

Universidad Mayor de San Andrés  
Facultad de Ciencias Puras y Naturales  
Carrera de Informática



PROYECTO DE GRADO

***“Sistema de Información para el control de rutas del  
transporte público vehicular de la ciudad de El Alto vía  
telefonía móvil”***

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

Postulante: Freddy López Gómez  
Tutor Metodológico: Mg. Sc. Miguel Cotaña Mier  
Asesor: Lic. Celia E. Tarquino Peralta

**La Paz – Bolivia  
2014**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**



**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## **DEDICATORIA**

A mis padres Raúl y Martha

A mi esposa Marizol

A mi hijo Jhosua

## **AGRADECIMIENTOS**

Agradecer al Lic. Miguel Cotaña Pomier por toda su colaboración y comprensión a lo largo de la elaboración del presente documento.

Agradecer a la Lic. Celia Tarquino Peralta, por todo el tiempo invertido en la elaboración de mi proyecto, sus recomendaciones, el conocimiento, material compartido y toda la colaboración brindada.

A la Universidad Mayor de San Andrés, a la carrera de Informática y a los docentes de esta carrera, por otorgarme el conocimiento y la formación profesional.

Agradecer al Gobierno Autónomo Municipal de El Alto, en especial al Ing. Rodrigo Calle Peñaloza, por permitirme realizar mi proyecto y brindarme el apoyo para la elaboración del presente documento.

Agradecer a mis padres Raul y Martha por apoyarme, guiarme y sobre todo amarme.

Agradecer a mi esposa Marizol por el apoyo incondicional que me da día a día.

Agradecer a mi hijo Jhosua Enrique por ser la personita que me inspira a seguir adelante sobre todas las cosas.

A mis compañeros de trabajo André y Víctor por apoyarme en la etapa más complicada de la elaboración de este documento.

Para todos ellos, muchas gracias.

## **RESUMEN**

La constante investigación tecnológica nos permite construir nuevas soluciones orientadas al servicio de los ciudadanos y a la resolución de sus problemas. Uno de los problemas más difíciles de solucionar es el transporte público, que por diversos motivos fueron creciendo hasta el punto de convertirse en una costumbre de la ciudadanía, esa costumbre del usuario y conductor de trasladarse de un lugar a otro haciendo el uso de tramos, es por el cual que se vio conveniente hacer uso de las tecnologías y desarrollar una aplicación móvil para brindar información de las rutas del transporte público vehicular, en este caso se lo implemento en el Gobierno Autónomo Municipal de El Alto.

Los resultados obtenidos a partir de las pruebas en la aplicación, el interés y aceptación de los usuarios nos indican que es posible la construcción de estas soluciones, la introducción de nuevas tecnologías y su uso al servicio de los ciudadanos.

## **ABSTRACT**

The constant technological research allows us to build new solutions aimed to serve citizens and solving their problems. One of the most difficult problems to solve is public transport, which for various reasons were growing to the point of becoming a habit of citizenship, that custom user and host of moving from one place to another by using sections, which is why it was appropriate to make use of technologies and develop a mobile application to provide route information of vehicular transport, in this case I implement in the Autonomous Municipal Government of El Alto.

The results obtained from testing the application, interest and acceptance from users indicate that it is possible to build these solutions, the introduction of new technologies and their use in the service of citizens.

## CONTENIDO

1. INTRODUCCIÓN.....	1
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES.....	2
1.2.1. CRUZERO.....	3
1.2.2. TU GAMEA.....	3
1.2.3. SIG.....	4
1.3. PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1. FORMULACIÓN DEL PROBLEMA.....	5
1.3.2. PROBLEMAS SECUNDARIOS.....	5
1.4. PLANTEAMIENTO DE OBJETIVOS .....	6
1.4.1. OBJETIVO GENERAL.....	6
1.4.2. OBJETIVOS ESPECÍFICOS.....	6
1.5. JUSTIFICACIÓN.....	6
1.5.1. TÉCNICA.....	6
1.5.2. SOCIAL.....	7
1.5.3. PRÁCTICA.....	8
1.5.4. ECONÓMICA.....	8
1.6. ALCANCE.....	8
2. MARCO TEÓRICO.....	9
2.1. TELEFONIA MOVIL.....	9
2.2. SISTEMAS OPERATIVOS PARA TELEFONOS MOVILES.....	10
2.2.1. COMPONENTES DE LOS SO MÓVILES.....	11
2.2.1.1. KERNEL.....	12
2.2.1.2. MIDDLEWARE.....	12
2.2.1.3. ENTORNO DE EJECUCION DE APLICACIONES ..	13
2.2.1.4. INTERFAZ DE USUARIO .....	14
2.2.2. SOS EN EL MERCADO.....	14

2.3.	SELECCIÓN DE LA HERRAMIENTA DE DESARROLLO. ....	15
2.3.1.	SO ANDROID. ....	15
2.3.2.	APRENDIZAJE. ....	16
2.3.3.	RELEVANTE. ....	17
2.3.4.	CODIGO ABIERTO. ....	17
2.3.5.	INTRODUCCION A LA PLATAFORMA ANDROID.....	18
2.3.5.1.	ARQUITECTURA.....	18
2.3.5.2.	DESARROLLO EN LA PLATAFORMA ANDROID .	21
2.3.5.3.	ENTORNO DEL DESARROLLO.....	21
2.4.	ASPECTOS METODÓLOGICOS .....	23
2.5.	METODOLOGÍAS DE DESARROLLO .....	24
2.6.	METODOLOGIAS AGILES PARA EL DESARROLLO DE SOFTWARE MÓVIL.....	25
2.6.1.	DESARROLLO AGIL : PROGRAMACION EXTREMA(XP) .....	26
2.6.2.	DESARROLLO AGIL: SCRUM.....	29
2.6.3.	MOBILE - D .....	31
2.6.4.	CARACTERISTICAS Y REQUERIMIENTOS DE ENTORNO MOVIL .....	34
2.6.5.	SELECCIÓN DE METODOLOGIA .....	36
2.6.5.1.	MOTIVACION .....	36
2.6.5.2.	PRINCIPIOS BASICOS METODOLOGIA MOBILE - D36	
2.6.5.2.1.	FASE DE EXPLORACIÓN.....	39
2.6.5.2.2.	FASE DE INICIALIZACIÓN .....	47
2.6.5.2.3.	FASE DE PRODUCCIÓN.....	48
2.6.5.2.4.	FASE DE ESTABILIZACIÓN.....	49
2.6.5.2.5.	FASE DE PRUEBA .....	50
2.7.	METRICAS DE CALIDAD .....	51
2.7.1.	ISO 9000 .....	51
2.7.2.	NORMAS ISO/IEC 9126.....	51

3.	DESARROLLO DE LA APLICACIÓN .....	61
3.1.	EXPLORACIÓN.....	61
3.1.1.	ESTABLECIMIENTOS DE LOS INVOLUCRADOS .....	61
3.1.2.	ALCANCE DEL PROYECTO. ....	63
3.1.3.	ESTABLECIMIENTO DEL PROYECTO.....	64
3.1.3.1.	DEFINICIÓN DE REQUERIMIENTOS INICIALES ..	64
3.1.3.1.1.	REQUERIMIENTOS FUNCIONALES .....	64
3.1.3.1.2.	REQUERIMIENTO NO FUNCIONALES .	65
3.1.3.2.	PLANIFICACIÓN .....	65
3.1.3.3.	DIAGRAMAS DE CASOS DE USO.....	65
3.1.3.4.	TARJETAS DE HISTORIAS DE USUARIO .....	68
3.1.3.5.	TARJETAS DE TAREAS .....	71
3.1.3.6.	PLANIFICACIÓN DEL DESARROLLO .....	78
3.2.	INICIALIZACIÓN.....	79
3.2.1.	CONFIGURACIÓN DEL PROYECTO.....	79
3.2.2.	DÍA DE PLANEACIÓN (ITERACIÓN 0).....	79
3.2.2.1.	NOMBRE DE ENTIDADES .....	79
3.2.2.2.	CLASES .....	79
3.2.2.3.	MÉTODOS.....	80
3.2.2.4.	VARIABLES. ....	80
3.2.2.5.	CONSTANTES.....	80
3.2.3.	DÍA DE LANZAMIENTO (ITERACIÓN 0).....	82
3.3.	PRODUCCIÓN O FASE DE PRODUCTO .....	83
3.3.1.	SEGUIMIENTO DE ITERACIONES .....	83
3.3.1.1.	PRIMERA ITERACIÓN.....	83
3.3.1.2.	SEGUNDA ITERACIÓN .....	84
3.3.1.3.	TERCERA ITERACIÓN .....	86
3.3.1.4.	CUARTA ITERACIÓN.....	86
3.3.1.5.	QUINTA ITERACIÓN .....	89

3.4.	ESTABILIZACIÓN.....	90
3.4.1.	DOCUMENTACIÓN.....	90
3.4.2.	LIBERACIÓN.....	90
3.5.	PRUEBAS DEL SISTEMA.....	90
4.	PRUEBAS DE CALIDAD Y RESULTADOS.....	91
4.1.	MÉTRICAS DE CALIDAD MODELO ISO – 9126.....	91
4.1.1.	FUNCIONALIDAD.....	91
4.1.2.	FIABILIDAD.....	95
4.1.3.	USABILIDAD.....	97
4.1.4.	EFICIENCIA.....	97
4.1.5.	MANTENIBILIDAD.....	98
5.	CONCLUSIONES Y RECOMENDACIONES.....	100
5.1.	CONCLUSIONES.....	100
5.2.	RECOMENDACIONES.....	101
5.2.1.	A LA INSTITUCIÓN.....	101
5.2.2.	PARA FUTUROS PROYECTOS.....	101
	BIBLIOGRAFÍA.....	102
	ANEXOS.....	104

