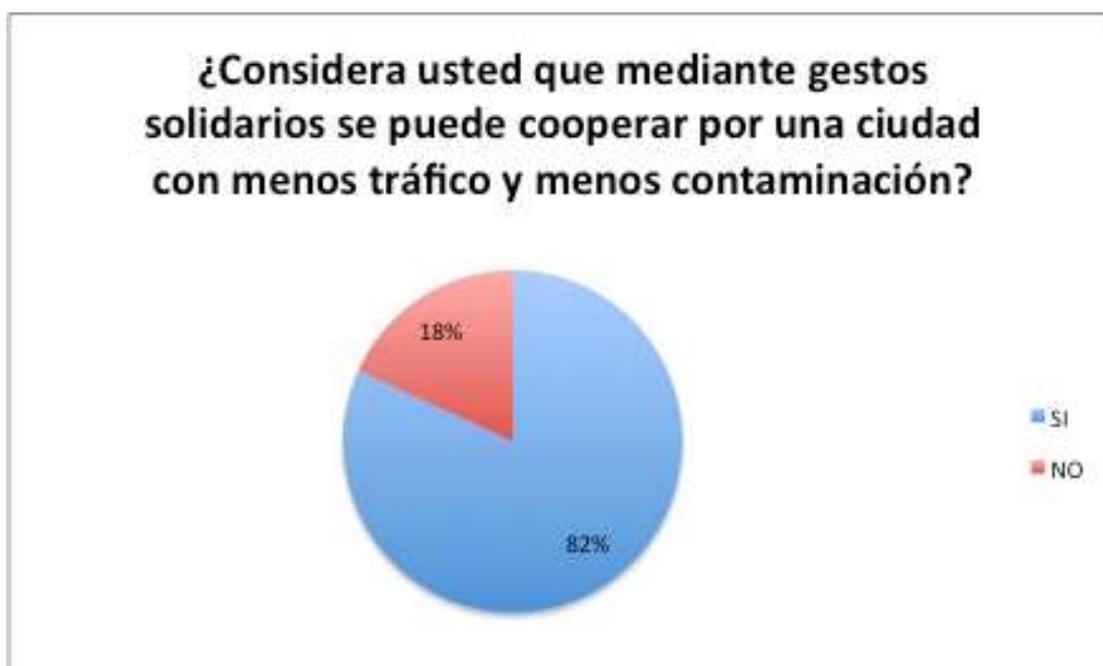
De acuerdo a nuestra encuesta planteamos una pregunta para saber que porcentaje de acogida tendría nuestra solución planteamos y pudimos observar que un 64% de las personas encuestadas si compartirías un auto, mientras que un 36% no lo harían.

Tabla 4.10 Tabla con resultados de opinión de la pregunta N° 9

INDICADORES	PARAMETROS	CANTIDAD	PORCENTAJE
¿Considera usted que mediante gestos solidarios se puede cooperar por una ciudad con menos tráfico y menos contaminación?	SI	180	81,82 %
	NO	40	18,18 %
	TOTAL	220	100,00

Nota: En esta tabla se muestra el resultado de la encuesta de la novena pregunta consultada a las personas en internet. (Agosto 2013)

Figura 4.34 Porcentajes de la pregunta N° 9



Nota: Este gráfico muestra el porcentaje de la novena pregunta de la encuesta consultada a las personas en internet. (Agosto 2013)

Podemos observar que en nuestro medio la cooperación es bien recibida el 82% de las personas encuestadas creen que podemos mejorar la ciudad mediante gestos solidarios, mientras que un porcentaje bajo, solamente 18% creen que no.

4.2 Arquitectura del Sistema

La aplicación móvil es un sistema cliente – servidor y se basa en un modelo de arquitectura de tres capas. En la capa de la base de datos se alojarán todos los datos que corresponden a los usuarios que se registrarán en la aplicación para así obtener con mayor facilidad los cálculos que la aplicación necesita para ofrecer al usuario los datos más exactos.

La capa de negocio es el vínculo entre la capa de base de datos y la capa de presentación en la aplicación la cual está formada por tres servicios web en donde se encuentran los diferentes métodos para poder ser invocados. Estos métodos se comunicarán directamente con la base de datos donde se encuentran los procedimientos

almacenados que realizan las diferentes operaciones para la recepción y envío de información.

En el entorno de desarrollo la interfaz creada en la capa de presentación será usada por el usuario final en su dispositivo móvil el cual tendrá el sistema operativo Android, a través de esta interfaz se permitirá enviar peticiones y recibir resultados utilizando la capa de negocio.

4.3 Descripción general de la Arquitectura

4.3.1 Diagrama de bloques

Figura 4.35 Diagrama de bloques de la aplicación



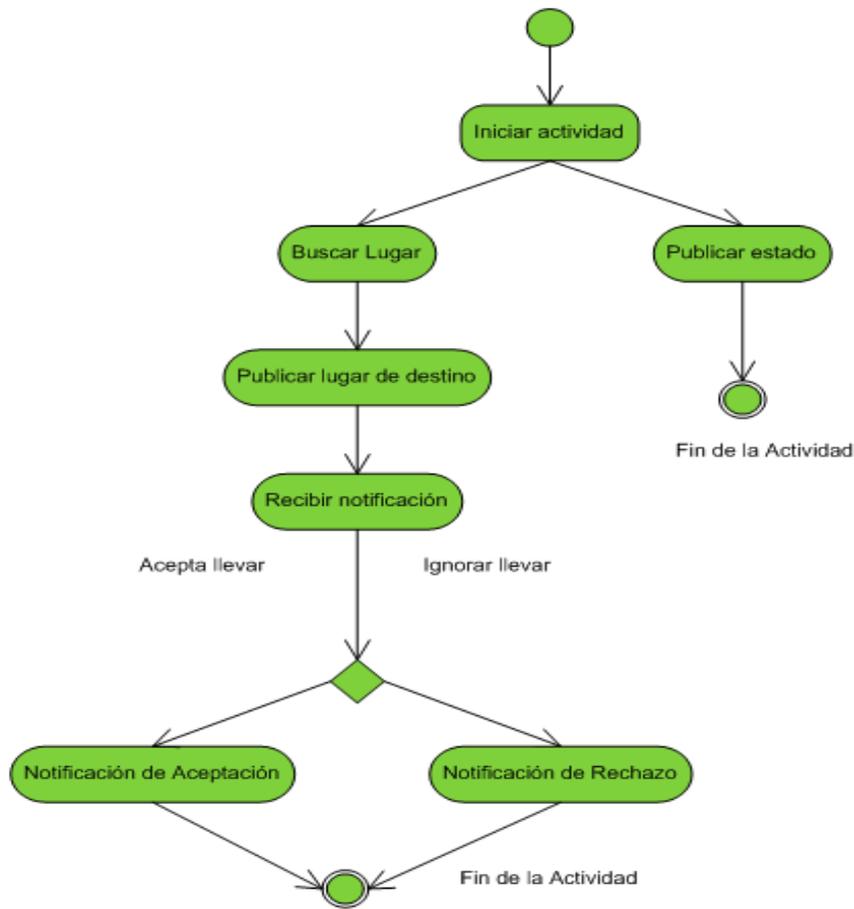
Nota: Muestra el diagrama de bloques de la aplicación cuando está registrado en la aplicación y las operaciones que puede realizar. Elaborado por los Autores. (Agosto 2013)

Este tipo de diagrama de bloques nos da una mejor visión del funcionamiento interno que posee la aplicación, desde el registro del usuario, y las diversas interacciones principales que podrá realizar usando la aplicación y el resultado final que en nuestro caso es la selección de la mejor ruta junto al mejor tiempo que esta pueda poseer, utilizando así los recursos que un teléfono móvil posee.

Como podemos apreciar en la figura 4.35, el proceso inicia desde que usuario se registra en la aplicación, el cual podrá realizar ciertas funcionalidades donde podrán agregar amigos, enviar solicitudes a tus amigos para que te ayuden a llegar al lugar que te quieres dirigir, obteniendo la mejor ruta considerando el tráfico vehicular que se reflejará en el visor del mapa dentro de la aplicación.

4.3.2 Diagrama de Actividades

Figura 4.36 Diagrama de Actividades para postear llevar



Nota: Muestra el diagrama de actividad de la aplicación cuando el usuario escribe una publicación al dirigirse a un lugar. Elaborado por los Autores. (Septiembre 2013)

4.3.3 Fórmula de Haversine

Para encontrar los usuarios que se encuentran a cierta distancia de un radio de latitud/longitud dada usamos una consulta SELECT a la base de datos usando la Fórmula de Haversine explicada con anteriormente.

Aquí está la sentencia de SQL usada para encontrar todos los usuarios cercanos que se encuentran dentro de un radio de 5km dependiendo de la latitud y longitud donde se encuentra el usuario que desea ser llevada. Se calcula la distancia sobre la base de la latitud/longitud de la fila y la latitud/longitud de destino, luego pide sólo las filas donde el valor de la distancia es menos de 5.

Figura 4.37 Fórmula de Haversine en MySql

```
SELECT id, (6371 * acos (cos (radians (37)) * cos (radians (lat ) ) * cos( radians( lng ) - radians(-122) ) + sin( radians(37) ) * sin( radians( lat ) ) ) ) AS distance FROM markers HAVING distance < 5 ORDER BY distance;
```

Nota: Muestra la fórmula de Haversine aplicada en motor de base de datos Mysql.
Elaborado por los Autores. (Agosto 2013)

4.3.4 Trazos de rutas en Google Maps

Para poder dibujar los trazos de las rutas en el mapa de Google que tiene nuestra aplicación móvil es necesario realizar un proceso el cual comprende de los siguientes pasos:

1. Armar la dirección de la consulta que utilizaremos para obtener los datos necesarios para la aplicación, para esto usamos el formato json para mostrar los datos, los parámetros que se utilizan son los siguientes: a) origen (latitud/longitud), b) destino (latitud/longitud).

Figura 4.38 Solicitud del API de rutas

```
https://maps.googleapis.com/maps/api/directions/json/parametros
```

Nota: Dirección url para realizar la solicitud de Api de rutas de Google.
Obtenido desde <https://developers.google.com/maps/documentation/directions/?hl=es>. (Agosto 2013)

2. Los datos que recibimos de la consulta esta en forma de nodo debido al formato utilizado, este formato nos ayuda a poder guardar en un listado los datos que contiene la consulta para posteriormente utilizarlos en nuestro mapa.

A continuación el código que utilizamos para lo explicado anteriormente.

Figura 4.39 Fragmento de código para la obtención de coordenadas

```
for(int i=0;i<jRoutes.length();i++){
jLegs = (JSONObject)jRoutes.get(i).getJSONArray("legs");
List path = new ArrayList<HashMap<String, String>>();
for(int j=0;j<jLegs.length();j++){
jSteps = (JSONObject)jLegs.get(j).getJSONArray("steps");
for(int k=0;k<jSteps.length();k++){
String polyline = "";
polyline
(String)((JSONObject)((JSONObject)jSteps.get(k)).get("polyline")).get("points");
List<LatLng> list = decodePoly(polyline);
for(int l=0;l<list.size();l++){
HashMap<String, String> hm = new HashMap<String, String>();
hm.put("lat", Double.toString(((LatLng)list.get(l)).latitude) );
hm.put("lng", Double.toString(((LatLng)list.get(l)).longitude) );
path.add(hm);
}
}
routes.add(path);
}
}
```

Nota: Muestra el código Android para poder obtener las coordenadas para el trazo de rutas. Elaborado por los Autores. (Agosto 2013)

Luego de que los datos los tenemos en un listado procedemos a dibujar en el mapa, el siguiente código realiza lo antes mencionado.

Figura 4.40 Fragmento de código para dibujar el trazo de la ruta

```
for(int i=0;i<result.size();i++){
points = new ArrayList<LatLng>();
lineOptions = new PolylineOptions();

List<HashMap<String, String>> path = result.get(i);
for(int j=0;j<path.size();j++){
HashMap<String,String> point = path.get(j);
double lat = Double.parseDouble(point.get("lat"));
double lng = Double.parseDouble(point.get("lng"));
LatLng position = new LatLng(lat, lng);
points.add(position);
}
lineOptions.addAll(points);
lineOptions.width(5);
lineOptions.color(Color.BLUE);
}
mMap.addPolyline(lineOptions);
```

Nota: Muestra el código Android para dibujar el trazo de rutas. Elaborado por los Autores. (Agosto 2013)

4.3.5 Búsqueda de lugares

El Google Places API es un servicio web que devuelve información sobre un conjunto de lugares en base a una cadena – por ejemplo “Mall del Sol”, “Aeropuerto Simón Bolívar”. El servicio responde con una lista de lugares que coinciden con la cadena de texto.

Los datos que nos retorna esta consulta nos ayudará a mostrarle al usuario las ubicaciones de una mejor manera y el usuario podrá seleccionar solamente lugares que existan además de tener un mejor acceso a los datos para poder hacer los cálculos necesarios para nuestra aplicación.

La petición de búsqueda es una URL HTTP de la siguiente forma:

Figura 4.41 Solicitud de búsqueda de lugares API Google

```
https://maps.googleapis.com/maps/api/place/textsearch/json?parametros
```

Nota: Dirección url para realizar búsqueda de lugares a través de Api de Google. Obtenido desde <https://developers.google.com/places/documentation/search?hl=es>. (Agosto 2013)

Parámetros necesarios

- **Consulta.-** La cadena de texto en el que va a buscar, por ejemplo: “restaurante”. El servicio devolverá los resultados de acuerdo a la relevancia percibida

- **Clave.-** Esta clave identifica a la aplicación, la cual provee google al ingresar en la consola de servicios.