## ÍNDICE DE FIGURAS

Figura 2.1: Componentes de los Sistemas Operativos Móviles.....	11
Figura 2.2: Popularidad de SO móviles .....	14
Figura 2.3: Arquitectura de capas de Android .....	19
Figura 2.4: Emulador de Android .....	22
Figura 2.5: Fases de la metodología para el desarrollo móvil Mobile - D.....	32
Figura 2.6: Ciclo de desarrollo de la metodología Mobile - D .....	38
Figura 2.7: Elementos básicos de los diagramas de casos de uso .....	40
Figura 2.8: Elementos Include .....	41
Figura 2.9: Elemento extend .....	42
Figura 2.10: Límite del sistema.....	42
Figura 2.11: Ejemplo de una tarjeta de historia de usuario.....	45
Figura 2.12: Calidad ISO/IEC.....	52
Figura 2.13: Características de funcionalidad.....	53
Figura 2.14: Calculo de puntos de función .....	55
Figura 2.15: Calculo de puntos de función .....	56
Figura 2.16: Características de fiabilidad.....	56
Figura 2.17: Características de usabilidad .....	58
Figura 2.18: Características de eficiencia .....	58
Figura 2.19: Características de mantenibilidad.....	59
Figura 2.20: Características de portabilidad.....	59
Figura 3.1: Caso de uso - Elemento involucrados .....	61
Figura 3.2: Caso de uso - Visión del Proyecto.....	63
Figura 3.3: Elementos involucrados .....	64
Figura 3.3: Cronograma del proyecto de producción.....	65
Figura 3.4: Caso de uso - Visualización de rutas .....	66
Figura 3.5: Caso de uso - Generar Denuncia .....	67
Figura 3.6: Modelo de datos.....	81

Figura 3.7: Pantalla Principal.....	84
Figura 3.8: Visualizacion del mapa de google maps enla aplicación .....	85
Figura 3.9: Busqueda de una ruta mediante zona .....	87
Figura 3.10: Obtención de rutas mediante una zona .....	87
Figura 3.11: Busqueda de ruta mediante el número de una línea de transporte .....	88
Figura 3.12: Visualización de una ruta mediante una zona .....	88
Figura 3.13: Visualización del módulo de denuncias .....	89

## ÍNDICE DE TABLAS

Tabla 1.1: Popularidad de sistemas operativos móviles en La Paz .....	7
Tabla 2.1: Popularidad de sistemas operativos móviles en La Paz .....	17
Tabla 2.2: Metodologías ágiles para desarrollo de software móvil .....	25
Tabla 2.3: Aportes de Diversas Metodologías Ágiles a Mobile - D .....	37
Tabla 3.1: Elementos Involucrados.....	62
Tabla 3.2: Roles y Responsabilidades.....	62
Tabla 3.3: Elementos involucrados .....	64
Tabla 3.4: Visualización de Rutas.....	66
Tabla 3.5: Generación de denuncias .....	67
Tabla 3.6: Historia de Usuario1 .....	68
Tabla 3.7: Historia de Usuario2 .....	69
Tabla 3.8: Historia de Usuario3 .....	69
Tabla 3.9: Historia de Usuario4.....	70
Tabla 3.10: Historia de Usuario5 .....	70
Tabla 3.12: Historia de Usuario7 .....	71
Tabla 3.13: Tarea de Ingeniería - Pantalla de Inicio .....	72
Tabla 3.14: Tarea de Ingeniería - Pantalla de Principal.....	72
Tabla 3.15: Tarea de Ingeniería - Integración con google play services .....	73
Tabla 3.16: Tarea de Ingeniería - Obtención del Api key de google.....	73
Tabla 3.17: Tarea de Ingeniería - Integración del mapa en la pantalla.....	74
Tabla 3.18: Tarea de Ingeniería - Diseño de la interfaz-Búsqueda por zonas .....	74
Tabla 3.19: Tarea de Ingeniería - Visualización de la ruta-Búsqueda por zonas.....	75
Tabla 3.20: Tarea de Ingeniería - Diseño de la interfaz-Búsqueda por línea.....	75
Tabla 3.21: Tarea de Ingeniería - Visualización de la ruta-Búsqueda por línea.....	76
Tabla 3.22: Tarea de Ingeniería - Diseño de la Interfaz-Búsqueda por sitio .....	76
Tabla 3.23: Tarea de Ingeniería - Visualización de la ruta -Búsqueda por sitio.....	77
Tabla 3.24: Tarea de Ingeniería - Diseño del módulo de denuncias.....	77

Tabla 3.25: Tarea Diseño del módulo de denuncias .....	77
Tabla 3.26: Planificación por Iteraciones.....	78
Tabla 4.1: Valores de complejidadsegun Press man .....	92
Tabla 4.2: Valores de Ajustes de complejidad.....	92
Tabla 4.3: Entradas de Usuario .....	93
Tabla 4.4: Salidas de Usuario .....	93
Tabla 4.5: Peticiones de Usuario .....	94
Tabla 4.6: Archivos de Usuario .....	94
Tabla 4.7: Calculo de Puntos de Funcion.....	94
Tabla 4.8: Valores de ajustes en Test de usuarios.....	97
Tabla 4.9: Factores de Eficiencia .....	98
Tabla 4.10: Factores de Ajuste de Mantenibilidad .....	99

### 1.1. Introducción

Actualmente nos encontramos inmersos dentro de los avances tecnológicos y esto se da a razón de la accesibilidad del internet, el cual está integrado en el uso de los teléfonos inteligentes (Smartphone), recientemente un estudio de la empresa IDC (International Data Corporation), dio a conocer que las ventas mundiales de los teléfonos inteligentes<sup>1</sup> superarán los mil millones de aparatos en el 2013, la reducción de costos y las ganancias en los mercados emergentes permiten el crecimiento de sus ventas, en Bolivia la Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes (ATT) sin dar cifras, informó que durante la gestión 2013 se incrementaron los usuarios de Internet móvil a través de los teléfonos inteligentes (Periódico Los Tiempos, 2013), el manejo de dichos teléfonos han superado las expectativas de alcance, ya que no solo la población joven está familiarizado con el uso de los teléfonos inteligentes, sino también niños y adultos y esto se da por la necesidad de integrar a las nuevas tendencias de comunicación en nuestra vida cotidiana.

---

<sup>1</sup> Teléfono Inteligente, es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor cantidad de almacenamiento de datos y su característica principal es el acceso a internet.

Existen varios sistemas operativos para dispositivos móviles (Android, iOS, Windows Phone, BlackBerry OS, Symbian OS, Firefox OS) pero el SO Android es el que más se destaca en cuanto a popularidad de los usuarios ya sea por su distribución gratuita o por ser de código abierto, es por el cual que se convierte en la mejor opción para desarrollar una aplicación móvil dirigida a la ciudadanía.

El presente documento presenta la implementación de un sistema de información con todas las rutas de minibuses de la ciudad de El Alto el cual podrán ser consultados en cualquier momento a través de la telefonía móvil, innovando el uso de las consultas sin el acceso a internet, y esto se realizara mediante el almacenamiento de información de rutas de las líneas de transporte y mostrar dicha información mediante el uso de mapas y puntos geo referenciales con el cual el usuario será capaz de evitar el uso de recorrido por tramos además de poder ir a una zona de la ciudad de El Alto sin conocer a exactitud dicha ciudad.

## **1.2.Antecedentes.**

Actualmente existen proyectos que brindan información del transporte público pero lamentablemente estos proyectos están dirigidos a otras ciudades (La Paz y Santa Cruz), en los cuales podemos mencionar:

### **1.2.1. Cruzero.**

Cruzero es una aplicación móvil para el transporte público local de Santa Cruz que fue creada por los ingenieros Jorge Espinoza y Charles Javier Ibáñez. El programa es gratuito y está disponible para teléfonos móviles Android en Google Playstore<sup>2</sup>. También cuenta con un portal web [www.cuzero.net](http://www.cuzero.net), ambas plataformas tienen la misma información, la cual contiene 5 módulos: calcular rutas, ruta de micros, taxi, lugares de interés y paradas.

Los creadores mencionaron que la aplicación ayuda a organizar su tiempo, explicó que "Cruzero" es una aplicación que busca entregar al usuario de una manera rápida y gráfica la solución a su necesidad diaria de conocer en qué modo de transporte puede llegar desde su origen (donde comienza su viaje) hasta su destino (donde termina su viaje). El usuario podrá obtener los beneficios de una planificación del tiempo que le ocupe transportarse y a la vez conocer la ubicación de diferentes puntos de interés dentro del radio urbano de la ciudad de Santa Cruz de la Sierra (Periódico El Deber, 2013).

### **1.2.2. TU GAMEA (Transporte Urbano del Gobierno Autónomo Municipal de El Alto).**

TU GAMEA es una aplicación móvil que fue diseñada por el Gobierno Autónomo Municipal de El Alto a cargo de la dirección de Desarrollo Organizacional y Tecnologías de la Información que como primer objetivo era el tratar de contrarrestar el recorrido mediante tramos del transporte público, indicando el recorrido completo de una línea

---

<sup>2</sup> Playstore, es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android.

perteneciente a algún sindicato, pero lamentablemente dicho proyecto se detuvo al no poseer una información precisa y actualizada de las rutas del transporte público de la ciudad de El Alto<sup>3</sup>.

### **1.2.3. SIG (Sistemas de Información Geo referencial vía telefonía móvil).**

En la Universidad Mayor de San Andrés, el informático Iván Ulo Aruquipa desarrollo como su Tesis de Grado un sistema de información geo referencial vía telefonía móvil para la Ciudad de La Paz, que en su primer objetivo es la localización exacta del usuario mediante la tecnología GPS e introduciendo datos del destino, la aplicación muestra las líneas de transporte público que debe abordar para llegar a su destino.

### **1.3.Planteamiento del Problema.**

El transporte público de El Alto funciona inadecuadamente. Si en algo los habitantes de una ciudad tan grande, diversa y compleja podrían coincidir sería en eso. La ola de protestas vecinales contra los choferes en los principales distritos, a lo largo del tiempo, refleja el malestar por el usuario ya que en la actualidad el transporte público se transformó en un verdadero caos y esto se da a razón de que el recorrido se lo realiza por tramos cada vez más cortos, incomodidad e inseguridad creciente en vehículos cada vez más viejos y peligrosos, lentitud exasperante, no sólo en horas pico y no sólo en vías troncales, y conductores estresados al límite, es por el cual que el usuario del transporte público se acostumbró a vivir el día a día con ello.

---

<sup>3</sup> Fuente: Unidad de Tecnologías de la Información, Área de Desarrollo de Sistemas

El recorrido por tramos es una estrategia creada por el transporte público con el fin de obtener mejores ganancias y esto es permitido por el ciudadano a razón del desconocimiento de la ruta completa (parada a parada) para luego transformarlo en un problema del ciudadano que día a día se traslada de un lugar a otro y por esta razón existe muchas pérdidas tanto económicas como de tiempo.

### **1.3.1. Formulación del Problema.**

¿Cómo evitar el recorrido por tramos que pretenden imponer los operadores del transporte público e integrar los conocimientos de las rutas completas al ciudadano alteño?

### **1.3.2. Problemas Secundarios.**

- Incumplimiento en la llegada a la parada por parte de los miembros del transporte público.
- Carencia de Información de las rutas completas del transporte público por parte del usuario.
- Existencia de líneas y rutas fantasma.
- Descontrol a los miembros del transporte público por parte de las autoridades.
- Pérdidas económicas de parte del usuario del transporte público.
- Pérdidas de tiempo del usuario.