- **Sensor.-** Indica si la solicitud del lugar vino de un dispositivo que utiliza un sensor de localización como es el caso de los dispositivos móviles.

Una vez que retornamos los datos debemos agregarlos a una lista para posteriormente agregar los datos a un ListView, a continuación el código que hace lo antes mencionado.

Figura 4.42 Fragmento de código para obtener lugar seleccionado

```

jObject = new JSONObject(jsonData[0]);
places = placeJsonParser.parse(jObject);
for(int i=0;i<places.size();i++){
    HashMap<String, String> hmPlace = places.get(i);
    HashMap<String, String> map = new HashMap<String, String>();

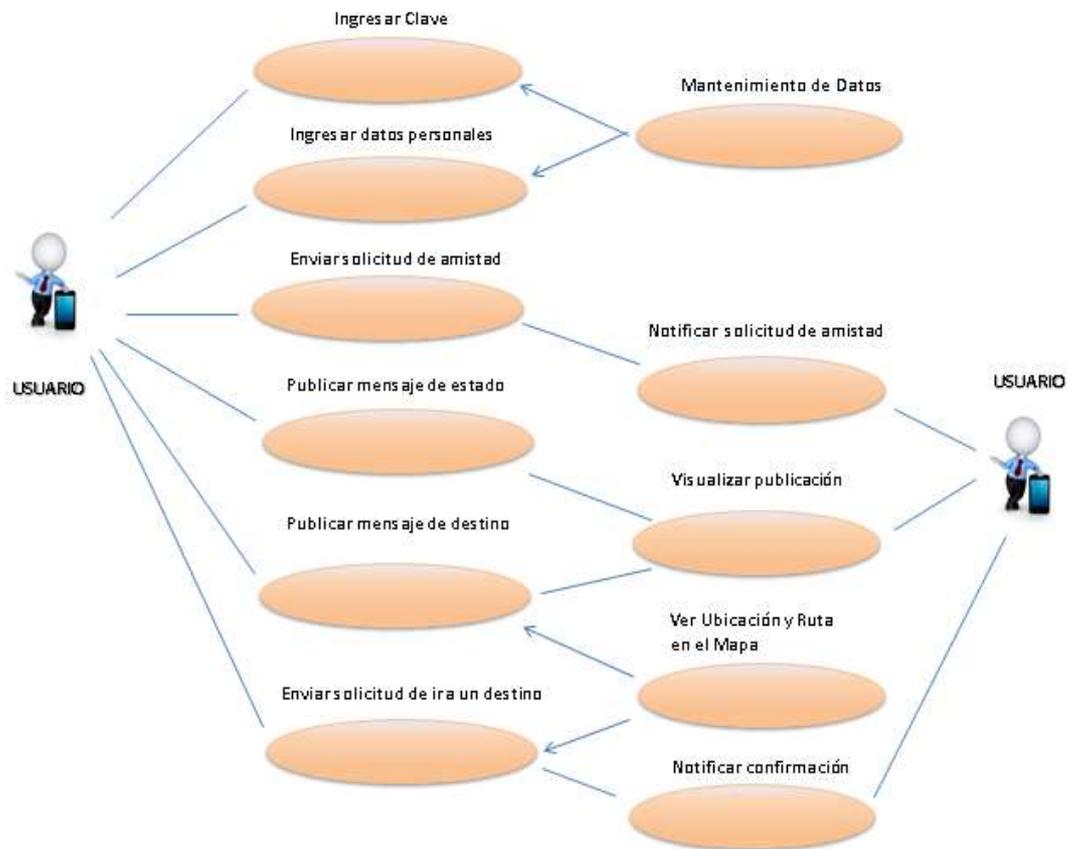
    map.put(Variables.KEY_NAMEPLACE,hmPlace.get("place_name"));
    map.put(Variables.KEY_LAT,hmPlace.get("lat"));
    map.put(Variables.KEY_LONG,hmPlace.get("lng"));
    ListDatos.add(map);
}

```

Nota: Muestra el código Android para obtener el nombre de lugar, latitud, y longitud del lugar resultado de la búsqueda. Elaborado por los Autores. (Agosto 2013)

4.3.6 Diagrama de Casos de Uso

Figura 4.43 Diagrama de casos de uso



Nota: Se muestra los pasos que el actor usuario con los casos de uso para interactuar con otro usuario dentro de la aplicación móvil. Elaborado por los Autores. (Agosto 2013)

Se ha determinado solo un actor:

El usuario.- El cual puede publicar sitios a donde se dirija, aceptar, llevar o ignorar a un amigo, enviar o rechazar solicitudes de amistad, buscar amigos en la aplicación, visualizar rutas y publicar mensajes de estado.

Se han definido los siguientes casos de uso:

- **Registrar Usuario.-** Para registrarse en la aplicación, se debe ingresar su nombre, apellidos, correo, contraseña, foto, fecha de nacimiento, sexo, teléfono.
- **Login Usuario.-** Podrás ingresar correo y la contraseña para poder acceder a la aplicación. O podrás realizarlo ingresar tu correo y contraseña de la red social Facebook donde tendrás aceptar los permisos de la información que se desea compartir.
- **Publicar mensaje.-** Podrás publicar mensajes de estados para que tus amigos puedan verlos.
- **Publicar Visitar.-** Podrás publicar un lugar de visita con un mensaje para que tus amigos se enteren dónde te dirijas y puedan mandar una solicitud para que lo lleves.
- **Lugares.-** Podrás ver los lugares que has visitado y ver tu ubicación a través del GPS y donde te estás dirigiendo con sus respectivos trazos de rutas para escoger la mejor ruta incluso los amigos que vas a llevar en ese momento.

Gracias a la ubicación que está constantemente actualizándose en la aplicación también podrás ver la ubicación de todos tus amigos y en donde se encuentran en ese momento y así les llegará una notificación que deseas ser llevado.

Mis amigos.- Aquí encontraras los amigos que has agregado a la aplicación

Configuración.- En esta opción podrás realizar las siguientes operaciones:

- **Buscar Amigos:** Podrás ver todos los usuarios registrados en la aplicación y mandarles una solicitud de amistad.
- **Configurar su cuenta:** Podrás cambiar tus datos personales como son tus nombres, apellidos, número de teléfono, dirección de email, agregar o modificar una biografía, sexo. También podrás cambiar tu contraseña ingresada anteriormente.
- **Cerrar sesión:** Cierre de sesión de la aplicación.

Notificaciones.- Podrás consultar cuando han aceptado tu solicitud de amistad, también podrás visualizar si un amigo te ha enviado solicitud para que lo llesves a lugar donde se dirige.

4.3.7 Diagramas de Eventos

Para entender un diagrama de eventos debemos primero definir el concepto de un evento. Un evento es un suceso que se realiza en algún lugar en el tiempo y este influye en el comportamiento del sistema.

Un evento se puede dar al interactuar con la interfaz del dispositivo móvil, al seleccionar una opción de una lista, o al hacer clic en un botón.

Todos los eventos van numerados en orden de acuerdo a la secuencia lógica que se vayan dando en la aplicación.

Tabla 4.11 Diagrama de Eventos

CONTROL DE PROYECTOS	FORMATO DE EVENTOS
Nombre del Caso de Uso:	Ingresar, Modificar, Publicar, Notificar, Consultar registros
Actor Responsable:	Usuario
EVENTO	RESPUESTA DEL SISTEMA
1.Si selecciono el botón "Login with Facebook"	2.Se abre la pantalla del Facebook

3. Ingreso los datos y pulso el botón

“Iniciar sesión “

4. Se solicita permiso para poder acceder a la información privada del Facebook

5. Si pulsa “Si” o “No”

6. Si pulso “Si” se incluye sus amigos de Facebook en caso de que estén registrados en la aplicación, caso contrario no.

Se abre la pantalla del HOME con las publicaciones de mis amigos y las más.

7. Si deslizo la lista hacia abajo

8. Actualiza las publicaciones

9. Si pulso en el botón con el logo

10. Se abre una pantalla con la consulta de notificaciones

11. Si pulso en el botón “<”

12. Regreso a la pantalla del HOME

13. Si selecciona la opción de “Mapas”

14. Se abre una pantalla con la ubicación actual, también podrá visualizar el lugar donde se dirija, los Amigos que llevas, con su ruta y su tiempo estimado

15. Si selecciona la opción de “Mis Amigos”

16. Se abre una pantalla con todos los amigos agregados

17. Si selecciona la opción de “Configuración”

18. Se abre una pantalla con opciones importantes.

- Buscar Amigos
- Configuración de la cuenta
- Cerrar sesión

19. Si pulsa en el botón de” Buscar Amigos”

	<p>20. Se abre un menú con las siguientes opciones</p> <ul style="list-style-type: none"> ▪ Solicitudes de Amigos ▪ Buscar amigos en Facebook ▪ Buscar amigos en la aplicación
<p>21. Si pulso en el botón de “Solicitudes de Amigos”</p>	<p>21. Se abre una pantalla con las solicitudes de amistad</p>
<p>22. Si pulso en el botón de “Buscar amigos en Facebook”</p>	<p>23. Se abre una pantalla con los amigos de Facebook para enviar las solicitudes.</p>
<p>24. Si pulso en el botón de “Buscar amigos en la aplicación”</p>	<p>25. Se abre una pantalla con los usuarios registrados en la aplicación</p>
<p>26. Si pulso en el botón de “Configuración de la Cuenta”</p>	<p>27. Se abre una menú con las siguientes opciones:</p> <ul style="list-style-type: none"> ▪ Tu perfil ▪ Cambiar Contraseña ▪ Configuración de privacidad
<p>28. Si pulso en el botón de “Tu perfil”</p>	<p>29. Se abre una pantalla con sus datos personales para que puedan ser modificados.</p>
<p>30. Si pulso el botón “Guardar”</p>	<p>31. Graba sus datos personales</p>
<p>32. Si pulso en el botón de “Cambiar Contraseña”</p>	<p>33. Se abre una pantalla para que modifique su contraseña.</p>
<p>34. Si pulso el botón “Guardar”</p>	

36. Si pulso el botón “Cerrar sesión”

35. Graba su clave nueva

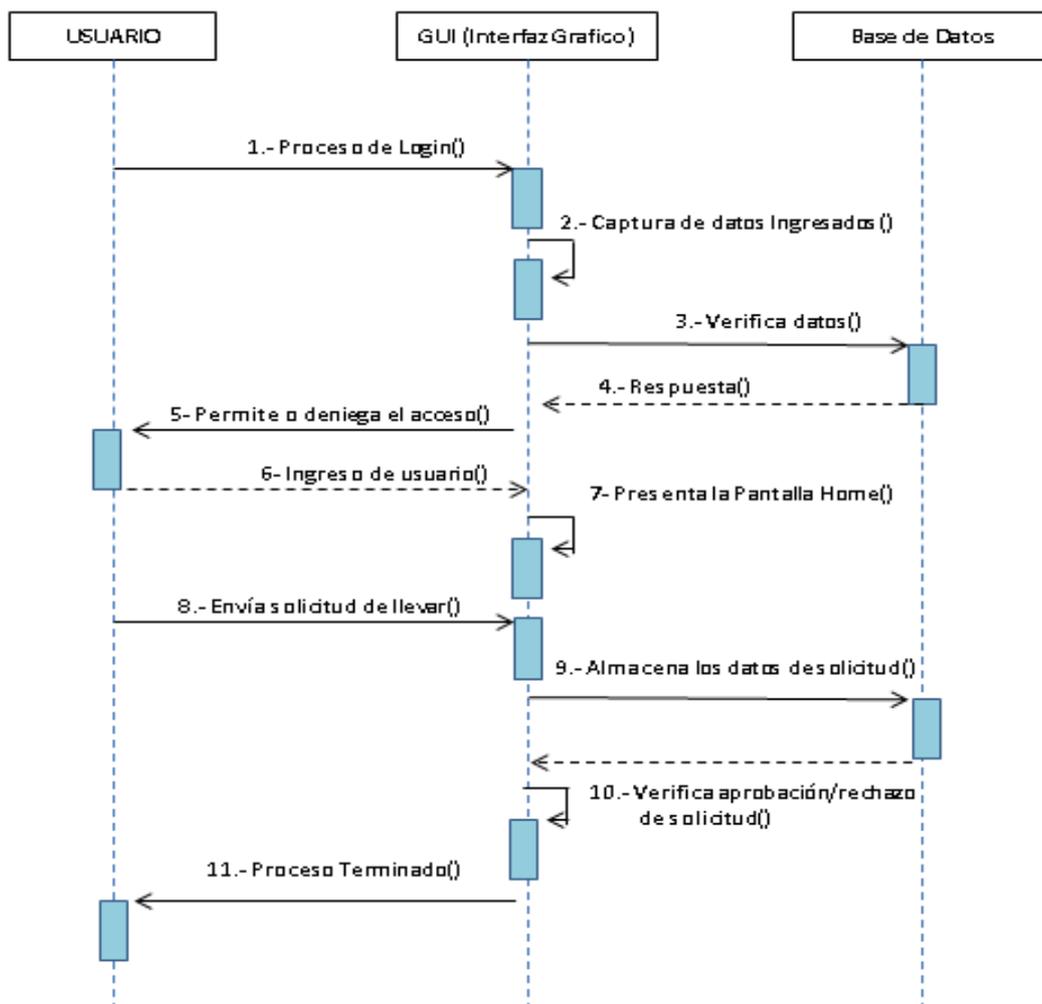
37. Cierra la sesión y abre pantalla de inicio de la aplicación

Nota: Muestra todos los eventos que se pueden realizar desde que inicia sesión con la aplicación móvil hasta de finaliza su sesión. Elaborado por los Autores. (Agosto 2013)

4.3.8 Diagrama de Interacción

En el primer diagrama de interacción observamos el comportamiento del sistema a partir de los eventos generados por el actor (El usuario) al momento de realizar una petición de llevar.