## **1.4.Planteamiento de Objetivos.**

### **1.4.1. Objetivo General.**

Controlar las rutas del transporte público vehicular a través del desarrollo de un sistema de información vía telefonía móvil, con el fin de facilitar información de los recorridos completos del transporte público, con el fin de ayudar a controlar el recorrido por tramos del transporte público vehicular de la ciudad de El Alto, evitando de manera eficaz y simple el Trameaje.

### **1.4.2. Objetivos Específicos.**

- Mostrar la información almacenada de las rutas del transporte público en mapas indicando los sitios más conocidos del recorrido.
- Implementar métodos de búsqueda de rutas, ya sea por línea de sindicato o por la zona de algún distrito.
- Desarrollar un módulo de denuncias por parte del usuario.

## **1.5.Justificación.**

### **1.5.1. Técnica.**

La adquisición de teléfonos inteligentes en la población ha superado todo tipo de expectativas, ya que no solo la población joven está familiarizada con el uso de dichos teléfonos, sino también niños y adultos y esto se da a razón del fácil e intuitivo manejo que estos teléfonos poseen (Periódico Los Tiempos, 2013).

De enero a marzo de 2013, el número de teléfonos con acceso a internet llegó a un crecimiento del 15% respecto a diciembre de 2012, cuando se registraron 458.691 teléfonos en la ciudad de La Paz, la información es proporcionado por la Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes (Periódico La Razón 2013).

Sistema Operativo	SmartPhone
Android	74,97%
IOS	19,32%
BlackBerry	1,47%
Windows Phone	1,13%

**Tabla 1.1. Popularidad de sistemas operativos móviles en La Paz.**

**Fuente: Colonización Mobile de Latinoamérica 2013.**

Android es un sistema operativo móvil, es el que más se destaca en cuanto a popularidad de los usuarios, ya sea por su distribución gratuita o por ser de código abierto (Colonización Mobile de Latinoamérica 2013), es por el cual que se convierte en la mejor opción para desarrollar el sistema de información a través de telefonía móvil dirigida a la ciudadanía Alteña.

### **1.5.2. Social.**

Sera de gran beneficio para toda la población alteña ya que el mismo contara con la información correcta y detallada del recorrido completo de una ruta y con esto tratar de evitar más engaños y de esa manera hacer llegar su molestia en caso del incumplimiento por parte del encargado del transporte público vehicular.

### **1.5.3. Práctica.**

El ciudadano se traslada día a día de un lugar a otro atravesando por muchas molestias que nos brinda el transporte público y por el desconocimiento de rutas somos culpables de estrés, es por el cual que el sistema de información trata de evitar dichos malestares brindándoles la información correcta y precisa.

### **1.5.4. Económica.**

Al integrar el sistema de información con una aplicación android se reduce de gran manera los gastos, se evita comprar un software y una licencia de funcionamiento, además que la distribución de la aplicación se lo realiza mediante la plataforma de distribución digital de android (Play Store) de forma gratuita.

### **1.6. Alcance.**

En primer lugar la investigación no pretende menospreciar el trabajo que realizan las personas encargadas del transporte público, como tampoco pretende sancionar a los miembros del transporte público, el proyecto trata de hacer notar las molestias del usuario hacia el transporte público y también brindar el conocimiento de todas las rutas que existe en la ciudad de El Alto y con ello tratar de evitar el recorrido por tramos.

El proyecto se desarrollará con una investigación exhaustiva acerca de la cantidad de sindicatos y líneas que existe en la ciudad de El Alto además de almacenar las rutas completas en la base de datos y mostrar dichas rutas en mapas para su mejor entendimiento y proporcionar la opción de denuncias en caso de que el usuario así lo disponga.

## **CAPITULO II**

### **MARCO TEÓRICO.**

#### **2.1.Telefonía Móvil.**

La telefonía móvil, también llamada telefonía celular, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red.

El teléfono móvil es un dispositivo inalámbrico electrónico que permite tener acceso a la red de telefonía celular o móvil. Se denomina celular debido a las antenas repetidoras que conforman la red, su principal característica es su portabilidad, que permite comunicarse desde casi cualquier lugar. Aunque su principal función es la comunicación de voz, como el teléfono convencional, su rápido desarrollo ha incorporado otras funciones como son cámara fotográfica, agenda, acceso a Internet, reproducción de vídeo y sonido e incluso GPS (Salazar Gerrero A.).

## 2.2.Sistemas operativos para teléfonos móviles.

Los sistemas operativos (SO) para teléfonos móviles se vuelven cada día más importantes pues la tecnología avanza y en materia de comunicaciones aún más, la telefonía celular cada vez se convierte más en una parte importante de nuestras vidas, y en una sociedad que exige más y más, es importante diseñar sistemas que soporten las aplicaciones que se demandan, que sean fluidos, fáciles, accesibles y hasta divertidos.

Es por eso que las compañías móviles han desarrollado una competencia bastante reñida en cuanto al desarrollo de SO se refiere, desde los inicios en los años 90 con las versiones de EPOC 32<sup>6</sup> para PDA's<sup>7</sup> hasta los más avanzados y sofisticados como Android, IOS, BlachBerry que además de ser eficientes y estables son multiplataforma, lo que hace que cualquier persona tenga acceso a ellos desde un celular básico hasta un Smartphone.

Además cada vez más usuarios les agradan la idea de manipular y estilizar sus equipos y es lo que los nuevos SO están ofreciendo y esto implica más retos de programación e incluso en el hardware.

Un sistema operativo móvil o SO móvil es un sistema operativo que controla un dispositivo móvil al igual que las Computadoras más grandes que utilizan Windows, Linux o Mac OS entre otros.

---

<sup>6</sup> EPOC 32, sistema operativo multitarea para PDA.

<sup>7</sup> PDA, es un dispositivo pequeño que combina un ordenador, teléfono/fax, Internet y conexiones de red.

Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos (Pedroso Petrazzini, G.).

### 2.2.1. Componentes de los SO móviles.

Al igual que los sistemas operativos presentes en los componentes informáticos más grandes, un sistema operativo móvil también se encuentra compuesto por varias capas.

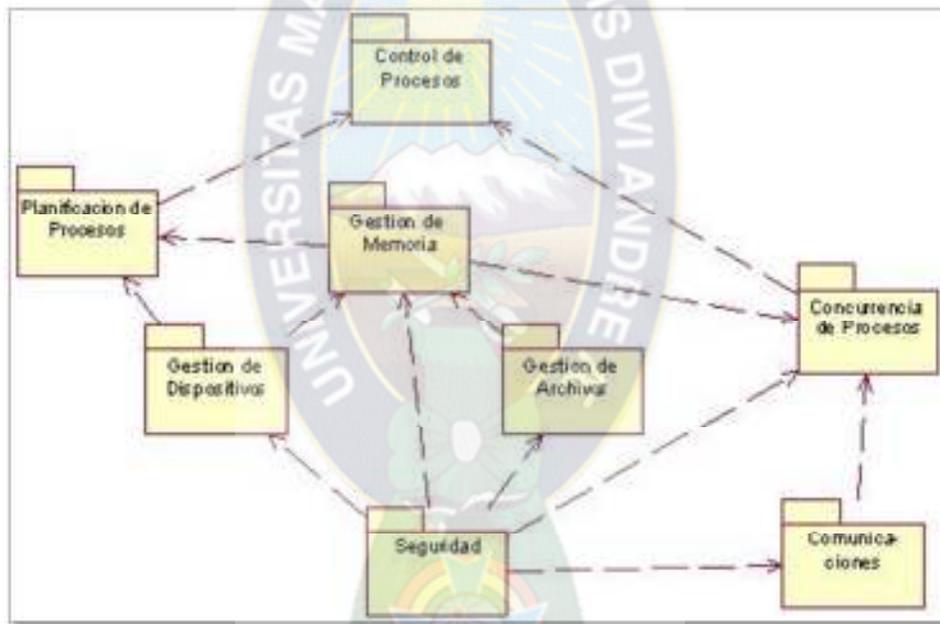


Figura 2.1. Componentes de los sistemas operativos móviles.

Fuente: Sistemas Operativos en Dispositivos Móviles

### **2.2.1.1. Kernel.**

Una de las más importantes y esenciales piezas que componen cualquier sistema operativo, sea el de nuestro móvil, o el de la PC, es el denominado núcleo o Kernel, el cual es la capa de software que permite el acceso a los diferentes elementos de hardware que conforman nuestro móvil.

También es el encargado de brindar diferentes servicios a las capas superiores como los controladores de hardware, gestión de procesos, sistemas de archivos, además del acceso administración de la memoria del sistema.

Los sistemas operativos para móviles pueden basarse en núcleos Linux, tal como lo hace Android, o hasta inclusive IOS, el SO del iPhone que utiliza un kernel heredado de Unix<sup>8</sup>.

En el caso de Android presenta la particularidad de contar con un motor Java<sup>9</sup> en el desarrollo de sus núcleos.

### **2.2.1.2. Middleware.**

Esta capa es el conjunto de módulos que permite que las aplicaciones diseñadas y escritas para tales plataformas puedan ser ejecutadas.

---

<sup>9</sup> Unix, es un sistema operativo multitarea y es multiusuario que trabaja y funciona de manera similar a linux.

<sup>8</sup> Java, es un lenguaje de programación y una plataforma informática comercializada por Sun Microsystems.

Su funcionamiento es totalmente transparente para el usuario, no debiendo realizar ninguna acción ni configurar alguna para su correcto desenvolvimiento, entre estos tenemos:

- Motor de mensajería
- Intérpretes de páginas web/WAP
- Motor de comunicaciones
- Codecs multimedia
- Gestión del dispositivo
- Seguridad

### **2.2.1.3. Entorno de ejecución de aplicaciones.**

Esta capa provee de todos los elementos necesarios para la creación y desarrollo de software a los programadores, es decir contiene elementos que serán de gran ayuda a los mismos, en el momento de escribir aplicaciones compatibles con ese sistema operativo.

Entre los servicios que los programadores pueden encontrar, se destacan un gestor de aplicaciones y una serie de interfaces programables (APIs<sup>10</sup>) abiertas.

---

<sup>10</sup> Api, es un conjunto de procedimientos que cumplen muchas funciones con el fin de ser utilizadas por otro software.

#### 2.2.1.4. Interfaz de usuario.

La interfaz de usuario es el elemento del teléfono que usualmente utilizamos para interactuar con el aparato. Sin esta capa no sería posible utilizar nuestro dispositivo, ya que la misma presenta todos los elementos necesarios para facilitar cualquier tipo de tarea que deseemos realizar en nuestra terminal.