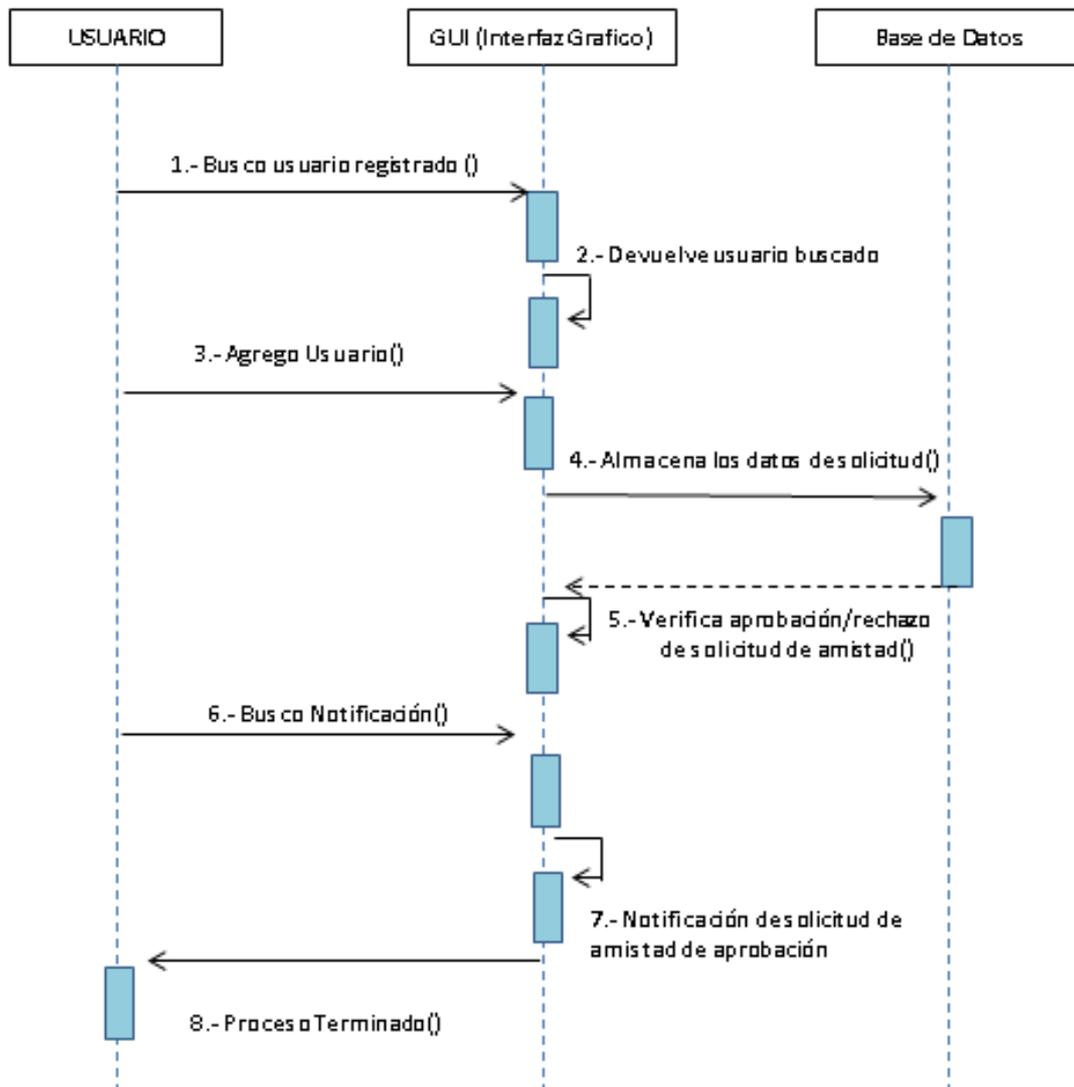
Figura 4.44 Diagrama de interacción para postear llevar



Nota: Muestra el diagrama de interacción cuando el usuario quieren que lo lleven al destino de su amigo Elaborado por los Autores. (Septiembre 2013).

En este segundo gráfico apreciamos el comportamiento del sistema al momento de enviar una solicitud de amistad por parte del usuario.

Figura 4.45 Diagrama de Interacción enviar solicitud de amistad



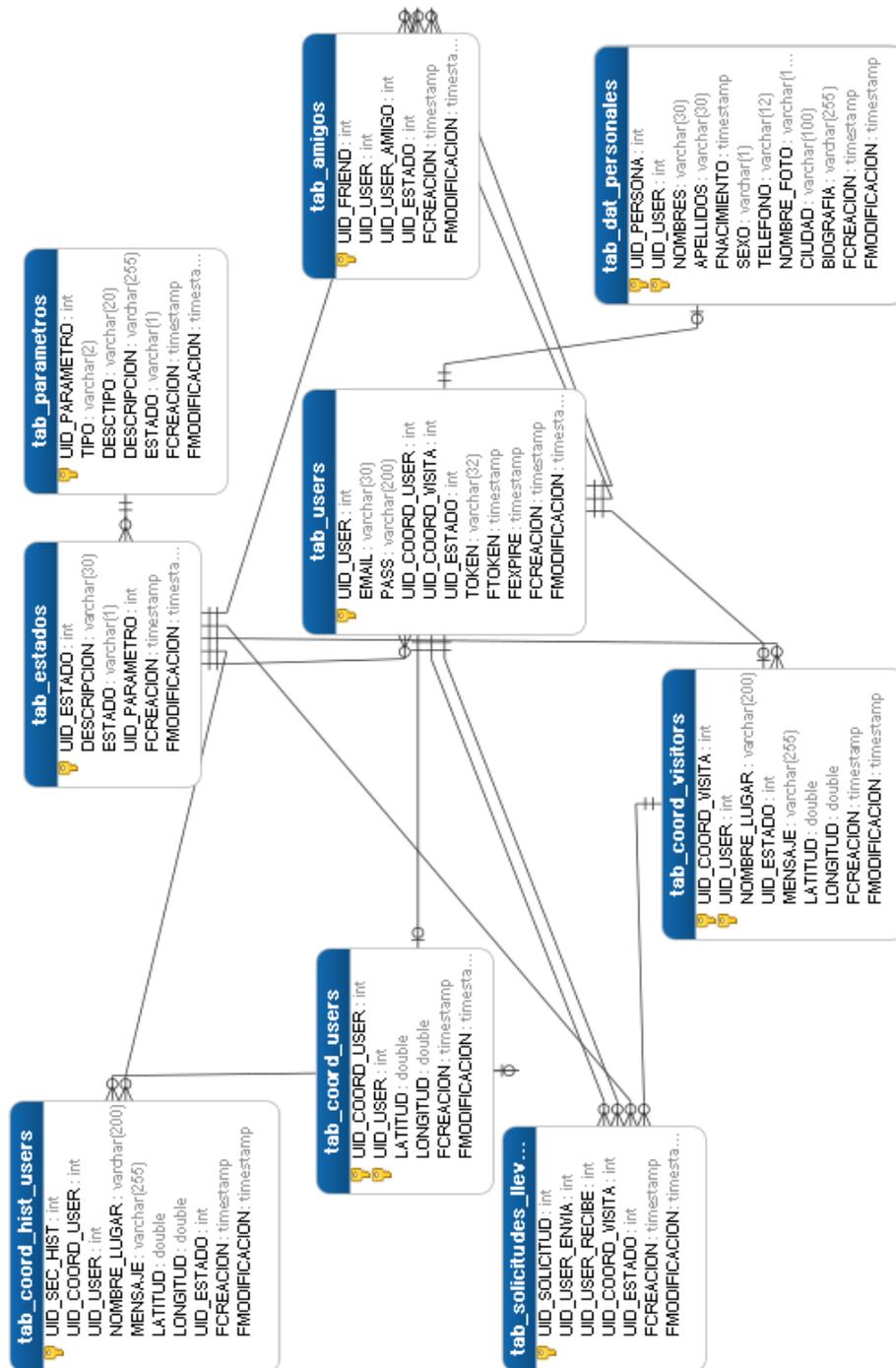
Nota: Muestra el diagrama de interacción cuando el usuario quiere agregar un amigo enviando una solicitud de amistad. Elaborado por los Autores. (Agosto 2013)

En este diagrama podemos apreciar cual sería la interacción del usuario para poder enviar una solicitud de amistad a otro usuario para así poder tenerlo dentro de su red de amigos, lo cual facilitará tener una mejor ayuda al momento de buscar usuarios cercanos a mi ubicación, esta es una de las seguridades que la aplicación proporciona pero es por parte del usuario el cual aceptará solamente usuarios conocido.

4.4 Diseño de la arquitectura

4.4.1 Capa de Base de Datos

Figura 4.46 Modelo Entidad-Relación



Nota: Muestra el modelo entidad – relación de la base de datos, la cual interactúa con la aplicación Llévame y se guardan los datos. Elaborado por los Autores. (Agosto 2013)

4.4.2 Definición de Tablas del Sistema

A continuación se detallan las tablas de la aplicación Llévame:

- **Tabla de Usuarios**

Tabla 4.12 Diccionario de Datos de la tabla usuarios

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_USER	integer	Primary Key. Auto-Increment Not null	Id usuario registrado
EMAIL	varchar	Not null	Correo del usuario registrado
PASS	varchar	Not null	Contraseña del usuario registrado
UID_COORD_USER	integer	Foreign Key	Id coordenada del usuario
UID_COORD_VISITA	integer	Foreign Key	Id coordenada visita del usuario
UID_ESTADO	integer	Foreign Key Not null	Id estado
TOKEN	varchar	Not null	Token de registro
FTOKEN	timestamp		Fecha del token
FEXPIRE	timestamp		Fecha de caducidad del token
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guardan los datos del usuario.
Elaborado por los Autores. (Septiembre 2013)

- **Tabla de Datos Personales**

Tabla 4.13 Diccionario de Datos de la tabla datos personales

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_PERSONA	integer	Primary Key Auto-Increment Not null	Id de la persona
UID_USER	integer	Primary Key Foreign Key Not null	Id usuario registrado

NOMBRES	varchar	Nombres del usuario
APELLIDOS	varchar	Apellidos del usuario
FNACIMIENTO	timestamp	Fecha de Nacimiento
SEXO	varchar	Sexo
TELEFONO	varchar	Teléfono
NOMBRE_FOTO	varchar	Nombre de Foto
CIUDAD	varchar	Ciudad de registro
BIOGRAFIA	varchar	Biografía del usuario
FCREACION	timestamp	Fecha de creación
FMODIFICACION	timestamp	Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guardan los datos personales del usuario. Elaborado por los Autores. (Septiembre 2013)

▪ **Tabla de Coordenadas de Usuarios**

Tabla 4.14 Diccionario de Datos de la tabla coordenadas de usuarios

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_COORD_USER	integer	Primary Key Auto-Increment Not null	Id coordenada del usuario
UID_USER	integer	Primary Key Foreign Key Not null	Id usuario registrado
LATITUD	double	Not null	Latitud de localización del usuario
LONGITUD	double	Not null	Longitud de localización del usuario
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guarda la coordenada actual de usuario. Elaborado por los Autores. (Septiembre 2013)

▪ **Tabla de Coordenadas de Visitas**

Tabla 4.15 Diccionario de Datos de la tabla coordenadas de visitas

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_COORD_VISITA	integer	Primary Key Auto-Increment Not null	Id coordenada de visita del usuario
UID_USER	integer	Primary Key Foreign Key Not null	Id usuario registrado
NOMBRE_LUGAR	varchar		Nombre del lugar a donde desea visitar
UID_ESTADO	integer	Foreign Key Not null	Id estado
MENSAJE	varchar		Mensaje de publicación
LATITUD	double	Not null	Latitud de localización del usuario
LONGITUD	double	Not null	Longitud de localización del usuario
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guarda las coordenadas de visitas cuando se desea ir a un lugar. Elaborado por los Autores. (Septiembre 2013)

▪ **Tabla de Historial de coordenadas de usuario**

Tabla 4.16 Diccionario de Datos de la tabla historial de mensajes de estado

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_SEC_HIST	integer	Primary Key Auto-Increment Not null	Id registro
UID_COORD_USER	integer	Foreign Key	Id coordenada del usuario
UID_USER	integer		Id usuario registrado
NOMBRE_LUGAR	varchar		Nombre del lugar de donde se publica

MENSAJE	varchar		Mensaje de publicación
LATITUD	double		Latitud de localización del usuario
LONGITUD	double		Longitud de localización del usuario
UID_ESTADO	integer	Foreign Key Not null	Id estado
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guarda los mensajes publicados por el usuario. Elaborado por los Autores. (Septiembre 2013)