#### 2.2.2. SO en el mercado.

A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. La cuota de mercado de sistemas operativos móviles en el tercer trimestre de 2012 era el siguiente:

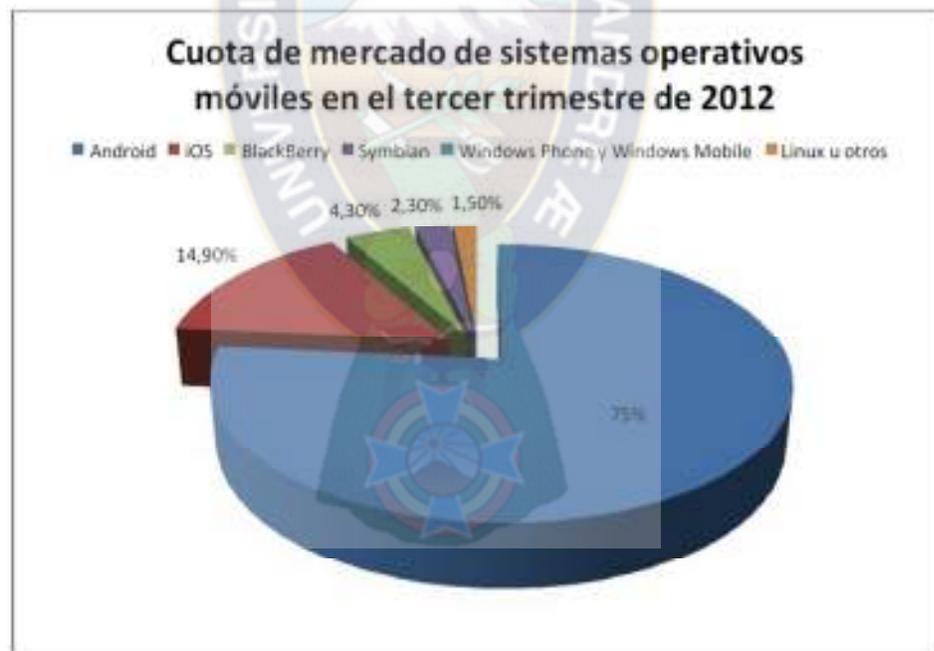


Figura 2.2. Popularidad de SO móviles.

Fuente: Monografía, Sistemas Operativos en Dispositivos Móviles.

Android tiene la mayor cuota, desde enero 2011, con más de la mitad del mercado, experimentó un creciente aumento y en sólo dos años (2009 a comienzos de 2011) ha pasado a ser el SO móvil más utilizado (Pedroso Petrazzini, G.).

## **2.3. Selección de la herramienta de desarrollo.**

### **2.3.1. SO Android.**

El año 2003 un grupo de visionarios, Andy Rubin, Richter Miner, Chris White y Nick Sears, daban comienzo a la compañía Android Inc, que comenzó a fabricar las primeras piezas del androide verde. En un comienzo esta compañía tenía como objetivo la creación de software para teléfonos móviles, nunca se imaginaron que terminarían fabricando el más grande sistema operativo para teléfonos inteligentes.

La compañía Android Inc. estuvo prácticamente 2 años trabajando sin gran reconocimiento, hasta que fue adquirida por Google en julio del año 2005. Desde entonces siempre que se habla de este sistema operativo se hace referencia a dicha compañía y en la actualidad es manejado por Open Handset Alliance cuya organización es liderada por Google<sup>11</sup>.

---

<sup>11</sup> <https://sites.google.com/site/swcuc3m/>

El proyecto se centrará en una solución del tipo nativa, la cual se desarrolla en la plataforma y sistema operativo móvil basado en Linux de Google, Android.

Su definición, Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos.

Una de las razones para adoptar Android como el sistema operativo para la implementación de este proyecto y en si como plataforma principal de desarrollo, se basa en que es una plataforma de código abierto.

La forma de programación en Android es mucha más intuitiva que otras plataformas como la desarrollada por Apple, iOS con su lenguaje particular.

Además, existen varios motivos para elegir implementar el sistema en esta plataforma:

### **2.3.2. Aprendizaje.**

Al obtener el SDK de Android, se requiere para implementar código (en esta plataforma) conocer el lenguaje de programación Java. Este lenguaje de programación es muy intuitivo, orientado a objetos y de vasta popularidad a nivel de implementación.

### 2.3.3. Relevante.

De enero a marzo de 2013, el número de teléfonos con acceso a internet llegó a un crecimiento del 15% respecto a diciembre de 2012, cuando se registraron 458.691 teléfonos en la ciudad de La Paz, la información es proporcionado por la Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes (Periódico La Razón 2013).

Sistema Operativo	SmartPhone
Android	74,97%
iOS	19,32%
BlackBerry	1,47%
Windows Phone	1,13%

**Tabla 2.1. Popularidad de sistemas operativos móviles en La Paz.**  
Fuente: Colonización Mobile de Latinoamérica 2013.

Android es un sistema operativo móvil, es el que más se destaca en cuanto a popularidad de los usuarios (Colonización Mobile de Latinoamérica 2013)

### 2.3.4. Código Abierto.

El Open Source es una tendencia que ha marcado el software tradicional desde el punto que no se necesita de licencias ni de contratos especiales por adopción de uso. Las plataformas móviles apuntan a ser un negocio de alta proyección, lo cual garantiza la permanencia en el mercado y su uso comercial tiene un factor importante al momento de licencias, contratos, y prestaciones de servicio.

### **2.3.5. Introducción a la plataforma Android.**

Android es una plataforma de código libre para smartphones, tabletas y dispositivos móviles en general, que se complementa con un sistema operativo, un sistema en tiempo de ejecución basado en Java, un conjunto de librerías de medio y bajo nivel y un conjunto de aplicaciones, desarrolladas en Java, destinadas al usuario final.

La plataforma se distribuye bajo una licencia libre del tipo Apache<sup>12</sup>, que permite integración con soluciones del tipo propietarias. La Open Handset Alliance, un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en la telefonía móvil, fue la encargada de dar paso a este Sistema Operativo para móviles. Además, de ser un compromiso para su comercialización en diferentes estándares de hardware de dichas compañías. Su principal desarrollo recae sobre Google una vez realizada la compra del Sistema Operativo en el 2005.

Entre estas empresas se encuentran grandes del negocio de las operadoras de telefonía como Telefónica, Vodafone, T-Mobile, fabricantes de móviles (Motorola, Samsung, Acer, LG, HTC...) o relacionadas al Hardware (nVidia, Intel o Texas Instruments).

#### **2.3.5.1.Arquitectura.**

Como se denota en la figura 2.4, la Arquitectura de esta plataforma se basa en 4 niveles o capas<sup>13</sup>:

---

<sup>12</sup> Apache, es un servidor web HTTP de código abierto.

<sup>13</sup> <https://conectica.com.mx>

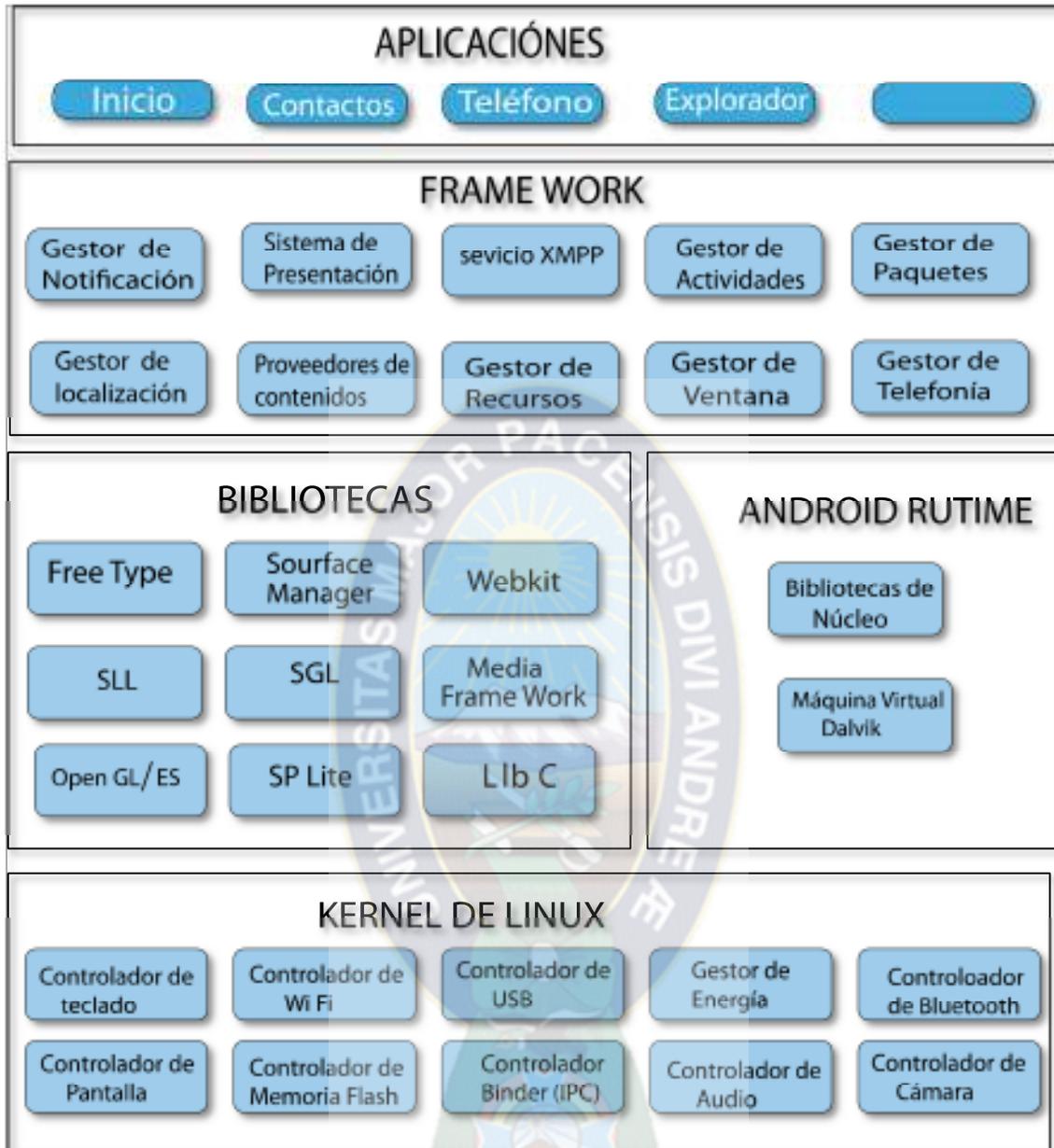


Figura 2.3.: Arquitectura de capas de Android

Fuente: [conectica.com.mx](http://conectica.com.mx)

- **Capa de aplicaciones.**

Constituyen un conjunto de aplicaciones que corren en el dispositivo como base para otras o para el uso normal del teléfono. Estas son: teléfono, cliente de email, programa de envío de SMS, calendario, mapas, navegador, contactos, etc. Y estas pueden ser usadas por otras aplicaciones. Dentro de esta capa se desarrollará la aplicación.