▪ Tabla de Amigos

Tabla 4.17 Diccionario de Datos de la tabla amigos

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_FRIEND	integer	Primary Key Auto-Increment Not null	Id registro de amigo
UID_USER	integer	Foreign Key Not null	Id usuario registrado que envía la solicitud de amistad
UID_USER_AMIGO	integer	Foreign Key Not null	Id usuario registrado que recibe la solicitud de amistad
UID_ESTADO	integer	Foreign Key Not null	Id estado
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guarda los amigos agregados por el usuario. Elaborado por los Autores. (Septiembre 2013)

▪ **Tabla de Solicitudes de Llévame**

Tabla 4.18 Diccionario de Datos de la tabla solicitudes de llévame

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_SOLICITUD	integer	Primary Key Auto-Increment Not null	Id registro de la solicitud
UID_USER_ENVIA	integer	Foreign Key Not null	Id usuario registrado que envía la solicitud del llévame
UID_USER_RECIBE	integer	Foreign Key Not null	Id usuario registrado que recibe la solicitud del llévame
UID_COORD_VISITA	integer	Foreign Key Not null	Id coordenada de visita del usuario
UID_ESTADO	integer	Foreign Key Not null	Id estado
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guarda las solicitudes. Elaborado por los Autores. (Septiembre 2013)

▪ **Tabla de Estados**

Tabla 4.19 Diccionario de Datos de la tabla estados

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_ESTADO	integer	Primary Key Auto-Increment Not null	Id estado
DESCRIPCION	varchar		Descripción del Estado
ESTADO	varchar		Activo "A" o Inactivo "I"
UID_PARAMETRO	integer	Foreign Key	Id parámetro
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guarda el estado del usuario. Elaborado por los Autores. (Septiembre 2013)

▪ **Tabla de Parámetros**

Tabla 4.20 Diccionario de Datos de la tabla parámetros

COLUMNA	TIPO DE DATO	DETALLE	DESCRIPCIÓN
UID_PARAMETRO	integer	Primary Key Auto-Increment Not null	Id registro de parámetros
TIPO	varchar		Código Tipo de parámetro
DESCTIPO	varchar	Not null	Descripción del tipo de parámetro
DESCRIPCION	varchar		Detalle del tipo de parámetro
ESTADO	varchar	Not null	Estado Activo “A” o Inactivo “I”
FCREACION	timestamp		Fecha de creación
FMODIFICACION	timestamp		Fecha de modificación

Nota: Muestra los campos y la longitud de la tabla donde se guardan los tipos de parámetros o tipos de mensajes en la aplicación. Elaborado por los Autores. (Septiembre 2013)

4.4.3 Definición de Procedimientos Almacenados

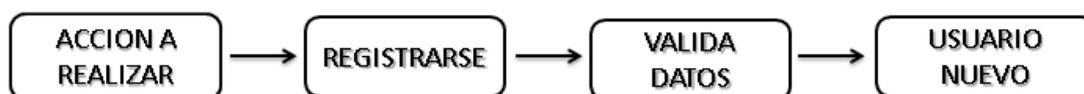
La aplicación propuesta, contiene algunos procedimientos para realizar las operaciones que tiene el usuario, como: ingreso, búsqueda, consulta, envío de información.

Registro del usuario

El registro es el primer paso que el Usuario realizará al momento de abrir la aplicación indiferentemente si se conecta por ‘Facebook’.

Cuando el usuario se registra, todos los datos son encriptados para luego ser guardados en la base de datos para poder así realizar todas las funciones que esta propone para facilidad misma del usuario.

Figura 4.47 Esquema del procedimiento registrar usuario



Nota: Esquema cuando un usuario se registra por primera vez a la aplicación móvil.
Elaborado por los Autores. (Agosto 2013)

- **Actualización de coordenada**

En la aplicación cada cierto tiempo actualizara la ubicación actual automáticamente de la persona donde se encuentre en ese momento, la misma que será usada para el trazado de ruta entre dos usuarios.

Figura 4.48 Esquema del procedimiento actualiza ubicación



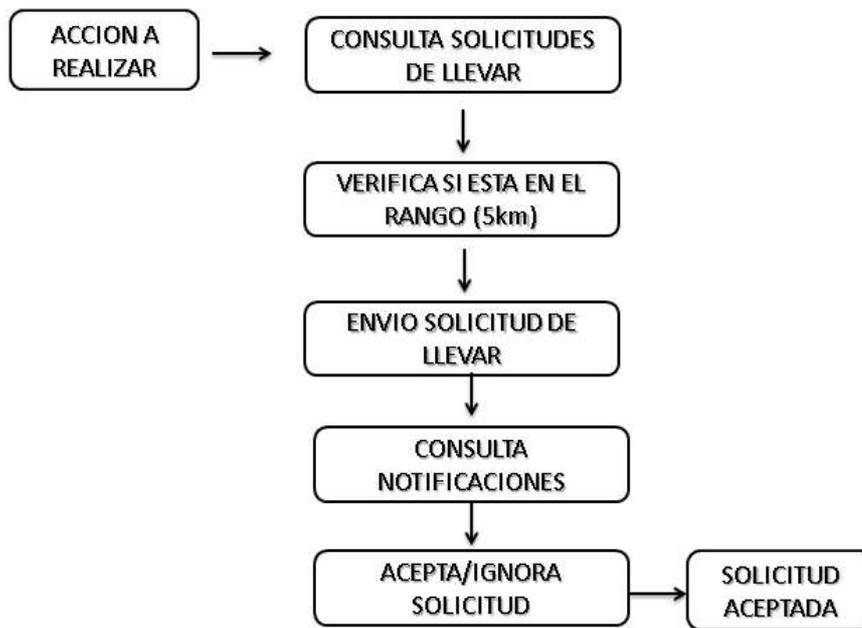
Nota: Esquema cuando un usuario actualiza su ubicación actual desde la aplicación móvil.
Elaborado por los Autores. (Agosto 2013)

- **Consulta y Envió de notificación de solicitudes Llévame**

Para que un usuario pueda enviar una solicitud de “Llévame” debe visualizar en la pantalla principal la publicación de visita del otro usuario para poder enviarle la solicitud, siempre y cuando este dentro del rango de cobertura de los 5km con el usuario a quien se le desea enviar la solicitud. Si el usuario acepta o rechaza la solicitud le aparecerá en su bandeja de notificaciones.

Las notificaciones que el usuario puede ver estarán presentes hasta que las deniegue o acepte en cualquier momento, debido a que en la base los datos simplemente cambian de estado estas notificaciones cambiarán de acuerdo a lo que el usuario decida hacer.

Figura 4.49 Esquema del procedimiento envía solicitud llévame



Nota: Esquema cuando un usuario envía una solicitud para que lo lleve al lugar de destino otro usuario desde la aplicación móvil. Elaborado por los Autores. (Agosto 2013)

4.5. Capa de Reglas de Negocio

En la capa de negocio se define las reglas que deben cumplirse, cuando el usuario envía o recibe peticiones.

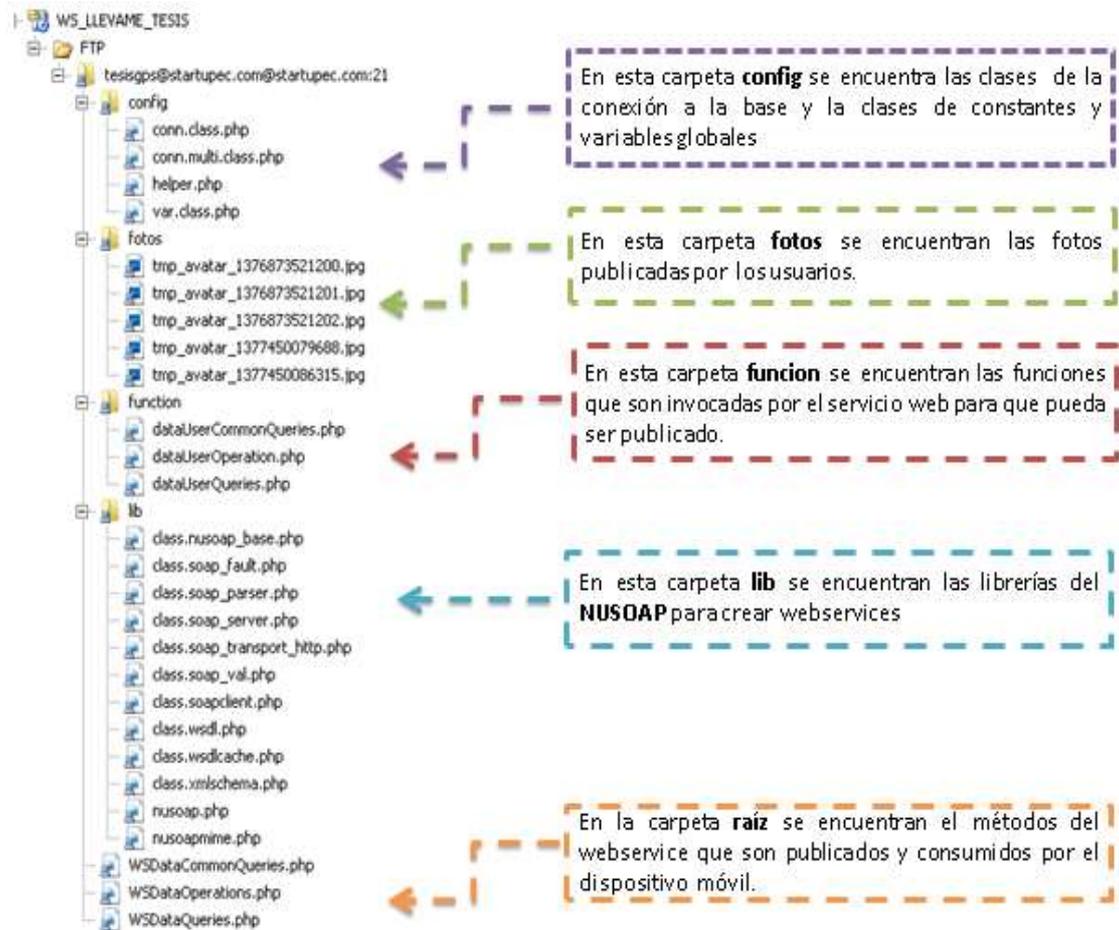
La capa de negocio se comunica con la capa de presentación de manera interactiva encargándose de recibir solicitudes y presentar resultados desde la capa de datos, donde este almacena o recupera información según la petición del usuario.

4.6 Especificación de Librerías

Para nuestra aplicación móvil la capa de negocio se desarrolló bajo el lenguaje de programaciones opensource PHP, en donde se debe incluir la librería de NUSOAP que se trata de un conjunto de clases en PHP para crear y consumir servicios web SOAP.

En el proyecto final de la aplicación de Llévame se crearon las siguientes carpetas en el servidor que contiene nuestro servicio web con las siguientes clases en php:

Figura 4.50 Descripción de carpetas creadas en la capa de negocio



Nota: Muestra en la figura los directorios creados en el servidor que forman parte de la capa de negocio. Elaborado por los Autores. (Agosto 2013)

En las carpetas descritas en la figura, es necesario mencionar las siguientes clases relevantes:

- Dentro de la carpeta/config se encuentra la clase **con.class.php** la cual permite la conexión a la base de datos MySQL donde se almacena la información.
- Dentro de la carpeta/function se encuentra tres clases las cuales detallaremos a continuación:
 - ✓ **dataUserCommonQueries.php:** Contiene las funciones que realizan consultas generales a la base de datos.
 - ✓ **dataUserOperation.php:** Contiene las funciones que realizan operación transaccional a la base de datos como son el registro de usuario, ingreso de publicaciones, solicitudes de amistad, etc.

- ✓ **dataUserQueries.php:** Contiene las funciones que realizan consultas específicas a la base de datos como son consulta de datos personales, consulta de publicaciones, consulta de notificaciones, consulta de amigos, etc.
- Dentro de la carpeta **raíz** se encuentra las clases como son: WSDataCommonQueries.php, WSDataQueries.php, WSDataOperations.php para permitir publicar los métodos web para que invoque las funciones creadas en la carpeta/function.

4.7 Capa de presentación (Interfaz Gráfica)

4.7.1 Descripción General de Ventanas y menús de la Aplicación móvil

Figura 4.51 Pantalla de inicio de la aplicación



Nota: Muestra la pantalla inicial de la aplicación móvil. Elaborado por los Autores. (Agosto 2013)

Al hacer clic en el icono de la aplicación móvil nos aparecerá la pantalla de inicio de la aplicación durante unos segundos, para luego mostrarnos la pantalla de inicio de sesión de la aplicación.

Figura 4.52 Pantalla de inicio de sesión de la aplicación



Nota: Muestra la pantalla para iniciar sesión y poder acceder a la aplicación móvil.
Elaborado por los Autores. (Agosto 2013)

En la pantalla de inicio de sesión al usuario se le muestra un menú con tres opciones:

- Login with Facebook
- Regístrate con tu email
- Entrar

Al seleccionar la primera opción se mostrara la pantalla para que inicie sesión con Facebook ya que la aplicación re-direccionara al Facebook para que el usuario ingrese sus credenciales de acceso para que nuestra aplicación móvil pueda verificar que sus datos son correctos y pueda permitir el registro automático y el acceso a la aplicación.

En esta interfaz se muestra también imágenes aleatorias de la ciudad de Guayaquil que permiten a la aplicación tener un toque atractivo ante la vista del usuario, esta ventana aparecerá la primera vez que se ejecute la aplicación o cuando el usuario al estar conectado se desconecta.