- **Framework de aplicaciones.**

Se encuentra separado por varios subsistemas y tienen un número de componentes específicos para cada módulo, por ejemplo el del administrador del teléfono, se encarga de la gestión del dispositivo como hardware. Se puede desarrollar partiendo de la característica de reúso de componentes de cada aplicación, de esta manera se orienta la arquitectura para facilitar la reutilización de componentes para las aplicaciones.

Estas normas de reúso se rigen bajo controles de seguridad proporcionados por el framework. También en esta capa se incluye el sistema de vistas que manejan la UI37 de las aplicaciones, con capacidades como el visualizar elementos HTML o el render de mapas en la vista de la aplicación.

- **Capa de bibliotecas de bajo nivel.**

Escritas en C y C++, sirven para persistencia de datos, gestión de gráficos 3D, navegadores web embebidos. Entre estas librerías se encuentran elementos como SQLite, OpenGL, Webkit, etc.

- **Kernel Linux.**

Sirve como una columna importante y fundamental de software la cual se encarga del sistema de gestión de drivers, seguridad, comunicaciones y algunas funciones esenciales del sistema.

El siguiente diagrama contiene el diagrama de arquitectura por capas del que se compone la plataforma Android.

### **2.3.5.2.Desarrollo en la plataforma Android.**

Android cuenta con una comunidad, la cual presta foros y algunos recursos como tutoriales de manera amigable para hacer el desarrollo de aplicaciones en esta plataforma una tarea muy sencilla y de esta manera generar un mayor grupo de aplicaciones generadas para esta tecnología. Estas prestaciones se las pueden descargar directamente y gratuitamente de su página web <https://developers.google.com>.

### **2.3.5.3.Entorno de desarrollo.**

Para el entorno de desarrollo generado para extender código en Android se cuenta con un plugin o adaptador para un IDE muy conocido llamado Eclipse.

Este plugin adapta y extiende funcionalidad para la creación, depurado y otras funciones mencionadas:

---

<sup>14</sup> Plugin, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

- Asistentes para la creación rápida de aplicaciones Android.
- Complemento para crear interfaces gráficas permitiendo el desarrollo de componentes visuales.
- Emulador personalizado de Android, el cual funciona sobre una terminal y versión móvil virtual personalizable.
- La interactividad e incrementos en funciones en herramientas de Android como la captura de pantalla, depuración en tiempo real, redirección de puertos, ver información de hilos de ejecución en proceso, etc.
- Editor de gráficos, XML los cuales proporcionan una mejor comprensión para el desarrollo.



**Figura 2.4.: Emulador de android.**

**Fuente: Elaboración propia**

## 2.4.Aspectos metodológicos.

El negocio de las aplicaciones móviles día tras día se encuentra en un proceso de cambio y desarrollo constante. Con ello crece su demanda, volviéndose más rentable, de la manera que se hace con el software convencional. Se hace hincapié en la forma de optimizar los recursos con los que el desarrollador llevará al éxito el proceso de desarrollo. Una forma de optimizar los recursos es contando con una metodología que conduzca al desarrollo oportuno, eficaz y eficiente de aplicaciones. Sin olvidar, la relación de estas con su entorno y características propias de las aplicaciones móviles. De esta manera, se tratara de mitigar el uso inadecuado de metodologías tradicionales por parte del desarrollador de aplicaciones móviles. A pesar de que los desarrolladores conocen que el software móvil debe satisfacer requerimientos y restricciones especiales, utilizan metodologías tradicionales que resultan ser incompatibles con este tipo de tecnología.

El proceso de desarrollo de aplicaciones móviles afronta también la segmentación del mercado y demanda de estas en una variada gama de plataformas incompatibles, varias de estas son: Symbian, Windows Phone, iPhone SDK, Android o Java, entre otras. Todo esto hace que el proceso de desarrollo para plataformas móviles sea más complejo.

En las próximas secciones se adecuará el tema para la selección de la metodología ágil de desarrollo de software móvil. Para ello, primero se abordará brevemente el tema de las metodologías ágiles, y después se explicará las características propias y problemas a la hora de tratar dispositivos móviles con sus respectivos entornos.

## 2.5. Metodologías de desarrollo.

Tras una reunión llevada a cabo en febrero del 2001, nace el concepto de metodologías de desarrollo ágil aplicado al desarrollo software. La principal meta era conseguir que los proyectos de software no se dilaten y exista una gestión de cambios de tal manera que sean accesibles a cambios durante el desarrollo. Esto en respuesta a las metodologías de desarrollo de aplicaciones tradicionales, definidos por su carácter rígido e intolerantes en el sentido de la documentación.

La aparición de estas metodologías “ágiles” aparentemente fue un reflejo del alto número de proyectos que terminan en el fracaso o se toman más tiempo del estimado y la calidad no mejorada ni explotada por parte de proyectos de desarrollo de software. Una vez asentados estos conceptos se establece The Agile Alliance<sup>14</sup>, propuesta a extender la nueva noción de Desarrollo de software “Ágil” destacando la creación de conceptos contenidos en el Manifiesto Ágil, conceptos y valores primordiales como:

- “El individuo y las interacciones del equipo de desarrollo están por encima del proceso y las herramientas. Construir un buen equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades”.
- “Desarrollar software que funciona más que conseguir buena documentación. No producir documentos a menos que sean necesarios de una forma inmediata. Si el software no funciona, los documentos no valen de nada”.

---

<sup>14</sup> The Agile Alliance, es una organización sin fines de lucro con membresía global, comprometida con el avance principios y prácticas de desarrollo ágil.

- “La colaboración con el cliente es más importante que la negociación de contratos. Tiene que haber una interacción constante entre el cliente y el equipo de desarrollo”.
- “La respuesta ante el cambio es más importante que el seguimiento de un plan. La planificación no debe ser estricta sino flexible y abierta, la habilidad de responder a los cambios que surjan determina el éxito o fracaso del proyecto”.

## 2.6. Metodologías ágiles para el desarrollo de software móvil.

Aunque existen un amplio número de metodologías de desarrollo de software ágiles, ninguna se ha centrado en el entorno y en la caracterización que el desarrollo de software para móviles necesita. Pero lo que sí se puede apreciar es que dichas metodologías coincidan con ciertos valores o criterios apreciables al desarrollo dirigido para plataformas móviles.

En forma de un análisis comparativo, se presenta la igualdad de criterios de los métodos ágiles sobre el desarrollo de software para móviles partiendo de una motivación lógica, que se muestra en la Tabla 2.1.

Características ágiles	Motivación lógica	En el caso del desarrollo para plataformas móviles
Alta volatilidad del entorno	Debido a la alta frecuencia en el cambio que sufren los requerimientos, tendremos menos necesidad de diseño y planificación inicial y mayor necesidad de desarrollos incrementales e iterativos.	Alta incertidumbre, entornos dinámicos, cientos de nuevos terminales cada año

**Tabla 2.2.: Metodologías ágiles para desarrollo de software móvil - análisis comparativo.**

**Fuente: Desarrollo de una aplicación móvil para el recuento de stock**

Aunque la forma de desarrollar aplicaciones para un entorno o plataformas móviles no está exenta de los problemas presentados con el desarrollo de aplicaciones convencionales, se ha estimado que la corta duración del desarrollo que las plataformas móviles exigen es relativamente corta. Esto interviene a la hora de optar por una metodología específica de desarrollo.

El mayor beneficio de una metodología específica para el desarrollo es sin duda la organización de tareas, sistematización del procedimiento y sobretodo un modelo estático y predictivo que administre las tareas y además, nos informe intuitivamente la situación del proyecto en cualquier período de tiempo en que nos lo preguntemos. A continuación, vamos a analizar una serie de metodologías aplicables, bajo los criterios antes mencionados, para el desarrollo de aplicaciones móviles:

### **2.6.1. Desarrollo ágil: Programación Extrema (XP).**

Formulado y creado inicialmente por Kent Beck es el más destacado de los procedimientos ágiles de desarrollo de software. La programación extrema se diferencia de metodologías tradicionales en que se encarga de la adaptabilidad y la previsión a la gestión de cambios y a los temas relacionados a la metodología.

Adeptos a esta metodología aseguran que los cambios de requerimientos no es algo fuera de lo común, es más, siempre han sido considerados y algunas veces hasta deseados a diferencia de otras prácticas. Se pretende que una gestión de cambios activa en cualquier punto del proyecto representa y simboliza una aproximación más realista que

definir todos los requerimientos al comienzo y sobre invertir energías en tratar de controlar la gestión de cambios en dichos requerimientos.

Esta metodología tiene cinco valores mencionados a continuación:

- Simplicidad.

En este valor se basa toda la programación extrema, se agiliza el desarrollo y se facilita el mantenimiento al tratarse de una implementación sencilla. Aunque se trata en esta metodología de cuidar la simplicidad, las sucesivas modificaciones del código y de su diseño hacen que aumente la complejidad.

Refactorizando el código, se mantiene este principio mientras el código crece con las múltiples modificaciones. Asimismo, la simplicidad se mantiene en la documentación, juntamente con el desarrollo del código se comenta en justa medida y sin sobre esforzarse, manejando estándares de tal manera que se cuide la auto documentación<sup>8</sup> de dicho código.

- Comunicación.

La programación además de reflejar la simplicidad muestra la comunicación que este tenga. La comunicación es primordial y mejor mostrada cuando el código es simple. Si el código es complejo, debe esforzarse y tratar a toda costa de hacerlo inteligible.

Las pruebas unitarias de una u otra forma transmiten información acerca del diseño de sus clases y de ciertos métodos al mostrar formaciones concretas de cómo utilizar la funcionalidad. Gracias a la programación en parejas, la comunicación se muestra constantemente entre desarrolladores. La comunicación con el cliente es de

manera directa ya que el equipo de desarrollo integra al cliente también gracias a la filosofía de la metodología.

- Retroalimentación (feedback).

Mientras el cliente, según la filosofía de la metodología, esté al tanto del desarrollo dentro del proyecto, su opinión y su perspectiva del mismo es considerado en tiempo real. Las iteraciones cortas hacen que se muestren tanto los resultados y su proyección en el proyecto en el instante que está ocurriendo.