Figura 4.53 Pantalla de registro de la aplicación



Nota: Muestra la pantalla para registrar los datos personales del usuario.
Elaborado por los Autores. (Agosto 2013)

Al seleccionar la segunda opción se mostrar la pantalla para que el usuario se pueda registrar a la aplicación, donde podrá ingresar sus datos personales como son sus nombres, apellidos, email, contraseña, fecha de nacimiento, teléfono y sexo.

Toda esta información será obligatoria el cual el sistema validará.

Adicional a esto, el usuario podrá incluir de manera opcional su foto ya sea seleccionando una foto de galería o una foto al instante, la cual se mostrará en su foto de perfil, cuando realice publicaciones, o escriba una publicación. Si el usuario no sube ninguna foto a la aplicación mostrará el siguiente icono.

Figura 4.54 Icono de Usuario de la aplicación móvil



Nota: Icono cuando el usuario no tiene foto de perfil.
Elaborado por los Autores. (Agosto 2013)

Al hacer clic en el botón “Registrarse en la aplicación” el usuario podrá grabar sus datos e iniciará sesión en la aplicación.

Figura 4.55 Pantalla de Login de la aplicación



Nota: Muestra la pantalla para que un usuario registrado puede iniciar sesión a la aplicación.
Elaborado por los Autores. (Agosto 2013)

Al seleccionar en la opción de “ENTRAR” se mostrará la pantalla de ingreso a la aplicación, donde solo los usuarios registrados en la aplicación podrán ingresar al sistema con su clave y contraseña la cual será validada en la aplicación, si la contraseña es incorrecta mostrará un mensaje de error, por el contrario si las credenciales del usuario son correctas al hacer clic en el botón de Ingresar se re-direccionará en la pantalla del Home, ya que previamente se registró.

Debido a que muchos formatos de aplicaciones son similares, el usuario está ya familiarizado con este tipo de interacción, lo cual facilita bastante el uso de las interfaces que esta posee ayudando a tener una mejor interacción y rapidez al momento de ver los datos y resultados que este necesite.

Figura 4.56 Pantalla del Home de la aplicación



Nota: Muestra la pantalla donde el usuario puede ver las publicaciones suyas y de sus amigos, y enterarse a que amigos agregan. Elaborado por los Autores. (Agosto 2013)

En la pantalla del “Home” podrá acceder una vez que el usuario haya ingresado, o se haya registrado a la aplicación mediante las opciones mencionadas anteriormente, esta opción contiene la siguiente información:

- ✓ Mis publicaciones de estado y las publicaciones de estado de mis amigos agregados
- ✓ Las publicaciones de donde se está dirigiendo mis amigos, para enviarle la solicitud del llevar.
- ✓ Los amigos agregados por mis amigos

Mensaje de Bienvenida cuando ingresa por primera vez.

Figura 4.57 Icono de Ubicación del lugar

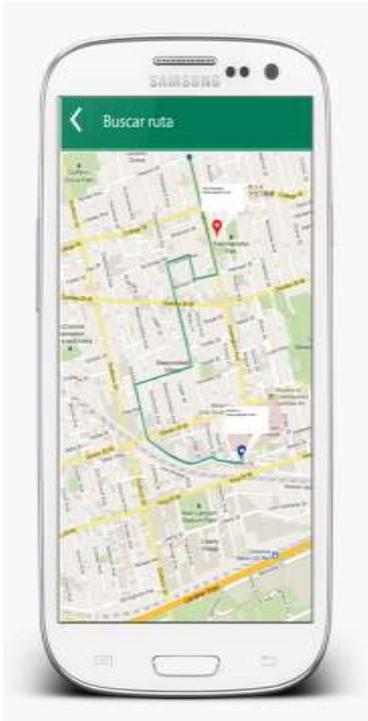


Nota: Icono cuando se ubica una coordenada. Elaborado por los Autores. (Septiembre 2013)

Para que el usuario pueda enviar la solicitud del llevar, debe hacer clic en el botón de llevar la cual le mandará una solicitud al usuario para que pueda aceptarla o rechazarla,

el usuario al tener un Timeline facilita a que se puedan ver resultados en tiempo para así facilitar la toma de decisiones del mismo.

Figura 4.58 Pantalla de Búsqueda de Ruta de la aplicación



Nota: Muestra la pantalla cuando queremos mandar la solicitud para que nos lleva a un lugar, donde podemos trazar la ruta del usuario que enviare la solicitud. Elaborado por los Autores. (Septiembre 2013)

El usuario en esta publicación podrá hacer clic en el icono del mapa, la cual re-direccionará a una pantalla donde el usuario podrá visualizar la ubicación y el trazo de ruta de su amigo hacia su ubicación.

Al hacer clic en el botón se retrocede a la pantalla anterior que es la pantalla del Home.

Figura 4.59 Icono de pestaña de retroceso



Nota: Icono cuando se ubica una coordenada. Elaborado por los Autores. (Septiembre 2013)

En la parte superior de la pantalla del home se puede visualizar un menú para que el usuario pueda realizar las siguientes acciones.

- Mapa
- Mis Amigos
- Configuración

Figura 4.60 Menú de la pantalla del Home



Nota: Muestra el menú de la pantalla principal para poder acceder a la opción de mapas, mis amigos, y configuración. Elaborado por los Autores. (Septiembre 2013)

Estas son las opciones básicas que la aplicación móvil presenta al usuario para realizar las actividades que el usuario necesita para interactuar con otras personas y saber ubicaciones de lugares y demás personas cercanas a él.

En este menú el usuario al hacer clic aparecerá la pantalla de notificaciones, en la cual se encuentran todas las notificaciones que la aplicación proporciona que mantendrán al usuario al tanto de las actividades que pueda realizar en la aplicación.

Figura 4.61 Icono de la aplicación móvil Llévame



Nota: Icono de la aplicación. Elaborado por los Autores. (Septiembre 2013)

Figura 4.62 Pantalla de Notificaciones de la aplicación



Nota: Muestra el menú de las notificaciones recibidas las cuales pueden darse cuando han aceptado tus solicitudes de amistad o de llévame. Elaborado por los Autores. (Septiembre 2013)

El usuario en la pantalla de notificaciones podrá visualizar las siguientes notificaciones:

- ✓ Cuando un amigo ha aceptado tu solicitud amistad.
- ✓ Cuando un amigo ha aceptado tu solicitud de llévame.
- ✓ Cuando te han enviado una solicitud llévame, donde el usuario podrá aceptar o ignorar la solicitud enviada de su amigo.

En este menú el usuario al hacer clic en el siguiente botón aparecerá la pantalla de publicación, que permitirá al usuario re direccionará a la pantalla de publicaciones.

Figura 4.63 Icono de escribir publicaciones



Nota: Icono para escribir una publicación de estado o de visitar un lugar. Elaborado por los Autores. (Septiembre 2013)

En esta pantalla el usuario podrá escribir una publicación de estado para poder compartirlo con sus amigos, con solo escribir un mensaje y que al presionar el botón

Publicar se grabará el mensaje y se re direccionará a la pantalla del Home donde podrá visualizar el mensaje publicado.

Figura 4.64 Pantalla de Publicación de la aplicación



Nota: Muestra la pantalla donde te permite publicar un mensaje de estado o puedes publicar un mensaje de visita.
Elaborado por los Autores. (Septiembre 2013)

Además de realizar la publicación de estado, también el usuario podrá publicar a dónde quiere dirigirse, permitiendo que sus amigos puedan observar esta publicación y de esta manera tú puedas compartir tu vehículo. Al presionar el botón se mostrara la pantalla para que el usuario pueda buscar el lugar a donde quiera dirigirse.

Figura 4.65 Icono para Seleccionar un lugar



Nota: Icono para seleccionar un lugar cuando se publica un lugar de visita.
Elaborado por los Autores. (Septiembre 2013)

Figura 4.66 Pantalla de Búsqueda del lugar en la aplicación



Nota: Muestra la pantalla donde te permite seleccionar el lugar de destino que se desea publicar.
Elaborado por los Autores. (Septiembre 2013)

Al seleccionar un lugar de la lista de sitios cercanos o al buscar algún otro lugar, esta se re-direccionará a la pantalla de publicación con el lugar que se seleccionó para que pueda publicar el lugar donde se quiere dirigir.

Al hacer clic en el botón de “Llevar”, se grabará la publicación y se re direccionará a la pantalla del Home donde podrá visualizar el mensaje que has publicado, y también las publicaciones de tus amigos.

En esta pantalla también se muestran todo los lugares segmentados por país (Ecuador) y la ciudad dependiendo de la ciudad donde el usuario se encuentre, sea esta Quito, Cuenca o Guayaquil que son las principales en el Ecuador.

Figura 4.67 Pantalla de Mapa en la aplicación

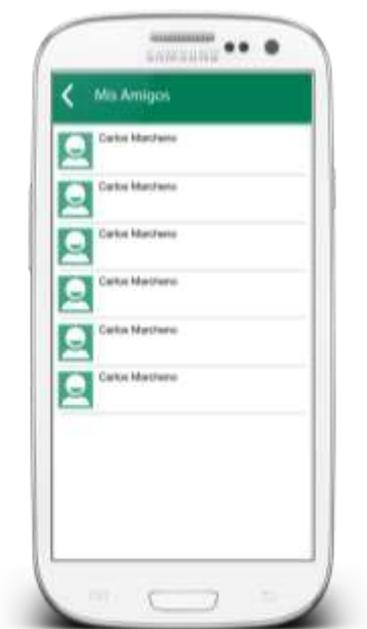


Si se selecciona en el menú del “Home” y nos dirigimos a la opción de visualización del “Mapa” donde el usuario podrá realizar las consultas de ubicación:

- ✓ Trazo de ruta y tiempo estimado hacia el lugar donde se dirige.
- ✓ Trazo de ruta y tiempo estimado de las personas que voy a llevar.
- ✓ Trazo de ruta y tiempo estimado de mis amigos cercanos en un rango de cobertura de 5 km.

Si se selecciona en el menú del “Home”, la opción de “Mis amigos”, el usuario podrá consultar sus amigos agregados, y además tendrá la opción eliminar cualquier amigo agregado.

Figura 4.68 Pantalla de mis amigos en la aplicación



Nota: Muestra la pantalla donde se visualiza los amigos que he agregado en la aplicación.
Elaborado por los Autores. (Septiembre 2013)

Si se selecciona en el menú del Home en la opción de “Configuración”, se mostrará una pantalla donde podrá realizar las siguientes operaciones importantes:

- Buscar Amigos
- Configuración de la Cuenta
- Compartir con otras redes
- Cerrar sesión

En esta pantalla solamente se encuentran los amigos que se han aprobado las solicitudes de amistad, en la aplicación las solicitudes que no son aprobadas ni rechazadas se mantienen en la base de datos por tanto se seguirán mostrando ante el usuario hasta que este escoja una opción.

Figura 4.69 Pantalla de Configuración



Nota: Muestra la pantalla donde están las opciones de configuración del usuario como son cambiar la contraseña, editar el perfil, cerrar sesión entre otras. Elaborado por los Autores. (Septiembre 2013)

En esta pantalla se encuentran todas las configuraciones que tendrá el usuario para poder modificar desde cambiar datos personales hasta agregar amigos tanto de redes sociales como desde la misma aplicación.

Si el usuario hace clic en Buscar amigos, se mostrará un menú con dos opciones:

- **Tu perfil.-** Permite modificar los datos personales del usuario como son: nombres, apellidos, número de teléfono, correo, biografía, sexo. Al pulsar el botón de Guardar grabará la información ingresada.
- **Cambiar Contraseña.-** Permite modificar su contraseña, el cual validará la contraseña anterior. Al pulsar el botón de “Guardar” la aplicación grabará la información ingresada.

Figura 4.70 Pantalla de Configuración de la Cuenta de la aplicación



Nota: Muestra la pantalla donde están las opciones de configuración de la cuenta como son cambiar la contraseña, editar sus datos personales. Elaborado por los Autores. (Septiembre 2013)

Cuando el usuario se registro en la aplicación los datos fueron previamente guardados, esta información se muestra en esta pantalla, para que por esta manera puedan ser editados para poder actualizar los datos para ayudar al usuario.

Además, en esta opción el usuario podrá cambiar su foto de perfil, donde podrá seleccionarla desde la galería de imágenes del teléfono móvil , o tomarse una foto en ese instante, tambien cuenta la funcionalidad de recortar la imagen.

Figura 4.71 Pantalla de edición de perfil



Nota: Muestra la pantalla donde se cambian los datos personales del usuario. Elaborado por los Autores. (Septiembre 2013)

Figura 4.72 Pantalla de cambio de contraseña



Nota: Muestra la pantalla donde se cambia la clave con que el usuario ingresa a la aplicación. Elaborado por los Autores. (Septiembre 2013)

Figura 4.73 Pantalla para buscar amigos en la aplicación



Nota: Muestra la pantalla donde permites seleccionar las opciones de revisar las solicitudes de amistad que te enviaron, búsqueda de amigos como Facebook, o del sistema. Elaborado por los Autores. (Septiembre 2013)

- **Solicitudes de amigos.-** Permite visualizar las solicitudes de otros usuarios, mediante las cuales puedo aprobar o rechazar la solicitud de amistad.
- **Buscar amigos en Facebook.-** Permite buscar los amigos que has agregado en tu red social Facebook, en caso de encontrarse mandarle la solicitud de amistad.
- **Buscar amigos en la aplicación.-** Permite buscar todos los amigos que se encuentran registrados en la aplicación móvil.

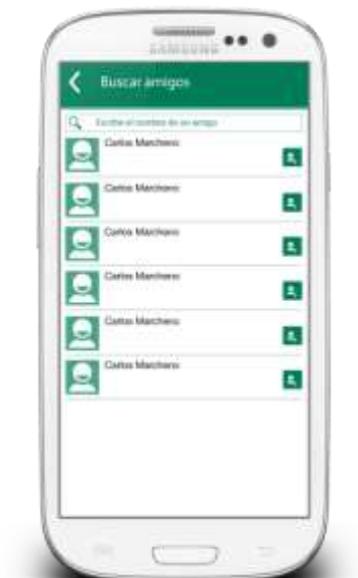
En la pantalla de “Solicitudes”, estas solicitudes permanecerán visibles hasta que el usuario las acepte o las deniegue, si un usuario vuelve a enviar la misma solicitud en la base de datos solamente cambiara el estado para que pueda ser visualizada.

Figura 4.74 Pantalla de Solicitudes de Amistad



Nota: Muestra la pantalla para aceptar o rechazar las solicitudes enviadas por otros usuarios. Elaborado por los Autores. (Septiembre 2013)

Figura 4.75 Pantalla de Buscar todos los amigos



Nota: Muestra la pantalla para aceptar o rechazar las solicitudes enviadas por otros usuarios. Elaborado por los Autores. (Septiembre 2013)

Si seleccionó la opción de “Cerrar Sesión”, permitirá salir de la aplicación y redireccionará a la pantalla de inicio para que vuelva a iniciar sesión.

4.7.2 Instalación de la Aplicación móvil Llévame

La instalación de la aplicación móvil es sumamente fácil para ello solo es necesario subirla al Play Store de google para que los usuarios puedan descargarlo, otra manera es instalar el archivo que se genera al compilar la aplicación(**archivo.apk**)para poder instalarlo en su dispositivo móvil Android. Al momento de instalar la aplicación esta pedirá los permisos necesarios.

4.8 Pruebas y Resultados

Realizamos una prueba de campo durante una semana, instalamos la aplicación en el celular de 10 personas con diferentes características, con amigos y conocidos de confianza, de los cuales 4 tenían automóvil y 6 no poseían.

Usaron la aplicación en el horario de 17:00 - 18:00 durante la semana donde hay mayor congestión de automóviles. A partir de este proceso, se esperó reparar posibles errores que se puedan producir y, a su vez mejorar la aplicación.

4.8.1 Resultados de las pruebas

Tabla 4.21 Tabla de resultados de pruebas iniciales

PRUEBA	RESULTADOS	OBSERVACIONES
Navegación del usuario en la aplicación	Debido al alto uso de aplicaciones móviles los usuarios no tuvieron inconvenientes.	
Conectividad a internet móvil	Los tiempos de respuestas fueron variantes debido a la conexión de internet de los diferentes dispositivos	Tener una conectividad LTE disminuirá el tiempo de respuesta y conexión del

		dispositivo.
Ubicación (GPS)	Se produjo un error al momento de calcular la ubicación de la persona	Este error fue reprogramado y solucionado posteriormente.
Rutas	Debido a las diferentes rutas el usuario tiene la decisión de escoger el mejor tiempo y camino para trasladar a su amigo	
Seguridad	Como toda aplicación se solicita cierta información básica del usuario de tal manera que los usuarios tienen confianza de dar dicha información.	

Nota: En esta tabla muestra los casos de usos de los respectivos resultados obtenidos desde la aplicación. Elaborado por los Autores. (Septiembre 2013)

4.8.2 Evaluación Final

Finalizada la etapa de pruebas, luego de obtener los resultados, podemos definir la siguiente evaluación:

Tabla 4.22 Parámetros para funcionamiento de la aplicación

PARAMETRO	EVALUACIÓN
Herramienta	El uso de la arquitectura y el modelo facilitó hacer modificaciones rápidas y mejoras a la aplicación durante las pruebas. Por tanto, obtiene un sistema modular y escalable. La utilización de MySQL permite tener una base robusta, segura, rápida y estable.

Documentación	La aplicación tiene una sección de Acerca de donde indica una breve descripción.
Seguridad	La seguridad de la información es muy importante para los usuarios que deben usar este tipo de aplicaciones, por tanto nos hemos asegurado de que las claves de acceso pasen por un método de encriptación y los datos que se consultan pasen de manera segura.
Navegación	Los resultados eran los esperados, debido al alto uso de aplicaciones móviles en la vida diaria, los usuarios se adaptaron rápidamente al uso de la misma.

Nota: En esta tabla se muestra los diversos datos de usabilidad que se pueden dar para que el usuario tenga una fácil manera de usarla Elaborado por los Autores. (Septiembre 2013)

4.9 Requerimientos de Hardware y Software

Los requisitos necesarios para la aplicación en tanto al hardware como en el software son los siguientes:

- Tener Sistema Operativo Android desde la versión 3.1 en adelante
- Soporte a GPS
- Poseer paquetes de datos o acceso a Internet.

Por el momento la aplicación funciona en dispositivos de gama media y alta.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Al permitir trasladar a un familiar o amigo cercano al lugar donde te diriges, permite que existan menos automóviles en las calles, el cual ayuda a ser parte de la solución del problema que existe en el Ecuador, también permite incentivar la unión y la cooperación de las personas, al sentirse seguras en transportarse a sus destinos.

El desarrollar una aplicación móvil gratuita la cual puede ser accedida por cualquier persona que posea un dispositivo con tecnología, ayuda a incentivar a que las personas se vayan adaptando el correcto uso de la aplicación.

La solución propuesta no solo ayudará a mejorar el transporte de las personas también contribuirá en la conservación y mejora del medio ambiente reduciendo considerablemente el uso de automóviles por parte de algunas personas.

El uso de este tipo de aplicaciones que aporten a la comunidad de tal manera que nos ayude a ser más cooperativos entre nosotros permitirá una mejor comunicación y socialización entre las personas para así poder vivir de una manera segura.

5.2 Recomendaciones

Se recomienda el buen uso de la aplicación para fines sociales o comunitarios, y no para fines delictivos que buscan apropiarse del bien ajeno o hacer algún tipo de daño físico a alguna persona.

El proveer información de los datos personales de los usuarios ayudan a dar confianza a los demás usuarios que desean ser parte de tu red social, por esa razón se recomienda que sus información sea verdadera y correcta. Toda información será privada y es garantizada su privacidad y confidencialidad.

Para poder evitar cualquier tipo de accidente de tránsito es recomendable que las personas sean precavidas y prudentes en el uso de la aplicación al momento de visualizar las rutas y ubicaciones de los amigos que se está llevando a sus destinos.

BIBIOGRAFIA

- Expreso, E. (07 de Julio de 2012). Guayaquil. *Pico y placa, ¿una solución?*
- García, Y. (9 de Mayo de 2013). *Ciudadanía Informada*. Retrieved 12 de Septiembre de 2013 from http://www.ciudadaniainformada.com/noticias-ciudadania-ecuador0/noticias-ciudadania-ecuador/ir_a/ciudadania-informada/article//calles-de-guayaquil-saturadas-de-vehiculos.html
- W.Frank Ableson, R. S. (2012). *Android in Action Second Edition*. Connecticut: Manning Publications.
- Condesa. (10 de Mayo de 2012). *Androideity*. Retrieved 1 de Septiembre de 2013 from <http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/>
- Jose Guillermo Valle, J. G. (2005). *Definición arquitectura clientes - servidor*. Guayaquil.
- Jorge. (2 de Mayo de 2011). *Lenguaje PHP*. Retrieved 2 de Septiembre de 2013 from <http://lenguajephpjc.blogspot.com/2011/05/caracteristicas-del-lenguaje-php.html>
- Suarez, V. (29 de Junio de 2007). *El Diario de Santo Domingo*. Retrieved 30 de Septiembre de 2013 from Gases y Automoviles: <http://diariodesantodomingo-salud.blogspot.com/2007/06/los-gases-de-los-automoviles.html>
- Andreina Silva, J. M. (Septiembre de 2013). *slideshare*. From slideshare: <http://www.slideshare.net/andreinasilva2/modelo-cliente-servidor-trabajo>
- Programación.Net. (Enero de 2013). *Programación.Net*. Retrieved Septiembre de 2013 from Versiones de Android y sus APIs: <http://www.tuprogramacion.com/programacion/versiones-de-android-y-sus-apis/>
- Condesa. (Noviembre de 2011). *Androideity*. Retrieved Septiembre de 2013 from Androideity: <http://androideity.com/2011/11/08/creando-un-servicio-propio-en-android/>
- Barco, A. (31 de Diciembre de 2006). *Blogger*. Retrieved 9 de Septiembre de 2013 from <http://arquitecturaorientadaaservicios.blogspot.com/2006/12/articulo-tecnolgico-wsdl-el-contrato-de.html>
- Benoffi, G. (18 de Mayo de 2011). *Blogs Gonzalo Benoffi*. Retrieved 2 de Septiembre de 2013 from Blogs Gonzalo Benoffi: <http://gonzalobenoffi.com.ar/2011/05/18/utilizar-el-gps-en-android-y-mostrar-ubicacin-actual/>
- Catalán, A. (26 de Mayo de 2011). *Maestros del Web*. Retrieved 1 de Septiembre de 2013 from Maestros del Web: <http://www.maestrosdelweb.com/editorial/curso-android-geolocalizacion-utilizacion-mapas-google/>

Google. (n.d.). *Developer Android*. Retrieved 2 de Septiembre de 2013 from Developer Android: <http://developer.android.com/guide/components/index.html>

Deyan José Quintero Teran, J. O. (28 de Mayo de 2010). *Blogger*. Retrieved 2 de Septiembre de 2013 from Blogger: <http://informaciondelossistemasgps.blogspot.com/2010/05/>

Diaz, A. I. (2012). *Calculadora Geoespacial*. La Plata.

Euroresidentes. (n.d.). *GPS*. Retrieved Septiembre de 2013 from Que es el GPS: <http://www.euroresidentes.com/gps/que-es-el-gps.htm>

Gabor Hojtsy, R. M. (2005). *Manual de PHP*. Zend Technologies.

Google. (3 de Enero de 2013). *Google Developer*. Retrieved 12 de Septiembre de 2013 from <https://developers.google.com/maps/documentation/directions/?hl=es>

Google. (30 de Enero de 2013). *Google Developer*. Retrieved 4 de Septiembre de 2013 from <https://developers.google.com/places/documentation/>

Google. (29 de Enero de 2013). *Google Developer*. Retrieved 3 de Septiembre de 2013 from Static Maps: <https://developers.google.com/maps/documentation/staticmaps/?hl=es>

Google. (29 de Enero de 2013). *Google Developer*. Retrieved 2 de Septiembre de 2013 from Distance Matrix: <https://developers.google.com/maps/documentation/distancematrix/?hl=es>

Ksoap-Android. (29 de Enero de 2013). Retrieved 2 de Septiembre de 2013 from Ksoap-Android: <https://code.google.com/p/ksoap2-android/>

Hoy, D. (12 de Diciembre de 2011). *Hoy*. Retrieved 2 de Septiembre de 2013 from Hoy: <http://www.hoy.com.ec/noticias-ecuador/record-en-la-matriculacion-524115.html>

Lara, A. L. (12 de Marzo de 2013). *Saduce Blog*. Retrieved 2 de Septiembre de 2013 from Saduce Blog: <http://blog.saducelabs.com/android/google-maps-en-android/>

Marco Besteiro, M. R. (n.d.). *Universidad de Pais Basco*. Retrieved 2 de Septiembre de 2013 from <http://www.ehu.es/mrodriguez/archivos/csharp/pdf/ServiciosWeb/WebServices.pdf>

Meza, M. (Septiembre de 2010). *Miriam Meza (BASE DE DATOS)*. Retrieved Septiembre de 2013 from <http://miriammeza.wordpress.com/2010/09/03/arquitectura-cliente-servidor-de-tres-capas/>

Peláez, J. (26 de Mayo de 2006). *Arquitectura basada en capas*. Retrieved 2 de Septiembre de 2013 from <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-capas/>

Pérez, R. R. (23 de Enero de 2011). *El blog de Ricardo Ruiz de Adana Pérez*. Retrieved 3 de Septiembre de 2013 from <http://ricardoruizdeadana.blogspot.com/2011/01/unidad-iv-diseno-de-una-investigacion.html>

USR.CODE. (n.d.). *WebServices con C#*. Retrieved 2 de Septiembre de 2013 from <http://img.redusers.com/imagenes/libros/lpcu104/capitulogratis.pdf>

Universo, E. (Agosto de 2013). *Pico y Placa rige hoy en Quito* .