El desarrollo y planificación de pruebas tienen embebido este valor a través de su especificación, por ejemplo, las pruebas y análisis de resultados muestran la calidad y sobreestima del código dentro de la solución.

- Coraje o valentía.

Este valor es fundamental para reconstruir código cuando sea necesario o cuando el requerimiento esbozado por el usuario amerite su replanteamiento. Con esto, se evita empantanarse en el diseño y a su vez estimar demasiado tiempo en su dedicación para la implementación. En el diseño también es fundamental a la hora de planificarlo para satisfacer las necesidades a corto plazo para estimar el tiempo y no centrarse en el trabajo a futuro.

La valentía también está asociada en la moral del desarrollador para desechar el código que no sirve o a su vez, ha perdido a través del tiempo su funcionalidad. Este valor se relaciona a pesar de cuanto esfuerzo, tiempo y recursos se hayan requerido para su implementación.

- Respeto.

Valor fundamental y de carácter moral. Gracias a este existe un ambiente de respeto y armonía entre desarrolladores, como la implementación es colaborada se respeta el esfuerzo y los recursos usados por un desarrollador en un momento específico. Además, no se pueden hacer cambios, adiciones, modificaciones ni eliminación de código que implique que las pruebas fallen o que demore más trabajo en su construcción.

### **2.6.2. Desarrollo ágil: Scrum.**

Más que una metodología, es un marco de trabajo para el desarrollo de software, bajo un proceso iterativo e incremental usado para ambientes de desarrollo basado en desarrollo ágil de software.

Orientado inicialmente en gestión de procesos de desarrollo de software, puede ser usado en equipos de mantenimiento de software.

Scrum define un conjunto de roles y prácticas y se toma como un punto de partida para definir procesos de desarrollo, en los cuales, la estima de los recursos humanos es primordial. Scrum maneja varios roles dentro de la metodología como el Scrum Master, el Product Owner y en general, se hace referencia a el Team.

El Scrum Master es el encargado de la dirección del proyecto; mientras que el Product Owner se encarga de la gestión de los stakeholders<sup>15</sup> o interesados del proyecto, tanto internos como externos, y el equipo de desarrollo encargado de la implementación.

Sus períodos iterativos denominados sprint comprenden un período entre una y cuatro semanas. Están conformados y designados por el equipo de desarrollo, este equipo definirá un entregable que será un incremento de software potencialmente utilizable.

El Product Backlog es un conjunto de requerimientos de alto nivel que poseen una prioridad definida y establecen los parámetros fijos y precisos enfocándose en el esfuerzo que se realizará. Este Product Backlog definirá las características de cada sprint y su realización.

El procedimiento se realiza desde el punto de vista del Product Owner que identifica los elementos que contendrá el Product Backlog y se los define en la reunión de Sprint Planning. El Team analiza la cantidad de esfuerzo y se plantea si se puede o no realizar dicho trabajo en el siguiente Sprint. Durante el desarrollo de Sprint en ejecución, no se aceptan cambios dentro del tiempo que este durara, dando así la certeza de que la gestión de cambios se encuentra congelada durante el período de tiempo determinado que durará el sprint.

---

<sup>15</sup> stakeholders, son todos los actores interesados directa o indirectamente en una empresa.

Se permiten los equipos auto-organizados dentro del desarrollo, considerando la comunicación verbal de los miembros y demás disciplinas dentro del proyecto.

Una ventaja por sobre otras metodologías y procesos de desarrollo de software es que Scrum es muy fácil de aprender y requiere poco esfuerzo tanto de interesados en el proyecto como desarrolladores para utilizar inicialmente.

La desventaja mayor indicada es que, aunque su gestión de cambios sea actualizada en gran medida y propuesta directamente por el cliente, no es tan flexible como otras metodologías ágiles de desarrollo de software.

### **2.6.3. Mobile-d.**

El método Mobile-D se desarrolló junto con un proyecto finlandés en el 2004. Fue realizado, principalmente, por investigadores de la VTT <sup>16</sup> y a pesar de que es un método antiguo, sigue en vigor y se está utilizando en proyectos de éxito y está basado en técnicas que funcionan.

El objetivo es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en menos de diez semanas.

---

<sup>16</sup> VTT, Centro de investigación aplicada finlandés.

Las fases son:



**Figura 2.5. Fases de la metodología para desarrollo móvil Mobile-D.**  
Fuente: Tesis - Aplicación móvil para el recuento de stocks

- **Exploración.** Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases.
- **Inicialización.** Se preparan e identifican todos los recursos necesarios. Se establece el entorno técnico.
- **Producción o fase de producto.** Se repiten iterativamente las sub fases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del test driven development para conseguir la mayor calidad.

- **Fase de estabilización.** Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente.
- **Fase de pruebas y reparación.** Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente.

Una vez acabadas todas las fases, deberíamos tener una aplicación publicable y entregable al cliente.

Mobile-D se basa en Extreme Programming<sup>16</sup> (prácticas), Metodologías de Cristal (escalabilidad) y Rational Unified Process (cobertura). La documentación completa de esta metodología se puede encontrar disponible en el sitio web del proyecto.



---

<sup>16</sup> [www.extremeprogramming.org/](http://www.extremeprogramming.org/)

#### 2.6.4. Características y requerimientos del entorno móvil.

- **Canal radio:** consideraciones tales como la disponibilidad, las desconexiones, la variabilidad del ancho de banda, la heterogeneidad de redes o los riesgos de seguridad han de tenerse especialmente en cuenta en este entorno de comunicaciones móviles.
- **Movilidad:** aquí influyen consideraciones como la migración de direcciones, alta latencia debido a cambio de estación base o la gestión de la información dependiente de localización. Sobre esta última, de hecho, se pueden implementar un sinfín de aplicaciones, pero la información de contexto asociada resulta muchas veces incompleta y varía frecuentemente.
- **Portabilidad:** la característica portabilidad de los dispositivos terminales implica una serie de limitaciones físicas directamente relacionadas con el factor de forma de los mismos, como el tamaño de las pantallas (algo que ha variado sustancialmente con la popularización de las pantallas táctiles), o del teclado, limitando también el número de teclas y su disposición.
- **Fragmentación de la industria:** la existencia de una considerable variedad de estándares, protocolos y tecnologías de red diferentes añaden complejidad al escenario del desarrollo móvil.
- **Capacidades limitadas de los terminales:** aquí incluimos factores como la baja potencia de cálculo o gráfica, los riesgos en la integridad de datos, las interfaces de usuario poco funcionales en muchos aspectos, la baja capacidad de

almacenamiento, la duración de las baterías o la dificultad para el uso de periféricos en movilidad.

Factores todos que, por otro lado, están evolucionando en la dirección de la convergencia de los ultra portátiles (netbooks) con los dispositivos inteligentes (smartphones) constituyendo cada vez menos un elemento diferencial.

- **Diseño:** desde el punto de vista del desarrollo, el diseño multitarea y la interrupción de tareas es clave para el éxito de las aplicaciones de escritorio; pero la oportunidad y frecuencia de éstas es mucho mayor que en el software tradicional, debido al entorno móvil que manejan, complicándose todavía más debido a la limitación de estos dispositivos.
- **Usabilidad:** las necesidades específicas de amplios y variados grupos de usuarios, combinados con la diversidad de plataformas tecnológicas y dispositivos, hacen que el diseño para todos se convierta en un requisito que genera una complejidad creciente difícil de acotar.
- **Time-to-market:** en un sector con un dinamismo propio, dentro de una industria en pleno cambio, los requisitos que se imponen en términos de tiempo de lanzamiento son muy estrictos y añaden no poca dificultad en la gestión de los procesos de desarrollo.

De esta manera, se deja en claro que el desarrollo y, principalmente, el diseño es mucho más complejo que el desarrollo tradicional contemplado en otros proyectos de desarrollo. Es por ello que, se ha tenido que deliberar sobre el planteo de otras metodologías de desarrollo de software para el tratamiento de sistemas móviles, como son el uso de metodologías ágiles. Estas últimas se han acercado más al desarrollo para

plataformas móviles, aunque las propiedades y características mencionadas anteriormente propias de estos dispositivos y la naturaleza de las redes de telefonía móvil requiere que se hagan todavía ajustes sobre las actuales metodologías ágiles.

### **2.6.5. Selección de la metodología.**

#### **2.6.5.1.Motivación.**

Para el desarrollo de una aplicación móvil para brindar información del control de rutas de la Ciudad de El Alto, se debe emplear una metodología estable, además de mencionar una metodología ágil donde se esté pensado un desarrollo para grupos no mayores a 10 desarrolladores colaborando en un mismo entorno físico. Es por ello que se ha escogido trabajar con la metodología ágil para aplicaciones y desarrollos móviles “Mobile-D”.

Dicha metodología propone que si “se trabajan el ciclo de desarrollo propuesto, los proyectos deberían finalizar con el lanzamiento de productos completamente funcionales en menos de diez semanas”.

#### **2.6.5.2.Principios básicos Metodología Mobile-D.**

La metodología creada inicialmente para la industria directa, con plataformas móviles se ha sustentado en otras metodologías ya conocidas. Además, estas tienen peso a la hora de la planificación del desarrollo e ingeniería de software: extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP).

En lo que se refiere a las prácticas de desarrollo de la metodología se han utilizado los principios de la Programación Extrema, así como se suministra un input valioso de la escalabilidad de los métodos gracias al aporte de las metodologías Crystal. Todo esto basándose en el diseño justo del ciclo de vida que ha proporcionado la metodología Rational Unified Process.

Programación Extrema	Metodología Crystal	Rational Unified Process
Aporte de las prácticas de desarrollo	Aporte Escalabilidad, gestión de cambios	Aporte del diseño del ciclo de vida

**Tabla 2.3.: Aportes de diversas Metodologías Agiles a Mobile-D**  
**Fuente: Elaboración propia.**

El ciclo de vida del proyecto dentro de la metodología Mobile-D se divide en cinco fases o etapas: exploración, inicialización, producción, estabilización, y finalmente las pruebas del sistema.

La metodología pretende fomentar el desarrollo iterativo lo más rápido posible para obtener un desarrollo más productivo, es por ello, que la fases (excepto la fase de exploración) tienen tres etapas de desarrollo cambiante destinados a la planificación, trabajo y liberación, la metodología sugiere un día para cada tarea. Se añadirá tiempo además para las tareas adicionales que se requieran para culminar el desarrollo de cada etapa.