Vierco. (7 de Agosto de 2010). *El androide libre*. Retrieved 2 de Septiembre de 2013 from El androide libre: <http://www.elandroidelibre.com/2010/08/aprendiendo-android-v-inicializacion-a-la-api-del-gps.html>

Facebook. (7 de Junio de 2012). *Facebook Developers*. Retrieved 12 de Septiembre de 2013 from <http://developers.facebook.com/docs/android/getting-started/facebook-sdk-for-android/>

Vargas, E. B. (10 de Noviembre de 2011). *Scribd*. Retrieved Septiembre de 2013 from Scribd: <http://es.scribd.com/doc/72220912/Concepto-GPS>

ANEXOS

Anexo 1. Código de procedimiento almacenado para registrar un usuario

```
/* Hora del Sistema*/
SELECT DATE_ADD(NOW (), INTERVAL 1 HOUR) INTO LD_FSISTEMA FROM DUAL;

/* Verifica si existe un usuario vinculado a ese correo siempre y cuando no se envíe el id del usuario*/
IF LV_ID="" OR LV_ID IS NULL THEN
  SELECT DISTINCT('y') INTO LV_VALIDA FROM TAB_USERS
  WHERE EMAIL=LV_CORREO and UID_ESTADO='7';
END IF;

/* Valida si ya existe correo vinculado */
IF LV_VALIDA ='y' OR LV_VALIDA IS NOT NULL THEN
  /* MENSAJE DE ERROR USUARIO YA EXISTE */
ELSE
  /* Obtengo el usuario, el token , fecha de expiración del token */
  SELECT UID_USER,TOKEN,FEXPIRE INTO LV_IDUSER,LV_TOKEN_1,LV_FEXPIRE FROM TAB_USERS
  WHERE EMAIL=LV_CORREO AND AES_DECRYPT(PASS, 'APLLEVAME')=LV_PASS AND UID_ESTADO='7';

  /* Verifico si ya existe el usuario */
  IF LV_IDUSER <> " OR LV_IDUSER IS NOT NULL THEN

    /* Actualizo información del usuario */
    UPDATE TAB_DAT_PERSONALES SET NOMBRES= LV_NOMBRES, APELLIDOS=LV_APELLIDOS,
      TELEFONO=LV_TELEFONO, SEXO=LV_SEXO,
      NOMBRE_FOTO=LV_NOMBRE_FOTO, BIOGRAFIA=LV_BIOGRAFIA,
      FMODIFICACION=LD_FSISTEMA WHERE UID_USER=LV_IDUSER;

    /* Actualizo usuario y contraseña */
    UPDATE TAB_USERS SET EMAIL=LV_CORREO WHERE UID_USER=LV_IDUSER;

  COMMIT;
  /* MENSAJE DE ERROR USUARIO ACTUALIZADO CON EXITO */

  ELSE

    /* Ingreso el usuario*/
    INSERT INTO TAB_USERS(EMAIL,PASS,TOKEN,FTOKEN,FEXPIRE,UID_ESTADO, FCREACION)
    VALUES (LV_CORREO, AES_ENCRYPT (LV_PASS,'APLLEVAME'),LV_TOKEN,LD_FSISTEMA,
    DATE_ADD(LD_FSISTEMA,INTERVAL 2 WEEK),7,LD_FSISTEMA);

    COMMIT;

    /* Obtengo el id del usuario, el token, y la fecha de expiración del token del usuario ingresado anteriormente */
    SELECT UID_USER,TOKEN,FEXPIRE INTO LV_IDUSER_2,LV_TOKEN_2,LV_FEXPIRE_2 FROM TAB_USERS
    WHERE EMAIL=LV_CORREO AND AES_DECRYPT (PASS, 'APLLEVAME')=LV_PASS AND UID_ESTADO='7';

    /* Valida que el id del usuario no este vacío */
    IF LV_IDUSER_2 <> " OR LV_IDUSER_2 IS NOT NULL THEN

    /* Inserta datos personales cuando ingresa por primera vez */
    INSERT INTO
    TAB_DAT_PERSONALES(UID_USER,NOMBRES,APELLIDOS,FNACIMIENTO,TELEFONO,SEXO,NOMBRE_FOTO,BIO
    GRAFIA,FCREACION)
    VALUES
    (LV_IDUSER_2,LV_NOMBRES,LV_APELLIDOS,LV_FNACIMIENTO,LV_TELEFONO,LV_SEXO,LV_NOMBRE_FOTO,
    LV_BIOGRAFIA,LD_FSISTEMA);

    COMMIT;

    /* MENSAJE DE ERROR USUARIO INGRESADO CON EXITO */
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado para registrar un usuario. Elaborado por los Autores. (Septiembre 2013)

Anexo 2. Código del procedimiento almacenado para actualizar la posición del usuario

```
/* Hora del Sistema*/
SELECT DATE_ADD(NOW (), INTERVAL 1 HOUR) INTO LD_FSISTEMA FROM DUAL;

/* Verifica si existe un usuario */
SELECT DISTINCT('y') INTO LV_VALIDA FROM TAB_USERS WHERE UID_USER=LV_IDUSER AND
UID_ESTADO=7;

IF LV_VALIDA="y" THEN
  /* Verifica si caduco la coordenada de visita*/
  IF LV_STATUS='C' THEN

/* Setea la coordenada de visita con nulo */
UPDATE TAB_USERS SET UID_COORD_VISITA=NULL WHERE UID_USER=LV_IDUSER AND UID_ESTADO=7;

/* Actualiza coordenada de usuario*/
UPDATE TAB_COORD_USERS SET LATITUD=LN_LATITUD, LONGITUD= LN_LONGITUD WHERE
UID_USER=LV_IDUSER;

COMMIT;

/* MENSAJE DE ERROR USUARIO ACTUALIZACION EXITOSA */

ELSE

/* Actualiza coordenada de usuario*/
UPDATE TAB_COORD_USERS SET LATITUD=LN_LATITUD, LONGITUD= LN_LONGITUD WHERE
UID_USER=LV_IDUSER;

COMMIT;

/* MENSAJE DE ERROR USUARIO ACTUALIZACION EXITOSA */

END IF;

END IF;
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado para actualizar la coordenada de usuario y cambiar la coordenada de visita cuando el usuario llegaba a su destino. Elaborado por los Autores. (Septiembre 2013)

Anexo 3. Código del procedimiento almacenado para enviar solicitud de llevar

```
/* Hora del Sistema*/
SELECT DATE_ADD(NOW (), INTERVAL 1 HOUR) INTO LD_FSISTEMA FROM DUAL;

/* Verifica si existe un usuario */
SELECT DISTINCT('y') INTO LV_VALIDA FROM TAB_USERS WHERE UID_USER=LV_IDUSER AND
UID_ESTADO='7';

IF LV_VALIDA="y" THEN

    /* Verifica si existe el id del amigo */
    SELECT DISTINCT('y') INTO VALIDA_A FROM TAB_USERS WHERE UID_USER=LV_IDUSERAMIGO AND
    UID_ESTADO='7';

    IF VALIDA_A ='y' THEN

        /* Verifica si el id de la coordenada de visita del usuario que recibe sea el publicado actualmente*/
        SELECT UID_COORD_VISITA INTO LV_IDCOORD_VISITA2 FROM TAB_USERS WHERE
        UID_USER=LV_IDUSERAMIGO AND UID_ESTADO='7';

        IF LV_IDCOORD_VISITA=LV_IDCOORD_VISITA2 THEN

            /* Verifica si ya se ingresó la solicitud */
            SELECT DISTINCT('y') INTO VALIDA_C FROM TAB_SOLICITUDES_LLEVAME WHERE
            UID_USER_ENVIA=LV_IDUSER AND UID_USER_RECIBE=LV_IDUSERAMIGO AND
            UID_COORD_VISITA=LV_IDCOORD_VISITA AND UID_ESTADO IN ('4','6');

            IF VALIDA_C='y' THEN

                /* MENSAJE DE ERROR SOLICITUD YA ENVIADA */

                ELSE

                    /*Ingresa solicitud de llevar */

                    INSERT INTO
                    TAB_SOLICITUDES_LLEVAME(UID_USER_ENVIA,UID_USER_RECIBE,UID_COORD_VISITA,UID_ESTADO,
                    FCREACION) VALUES(LV_IDUSER,LV_IDUSERAMIGO,LV_IDCOORD_VISITA,6,LD_FSISTEMA);

                    COMMIT;

                /* MENSAJE DE ERROR SOLICITUD INGRESADA EXITOSAMENTE*/

                END IF;
            ELSE

                /* MENSAJE DE ERROR COORDENADA VISITA NO VIGENTE */

                END IF;
            ELSE

                /* MENSAJE DE ERROR AMIGO NO EXISTE */

                END IF;
            END IF;
        END IF;
    END IF;
END IF;
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado para poder enviar la solicitud de llevar a quien ha publicado un lugar de destino.
Elaborado por los Autores. (Septiembre 2013)

Anexo 4. Código de procedimiento almacenado para los tipos de publicaciones

```
/* Publicación Tipo 1 */
/* Tipo 1 => Mis publicaciones y de mis amigos*/

SELECT '1' AS TIPO,TAB.UID_USER,TAB2.NOMBRES,TAB2.APELLIDOS, LV_DESCRIPCION AS RUTA_FOTO,
IF(TAB2.NOMBRE_FOTO<>'',TAB2.NOMBRE_FOTO,'NULL')AS NOMBRE_FOTO,'NULL' AS LATITUD, 'NULL' AS
LONGITUD, TAB.NOMBRE_LUGAR,TAB.MENSAJE,'NULL' AS UID_COORD_VISITA,TAB.FCREACION AS FECHA,
'NULL' AS MARCA FROM
(SELECT C.UID_USER,C.NOMBRE_LUGAR,C.MENSAJE,C.FCREACION FROM TAB_COORD_HIST_USERS C WHERE
C.UID_USER IN (SELECT F.UID_USER_AMIGO FROM TAB_AMIGOS F WHERE F.UID_USER=LV_IDUSER AND
F.UID_ESTADO='1'
UNION SELECT U.UID_USER AMIGO FROM TAB_AMIGOS U WHERE U.UID_USER_AMIGO=LV_IDUSER AND
U.UID_ESTADO='1')
AND C.UID_ESTADO='9'
UNION SELECT M.UID_USER,M.NOMBRE_LUGAR,M.MENSAJE,M.FCREACION FROM TAB_COORD_HIST_USERS
M WHERE M.UID_USER=LV_IDUSER AND M.UID_ESTADO='9') TAB,TAB_DAT_PERSONALES TAB2 WHERE
TAB.UID_USER=TAB2.UID_USER
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de mis publicaciones de estado y la de mis amigos. Elaborado por los Autores. (Septiembre 2013)

```
/* Publicación Tipo 2 */
/* Tipo 2 => Las publicaciones de visita de mis amigos*/

SELECT '2' AS TIPO, TAB.UID_USER,TAB2.NOMBRES,TAB2.APELLIDOS, LV_DESCRIPCION AS RUTA_FOTO,
IF(TAB2.NOMBRE_FOTO<>'',TAB2.NOMBRE_FOTO,'NULL')AS NOMBRE_FOTO,
TAB.LATITUD,TAB.LONGITUD,TAB.NOMBRE_LUGAR,TAB.MENSAJE,TAB.UID_COORD_VISITA, TAB.FECHA,
TAB.MARCA FROM (
SELECT C.UID_USER,C.LATITUD,C.LONGITUD,C.NOMBRE_LUGAR,C.MENSAJE,C.UID_COORD_VISITA,
CASE
WHEN C.FMODIFICACION IS NOT NULL THEN
C.FMODIFICACION
ELSE
C.FCREACION END AS FECHA
,
CASE WHEN (SELECT U.UID_COORD_VISITA FROM TAB_USERS U WHERE U.UID_USER=C.UID_USER AND
U.UID_ESTADO=7)= C.UID_COORD_VISITA THEN 'Y'
ELSE 'N' END AS MARCA
FROM TAB_COORD_VISITORS C WHERE C.UID_USER IN(
SELECT F.UID_USER_AMIGO FROM TAB_AMIGOS F WHERE F.UID_USER=LV_IDUSER AND F.UID_ESTADO='1'
UNION
SELECT U.UID_USER AMIGO FROM TAB_AMIGOS U WHERE U.UID_USER_AMIGO=LV_IDUSER AND
U.UID_ESTADO='1')
AND C.UID_ESTADO='9') TAB,TAB_DAT_PERSONALES TAB2
WHERE TAB.UID_USER=TAB2.UID_USER
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de las publicaciones de visita de mis amigos. Elaborado por los Autores. (Septiembre 2013)

```

/* Publicación Tipo 3 */
/* Tipo 3 => Mensaje de Bienvenida */

```

```

SELECT '3' AS TIPO,TD.UID_USER,TD.NOMBRES,TD.APELLIDOS, LV_DESCRIPCION AS RUTA_FOTO,
IF(TD.NOMBRE_FOTO<>',TD.NOMBRE_FOTO,NULL')AS NOMBRE_FOTO,'NULL' AS LATITUD, 'NULL' AS
LONGITUD, 'NULL' AS NOMBRE_LUGAR, LV_MENSAJE AS MENSAJE,'NULL' AS UID_COORD_VISITA,'01-01-1990
00:00:01' AS FECHA, 'NULL' AS MARCA FROM TAB_DAT_PERSONALES TD, TAB_USERS TU
WHERE TD.UID_USER=LV_IDUSER AND TU.UID_ESTADO=7 AND TD.UID_USER=TU.UID_USER

```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de un mensaje de bienvenida a la aplicación. Elaborado por los Autores. (Septiembre 2013)

```

/* Publicación Tipo 4 */
/* Tipo 4 =>Query que retorna la descripción de los amigos agregados */

```