El siguiente grafico muestra el ciclo de desarrollo de la metodología para el desarrollo de aplicaciones móviles Mobile-D.



Figura 2.6.: Ciclo de desarrollo de la metodología Mobile-D

Fuente: [standardapp.blogspot.com](http://standardapp.blogspot.com) – Traducción propia.

#### **2.6.5.2.1.1.Fase de exploración.**

La primera etapa del ciclo de vida refleja, en su diferencia no muy notable al resto de etapas, el establecimiento del proyecto y su planificación inicial debidamente marcada.

Tiene tres etapas:

##### **2.6.5.2.1.1.1. Establecimiento de involucrados.**

En esta etapa se hace mención a las personas que interactúan con el sistema y se define el tipo de rol que poseen.

##### **2.6.5.2.1.1.2. Alcance del proyecto.**

Se define la visión general del sistema, indicando los alcances que se definió en el proyecto.

Para determinar el establecimiento de involucrados haremos uso de los diagramas de casos de uso.

#### **Diagramas de casos de uso.**

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

## Elementos Básicos.



**Figura 2.7: Elementos básicos de las diagramas de casos de uso.**

**Fuente: Diagramas de casos de uso – Universidad de Alcalá**

**Actores:** Los actores representan un tipo de usuario del sistema. Se entiende como usuario cualquier cosa externa que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático o unidades organizativas o empresas.

**Caso de uso:** Es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando. Se representan mediante un óvalo. Cada caso de uso debe detallarse, habitualmente mediante una descripción textual.

**Asociaciones:** Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso, entre los tipos de asociación tenemos:

**Include:** Se puede incluir una relación entre dos casos de uso de tipo “include” si se desea especificar comportamiento común en dos o más casos de uso.

Las ventajas de esta asociación son:

- Las descripciones de los casos de uso son más cortas y se entienden mejor.
- La identificación de funcionalidad común puede ayudar a descubrir el posible uso de componentes ya existentes en la implementación.

Las desventajas son:

- La inclusión de estas relaciones hace que los diagramas sean más difíciles de leer, sobre todo para los clientes.

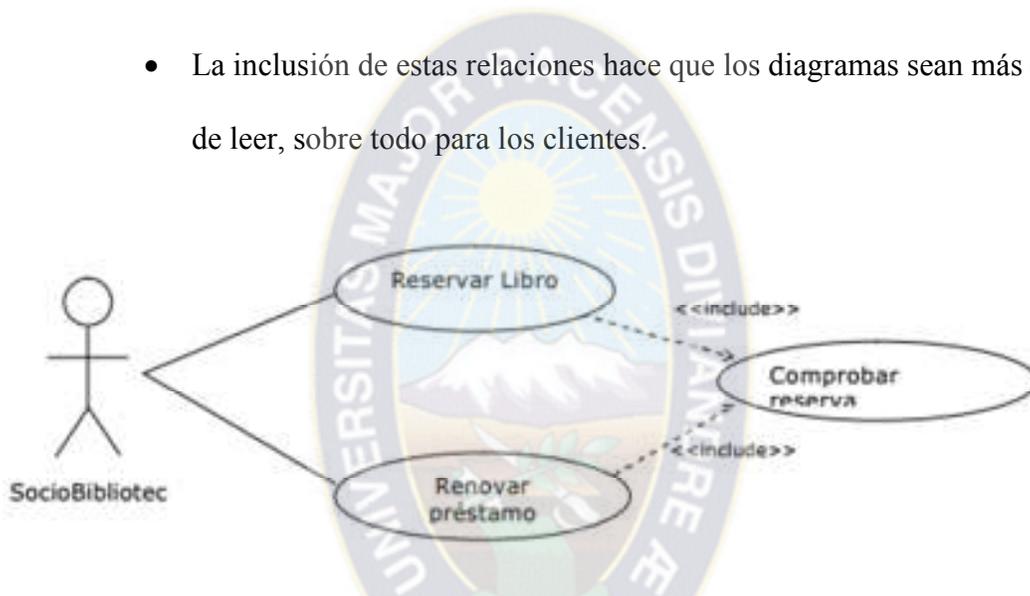
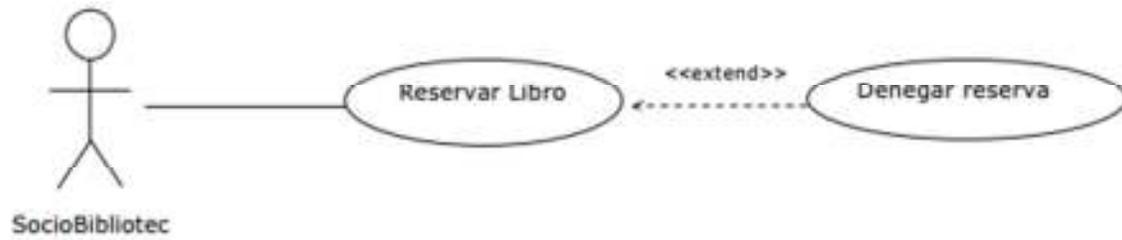


Figura 2.8: Elemento include

Fuente: Diagramas de casos de uso – Universidad de Alcalá

**Extend:** Se puede incluir una relación entre dos casos de uso de tipo “include” si se desea especificar diferentes variantes del mismo caso de uso. Es decir, esta relación implica que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias. En principio esas variaciones pueden también mostrarse como diferentes descripciones de escenarios asociadas al mismo caso de uso.

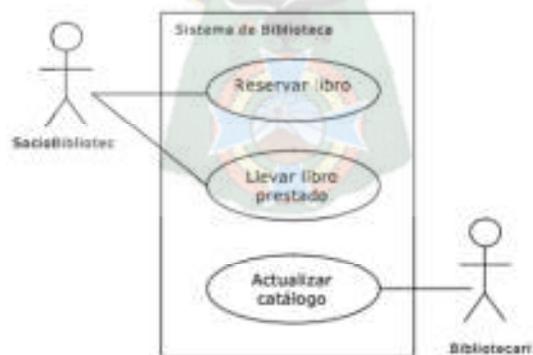


**Figura 2.9: Elemento extends**

Fuente: Diagramas de casos de uso – Universidad de Alcalá

**Generalizaciones:** En un diagrama de casos de uso también pueden mostrarse generalizaciones (relaciones de herencia) para mostrar que diferentes elementos están relacionados como tipos de otros. Son aplicables a actores o casos de uso, pero para estos últimos la semántica es muy similar a las relaciones “extend”.

**Límites del sistema:** Resulta útil dibujar los límites del sistema cuando se pretende hacer un diagrama de casos de uso para parte del sistema.



**Figura 2.10: Limite del sistema**

Fuente: Diagramas de casos de uso – Universidad de Alcalá

### **2.6.5.2.1.1.3. Establecimiento general del proyecto.**

La metodología en sus inicios propone una cooperación y participación activa de los clientes en esta fase ya que se generarán los requerimientos del sistema.

Si bien la metodología Mobile-D no cuenta con herramientas propias, en esta fase, para el establecimiento general del proyecto haremos uso de requerimientos funcionales y no funcionales para definir de mejor manera los requerimientos del sistema.

En esta etapa se tiene que definir el cronograma del proyecto y para describir las tareas a realizar emplearemos las tarjetas de historias de usuario.

- Requerimientos funcionales.

Describen la interacción entre el sistema y su ambiente independientemente de su implementación.

- Requerimientos no funcionales.

Los requerimientos no funcionales contemplan y especifican criterios en los que la operación de un sistema es satisfactoria.

- Historias de usuario.

El uso de historias de usuario es manera simple de describir una tarea concisa, que aporta valor al usuario o al negocio. No se detalla más hasta el momento que la historia de usuario se vaya a desarrollar.

Las historias de usuario pueden ser creadas durante la conversación con los usuarios interesados (stakeholders) sobre nuevas funcionalidades o mejoras del proyecto.

La manera más estándar de construir el enunciado es: Como Quiero Para.

Ejemplo. Quiero poder pulsar el botón. Quiero ver un listado con las marcas de coches. Quiero abrir la página desde mi móvil y sea agradable de ver.

Además, las Historias de Usuario deben cumplir las siguientes características para que puedan realizar su función de manera correcta:

**Independientes.** Deben ser atómicas en su definición. Es decir, se debe intentar que no dependa de otras historias para poder completarla.

**Negociables.** Como he dicho anteriormente, son entidades vivas. Deben ser ambiguas en su enunciado para poder debatirlas, dejando su concreción a los criterios de aceptación.

**Valoradas.** Deben ser valoradas por el cliente. Para poder saber cuánto aporta al Valor de la aplicación y junto con la estimación convertirse en un criterio de prioridad.

**Estimables.** Aunque sea siempre un poco como leer de una bola de cristal, deben poder ser estimadas. Tener su alcance lo suficientemente definido como para poder suponer una medida de trabajo en la que pueda ser completarla.

**Pequeñas.** Para poder realizar una estimación con cierta validez y no perder la visión de la Historia de Usuario, se recomienda que sean mayores de dos días y menores de dos semanas.

**Verificables.** Este es el gran avance de las Historias de Usuario. Que, junto con el cliente, se acuerdan unos Criterios de Aceptación que verifican si se ha cumplido con las funcionalidades descritas y esperadas.

Las tarjetas de historias de usuario contienen las siguientes partes, ver figura. 2.7.:

<b>Historia de Usuario</b>	
<b>Número:</b> 6	<b>Nombre:</b> Representación gráfica de magnitudes.
<b>Usuario:</b> Especialista	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2.0
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 2.2
<b>Descripción:</b> Se graficarán los valores de las magnitudes insertadas en la rejilla de datos de la ventana activa, para esto se mostrará un diálogo de selección donde se podrán escoger la magnitud o combinación de estas que se desea graficar, el tipo de gráfico y el rango de tiempo que abarcará la representación. El sistema analizará las selecciones y mostrará el gráfico correspondiente. Como resultado de la selección anterior se generará el gráfico con opciones de personalización, formato e impresión.	
<b>Observaciones:</b> La representación gráfica de magnitudes podrá realizarse desde las ventanas de trabajo Método de Balance, Tabla de Magnitudes del Balance del Método de Balance y Caudal del Escurrimiento Subterráneo.	

Figura 2.11.: Ejemplo de una tarjeta de historia de usuario

Fuente: [www.monografias.com](http://www.monografias.com)

En donde:

**Número**, es el orden correlativo que se le da cada historia de usuario, esto es para la identificación única de las tarjetas de historias de usuario.

**Nombre**, es la identificación que se le da a manera de nombre que por lo general es la función que realiza la tarjeta de historia de usuario.