```

SELECT '4' AS TIPO,T1.UID_USER_AMIGO UID_USER,T3.NOMBRES NOMBRES,T3.APELLIDOS
APELLIDOS,LV_DESCRIPCION AS RUTA_FOTO,IF(T3.NOMBRE_FOTO<>',T3.NOMBRE_FOTO,NULL')AS
NOMBRE_FOTO,'NULL' AS LATITUD, 'NULL' AS LONGITUD,'NULL' AS NOMBRE_LUGAR , CONCAT('HA
AGREGADO A ',INITCAP(CONCAT(T2.NOMBRES,' ',T2.APELLIDOS))) AS MENSAJE,'NULL' AS
UID_COORD_VISITA,T1.FMODIFICACION FECHA,'NULL' AS MARCA FROM (
SELECT A.UID_USER_AMIGO,A.UID_USER,A.FMODIFICACION FROM TAB_AMIGOS A WHERE
A.UID_USER_AMIGO IN(
SELECT F.UID_USER_AMIGO FROM TAB_AMIGOS F WHERE F.UID_USER=LV_IDUSER AND F.UID_ESTADO='1'
UNION
SELECT U.UID_USER AMIGO FROM TAB_AMIGOS U WHERE U.UID_USER_AMIGO=LV_IDUSER AND
U.UID_ESTADO='1')
AND A.UID_ESTADO='1' AND A.UID_USER<>LV_IDUSER ORDER BY A.FMODIFICACION) T1,
(SELECT C.UID_USER,D.NOMBRES,D.APELLIDOS,D.NOMBRE_FOTO FROM TAB_USERS C,
TAB_DAT_PERSONALES D
WHERE C.UID_USER=D.UID_USER AND C.UID_ESTADO=7) T2,
(SELECT E.UID_USER,F.NOMBRES,F.APELLIDOS,F.NOMBRE_FOTO FROM TAB_USERS E,
TAB_DAT_PERSONALES F
WHERE E.UID_USER=F.UID_USER AND E.UID_ESTADO=7) T3
WHERE T1.UID_USER=T2.UID_USER
AND T1.UID_USER_AMIGO=T3.UID_USER
) TAB7

```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de la descripción de los amigos que agregan. Elaborado por los Autores. (Septiembre 2013)

Anexo 5. Código de procedimiento almacenado para los tipos de notificaciones

```
/* Publicación Tipo 1 */
/* Tipo 1 notificaciones cuando te han agregado */

SELECT      '1'      AS      TIPO,D.UID_USER,D.NOMBRES,D.APELLIDOS,LV_DESCRIPCION      AS
RUTA_FOTO,IF(D.NOMBRE_FOTO<>'',D.NOMBRE_FOTO,'NULL') NOMBRE_FOTO, U.FMODIFICACION AS FECHA,
IF(U.UID_ESTADO=1,'HA ACEPTADO TU SOLICITUD DE AMISTAD.','HA RECHAZADO TU SOLICITUD DE
AMISTAD.') MENSAJE,'NULL' UID_SOLICITUDFROM TAB_DAT_PERSONALES D, TAB_AMIGOS U WHERE
D.UID_USER=U.UID_USER AND
U.UID_USER_AMIGO=LV_IDUSER AND
U.UID_ESTADO='1'
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de notificaciones para conocer si un amigo te ha agregado o rechazado. Elaborado por los Autores.

```
/* Publicación Tipo 2 */
/* Tipo 2 notificación cuando te han aceptado la solicitud de llévame */

SELECT      '2'      AS      TIPO,D.UID_USER,D.NOMBRES,D.APELLIDOS,LV_DESCRIPCION      AS
RUTA_FOTO,IF(D.NOMBRE_FOTO>'',D.NOMBRE_FOTO,'NULL') NOMBRE_FOTO,S.FMODIFICACION AS FECHA,
IF(S.UID_ESTADO=4,'HA ACEPTADO TU INVITACIÓN A LLEVARTE.','HA RECHAZADO TU INVITACIÓN DE
LLEVARTE.') MENSAJE, 'NULL' UID_SOLICITUDFROM TAB_DAT_PERSONALES D, TAB_SOLICITUDES_LLEVAME
S WHERE S.UID_USER_ENVIA=LV_IDUSER
AND D.UID_USER=S.UID_USER_RECIBE
AND S.UID_ESTADO IN (4, 5)
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de notificaciones para conocer si tu solicitud de llevar ha sido aceptada o ignorada. Elaborado por los Autores. (Septiembre 2013)

```
/* Publicación Tipo 3 */
/* Tipo 3 notificaciones quienes te iban solicitud llevame */

SELECT      '3'      AS      TIPO,D.UID_USER,D.NOMBRES,      D.APELLIDOS,'NULL'      AS
RUTA_FOTO,IF(D.NOMBRE_FOTO>'',D.NOMBRE_FOTO,'NULL') NOMBRE_FOTO,S.FCREACION AS FECHA,
'DESEA QUE LO LLEVES A DONDE TE DIRIJES.' AS MENSAJE, S.UID_SOLICITUD
FROM TAB_DAT_PERSONALES D, TAB_SOLICITUDES_LLEVAME S
WHERE S.UID_USER_RECIBE=10
AND D.UID_USER=S.UID_USER_ENVIA
AND S.UID_ESTADO=6) TAB1
```

Nota: En este código se visualiza los fragmentos importantes de código del procedimiento almacenado que realiza la consulta de notificaciones que me envían para que los lleve a mi lugar de destino. Elaborado por los Autores. (Septiembre 2013)

Anexo 6. Código del WebServices WSDDataQueries

En el archivo WSDDataQueries se encuentran todos los métodos para consultas específicas realizadas por el usuario, debido a que en esta función encontramos algunos métodos explicaremos a continuación algunos fragmentos de código que contiene este archivo:

```
server->wsdl->addComplexType(
    'ListPosts',
    'complexType',
    'struct',
    'all',
    "",
    array('TIPO'=>array('name'=>'TIPO','type'=>'xsd:string'),
        'UID_USER'=>array('name'=>'UID_USER','type'=>'xsd:string'),
        'NOMBRES'=>array('name'=>'NOMBRES','type'=>'xsd:string'),
        'APELLIDOS'=>array('name'=>'APELLIDOS','type'=>'xsd:string'),
        'RUTA_FOTO'=>array('name'=>'RUTA_FOTO','type'=>'xsd:string'),
        'NOMBRE_FOTO'=>array('name'=>'NOMBRE_FOTO','type'=>'xsd:string'),
        'LATITUD'=>array('name'=>'LATITUD','type'=>'xsd:string'),
        'LONGITUD'=>array('name'=>'LONGITUD','type'=>'xsd:string'),
        'NOMBRE_LUGAR'=>array('name'=>'NOMBRE_LUGAR','type'=>'xsd:st
tring'),
        'MENSAJE'=>array('name'=>'MENSAJE','type'=>'xsd:string'),

    'UID_COORD_VISITA'=>array('name'=>'UID_COORD_VISITA','type'=>'xsd:st
ring'),
    'FECHA'=>array('name'=>'FCREACION','type'=>'xsd:string'),
    'MARCA'=>array('name'=>'MARCA','type'=>'xsd:string'),
    'TIEMPO'=>array('name'=>'TIEMPO','type'=>'xsd:string')
    )
);
```

Nota: Código para definición de campos de una estructura compleja, listado de post, estos datos se muestran en el inicio. Elaborado por Los Autores (Septiembre 2013)

```

$server->wsdl->addComplexType('ListUserArray',
    'complexType',
    'array',
    "",
    'SOAP-ENC:Array',
    array(),
    array(
        array('ref'=>'SOAP-ENC:arrayType','wsdl:arrayType'=>'tns:ListUser[]')
    ),
    'tns:ListUser'
);

```

Nota: Código de encapsulación de las estructuras complejas, permite crear un arreglo para ser mostrado, esta encapsulación es del listado de Usuarios. Elaborado por Los Autores (Septiembre 2013)

```

$server->register('myfriendcoordinates', // method name
    array('idUser'=>'xsd:string'), // input parameters
    array('return' => 'tns:ListCoordinatesArray'), // output parameters
    $ns,
    $ns.'wsdl#myfriendcoordinates',
    'rpc',
    'encoded',
    'Ver las coordenadas de tus amigos ');

```

Nota: Código que me permite registrar el nombre de una función y donde indicamos parámetros de entrada y salida, este código pertenece a las coordenadas de los amigos. Elaborado por Los Autores (Septiembre 2013)

Anexo 7. Código del web Services WSDDataOperations

En el archivo WSDDataOperations se encuentran todos los métodos para mantenimiento, debido a que en esta función encontramos algunos métodos explicaremos a continuación algunos fragmentos de código que contiene este archivo:

```
$server->register('publishStatus', array(
    'idUser' => 'xsd:string',
    'lugar' => 'xsd:string',
    'mensaje' => 'xsd:string',
    'latitud' => 'xsd:string',
    'longitud' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#publishStatus', 'rpc', 'encoded', 'Publicar mensaje del usuario');

$server->register('publishStatusVisitors', array(
    'idUser' => 'xsd:string',
    'lugar' => 'xsd:string',
    'mensaje' => 'xsd:string',
    'latitud' => 'xsd:string',
    'longitud' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#publishStatusVisitors', 'rpc', 'encoded', 'Publicar mensaje de visita del
usuario');
```

Nota: Código donde registramos las funciones de publicación por parte del usuario, publicar estado y publicar Ubicación. Elaborado por Los Autores

```
$server->register('sendRequestFriend', array(
    'idUser' => 'xsd:string',
    'idUserFriend' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#sendRequestFriend', 'rpc', 'encoded', 'Enviar solicitud de amistad');

$server->register('refreshLocation', array(
    'idUser' => 'xsd:string',
    'latitud' => 'xsd:string',
    'longitud' => 'xsd:string',
    'status' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#refreshLocation', 'rpc', 'encoded', 'Actualiza ubicación del usuario');
```

Nota: Código donde registramos las funciones de envío de solicitud de amistad y actualización de ubicación del usuario. Elaborado por Los Autores (Septiembre 2013)

```

$server->register('sendRequestFriend', array(
    'idUser' => 'xsd:string',
    'idUserFriend' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#sendRequestFriend', 'rpc', 'encoded', 'Enviar solicitud de amistad');

$server->register('addFriend', array(
    'idUser' => 'xsd:string',
    'idUserFriend' => 'xsd:string',
    'status' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#addFriend', 'rpc', 'encoded', 'Aprueba o Rechaza la solicitud de Amistad');

$server->register('addRequest', array(
    'idUser' => 'xsd:string',
    'idSolicitud' => 'xsd:string',
    'status' => 'xsd:string'), array('return' => 'tns:Msj'), $ns, $ns .
    'wsdl#addRequest', 'rpc', 'encoded', 'Aprueba o Rechaza la solicitud de llévame');

```

Nota: Código donde registramos el envío de solicitud de amistad, la confirmación o negación de la solicitud de amistad y la confirmación o negación de la solicitud de llevar. Elaborado por Los Autores (Septiembre 2013)

Anexo 8. Código del web Services WSCCommonQueries

En el archivo WSCCommonQueries se encuentran todos los métodos relacionados con consultas generales o comunes de la base de datos, debido a que en esta función encontramos algunos métodos explicaremos a continuación algunos fragmentos de código que contiene este archivo:

```

$server->wsdl->addComplexType('ListParams', 'complexType', 'struct', 'all', "",
    array('DESCTIPO' => array('name' => 'DESCTIPO', 'type' => 'xsd:string'),
        'DESCRIPCION' => array('name' => 'DESCRIPCION', 'type' => 'xsd:string')));

$server->wsdl->addComplexType('ListCoordinate', 'complexType', 'struct', 'all',
    "", array(
    'UID_USER' => array('name' => 'UID_USER', 'type' => 'xsd:string'),
    'LATITUD' => array('name' => 'LATITUD', 'type' => 'xsd:string'),
    'LONGITUD' => array('name' => 'LONGITUD', 'type' => 'xsd:string')));

```

Nota: Código para definición de campos de una estructura compleja, Listado de coordenadas y Listado de parámetros que se utilizan en la aplicación. Elaborado por los Autores (Septiembre 2013)

```

$server->wsdl->addComplexType('ListParamsArray', 'complexType', 'array', "",
'SOAP-ENC:Array', array(), array(array('ref' => 'SOAP-ENC:arrayType',
'wsdl:arrayType' => 'tns:ListParams[]')), 'tns:ListParams');

$server->wsdl->addComplexType('ListCoordinateArray', 'complexType', 'array', "",
'SOAP-ENC:Array', array(), array(array('ref' => 'SOAP-ENC:arrayType',
'wsdl:arrayType' => 'tns:ListCoordinate[]')), 'tns:ListCoordinate');

```

Nota: Código de encapsulación de estructuras complejas, Listado de Parámetros y Listado de Coordenadas
Elaborado por los Autores (Septiembre 2013)

```

$server->register('searchParams', // method name
array(), // input parameters
array('return' => 'tns:ListParamsArray'), // output parameters
$ns, $ns . 'wsdl#searchParams', 'rpc', 'encoded',
'Ver los parámetros para el FTP');

$server->register('searchCoordinate', // method name
array('idUser' => 'xsd:string'), // input parameters
array('return' => 'tns:ListCoordinateArray'), // output parameters
$ns, $ns . 'wsdl#searchCoordinate', 'rpc', 'encoded',
'Ver la coordenada del usuario');

```

Nota: Código de registro de funciones de los parámetros del Ftp necesarios para guardar la foto y ver coordenadas del Usuario. Elaborado por Los Autores (Septiembre 2013)