**Iteración asignada**, es el orden con el cual se va a desarrollar.

**Puntos estimados**, es puntaje de dificultad que se le da al inicio del desarrollo.

**Puntos reales**, es el puntaje que se le da al finalizar el desarrollo de dicha historia de usuario.

**Prioridad de negocio**, es el nivel que le asigna a la historia de usuario dependiendo de su prioridad en cuanto al negocio.

**Prioridad de desarrollo**, es el nivel que le asigna a la historia de usuario dependiendo de su prioridad en cuanto al desarrollo del proyecto.

**Descripción**, es en donde se indica que es lo que se va a obtener al finalizar la tarjeta de historia de usuario.

**Observaciones**, es la parte donde se menciona alguna observación y/o detalle que se desea hacer notar al momento de realizar dicha tarjeta de historia de usuario.

#### **2.6.5.2.1.2.Fase de inicialización.**

En esta etapa los productores identifican, analizan y determinan los recursos primordialmente necesarios en primera instancia para la generación del producto.

##### **2.6.5.2.1.2.1. Configuración del proyecto.**

Se hace un desarrollo y aprovisionamiento de planes y documentación definida para las siguientes etapas. Además, y de suma importancia, se establece el enfoque y entorno técnico y tecnológico incluyendo en este, la capacitación del equipo de desarrollo y lo necesario para poder empezar a trabajar.

Los autores de esta metodología aseguran que su principal aporte al desarrollo ágil se encuentra y se apoya en esta etapa, delimitando y fortaleciendo el sondeo y la investigación con respecto a la línea arquitectónica.

##### **2.6.5.2.1.2.2. Día de planeación (iteración 0).**

Se examina el conocimiento y los patrones arquitectónicos usados por los clientes (si estos existen) o usados por los desarrolladores en ocasiones anteriores y a estos, los relacionan con el proyecto actual en curso.

##### **2.6.5.2.1.2.3. Día de trabajo (iteración 0).**

Es en donde se comienza a armar la arquitectura inicial del proyecto e identificar las herramientas que el equipo de desarrollo posee.

Se añaden comentarios, observaciones, se identifican diferencias y semejanzas entre casos de estudio y se toman decisiones sobre soluciones posibles para la implementación del proyecto.

#### **2.6.5.2.1.2.4. Día de lanzamiento (iteración 0).**

Es el punto en donde ya se tiene la arquitectura del proyecto y las definiciones para la implementación del proyecto en palabras simples, es el punto de inicio.

#### **2.6.5.2.1.3.Fase de producción o fase de producto.**

En esta fase se repite el patrón de la metodología por días y tareas específicas (planificación, trabajo, liberación) y se ejecutan estas a través de iteraciones hasta llegar a satisfacer todas las funcionalidades. Se podría decir que, en esta fase de la metodología se concentra el trabajo realizado por desarrolladores al momento de implementar tecnología. Los días y las tareas específicas se resuelven de la siguiente manera:

##### **2.6.5.2.1.3.1. Planificación.**

Se contempla la iteración de establecimiento de requisitos y tareas concretas por cumplir. Se realiza la implementación de requisitos para pruebas de iteración anterior.

##### **2.6.5.2.1.3.2. Trabajo.**

Dichos requerimientos por cumplir son satisfechos en el correspondiente día de trabajo produciendo el código, subiendo a repositorios diseñados para la integración y la unificación del mismo.

#### **2.6.5.2.1.3.3. Liberación.**

La integración y composición del sistema ocurre en este día, destinado para el armado de código (si hubiera dos o más equipos de desarrollo de forma independiente) y las pruebas adecuadas y ajustadas para su verificación.

#### **2.6.5.2.1.4.Fase de estabilización.**

Las últimas labores de unificación del código e integración se obtienen en esta etapa garantizando que el código combinado e implementado funcione fielmente apegado a los requerimientos. En los equipos de desarrollo conformados por varios grupos de desarrolladores, esta fase es primordial ya que se afina la integración entre componentes o subsistemas desarrollados.

##### **2.6.5.2.1.4.1. Planificación.**

Se identifican los últimos detalles para la presentación del primer prototipo.

##### **2.6.5.2.1.4.2. Trabajo.**

Se realiza la unificación del proyecto y se implementa los últimos detalles.

##### **2.6.5.2.1.4.3. Documentación.**

Es la parte donde se realiza la documentación del proyecto indicando las funcionalidades que posee el proyecto.

#### **2.6.5.2.1.4.4. Liberación.**

Se realiza la presentación del primer prototipo al cliente para que después se pueda realizar las pruebas.

#### **2.6.5.2.1.5.Fase de pruebas.**

Esta que es la última fase del proyecto posee el término de obtener un reléase funcional y con la suficiente estabilidad necesaria y requerida. El entregable terminado se prueba hacia los requerimientos del cliente y si se encuentran defectos en dicho entregable se hace una iteración más para eliminarlos.

#### **2.6.5.2.1.5.1. Testeo del sistema.**

Es la etapa en donde se realiza las pruebas tanto funcionales como de los requerimientos mencionados inicialmente.

#### **2.6.5.2.1.5.2. Codificación.**

Luego de obtener los resultados de las pruebas, se realizan las modificaciones que el cliente hizo y/o el equipo de desarrollo hizo notar.

#### **2.6.5.2.1.5.3. Liberación.**

Es la última etapa, y es en esta etapa en donde se realiza la presentación final del proyecto.

## **2.7.Métricas de calidad.**

**Calidad**, es el conjunto de propiedades o características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas (ISO 8402).

**Calidad de software**, es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

### **2.7.1. ISO 9000.**

Es un Conjunto de normas internacionales que definen requisitos para los llamados sistemas de gestión de calidad. ISO 9000 tiene reconocimiento internacional y se aplica a todas las industrias, incluso al desarrollo de tecnologías de la información. Los requisitos de la norma están en el documento conocido como ISO 9001:2000.

### **2.7.2. NORMA ISO/IEC 9126.**

Esta norma Internacional fue publicado en 1992, la cual es usada para la evaluación de la calidad de software, llamado Information technology - Software product evaluation - Quality characteristics and guidelines for their use; o también conocido como ISO 9126 (o ISO/IEC 9126).

La ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software.

Este estándar describe 6 características generales:



Figura 2.12: Calidad ISO/IEC

Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)

**Funcionalidad**, es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.



**Figura 2.13: Características de funcionalidad.**

Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)

Para determinar la funcionalidad de la aplicación hicimos uso del punto función:

- **Punto función.**

Punto función es un método utilizado en ingeniería del software para medir el tamaño del software. Fue definida por Allan Albrecht, de IBM, en 1979 y pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software, y también ser útil en cualquiera de las fases de vida del software.

La técnica de medición del tamaño en punto-función consiste en asignar una cantidad de "puntos" a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos, siempre tratando de considerarlo desde el punto de vista del usuario.

Se determinan cinco características de dominios de información y se definen de la siguiente forma:

**Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

**Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

**Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.

**Número de archivos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).

**Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad ver tabla 2.14.

Parámetro de medición	Cuenta	Factores de ponderación			Resultado
		Simple (x1)	Media (x3)	Complejo (x6)	
Número de entradas de usuario					
Número de salidas de usuario					
Número de peticiones de usuario					
Número de archivos					
Número de interfaces externas					
<b>Cuenta total</b>					

Tabla 2.14: Cálculo de puntos de función.

Fuente: Elaboración propia

Para calcular el punto función se utiliza la formula siguiente:

$$FP = \text{cuenta total} \times \left( 0,65 + 0,01x \sum F_i \right)$$

Dónde:

$\sum F_i$ , es el total del valor de ajuste.

Cuenta total, es la suma de todas las entradas del punto de función.

Además cuenta-total es la suma de todas las entradas obtenidas de la tabla 2.15.

N°	Factores de ajuste	Valor
1	¿Requiere el sistema copias de seguridad?	
2	¿Se requiere comunicación de datos?	
3	¿Existe funciones de procesamiento distribuido?	
4	¿Es crítico el rendimiento?	

5	¿Se ejecutará el sistema en un entorno operativo existente y utilizado?	
6	¿Se requiere entrada de datos?	
7	¿Requiere la entrada de datos que las transacciones de entrada se hagan sobre múltiples pantallas u operaciones?	
8	¿Se utilizan archivos maestros de forma interactiva?	
9	¿Son complejos las entradas, salidas, los archivos o las peticiones?	
10	¿Es complejo el procesamiento interno?	
11	¿Se ha diseñado el código para ser reutilizable?	
12	¿Están incluidas en el diseño la conversión y la instalación?	
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	
14	¿Se ha diseñado la aplicación para facilitar los cambios y ser fácilmente utilizada por el usuario?	
	<b>TOTAL <math>\Sigma F_i</math></b>	

**Tabla 2.15: Cálculo de puntos de función.**

**Fuente: Elaboración propia**

**Fiabilidad**, es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas, según Pressman es la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico.



**Figura 2.16: Características de fiabilidad.**

**Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)**

La fiabilidad del software es la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico (Pressman, 2002).

Para determinar la fiabilidad, en este proyecto se hizo uso del tiempo medio entre fallos

Entonces una sencilla medida de fiabilidad es el tiempo medio entre fallos y está dado por:

$$TMEF = TMDF + TMDR$$

Dónde:

TMEF es el tiempo medio entre fallos.

TMDF es el tiempo medio de fallo.

TMDR es el tiempo medio de reparación.

Luego debemos calcular la medida de disponibilidad del software, que será la probabilidad de que un programa funcione de acuerdo a los requisitos en un momento dado y se define como:

$$\text{Disponibilidad} = \frac{TMDF}{TMDF + TMDR}$$

**Usabilidad**, es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva y está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.

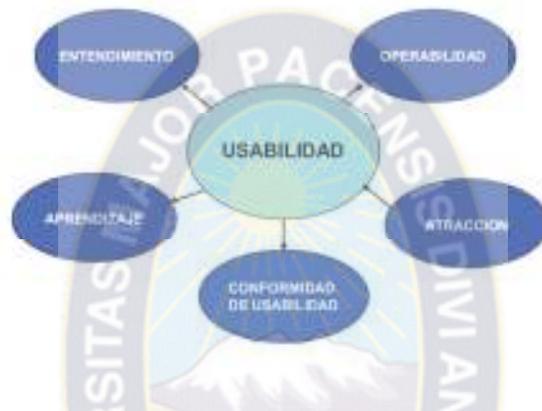


Figura 2.17: Características de usabilidad

Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)

**Eficiencia**, es el grado en el que el software es fácil de usar viene reflejada en la facilidad de comprensión, facilidad de aprendizaje y facilidad de operabilidad.



Figura 2.18: Características de eficiencia

Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)

**Mantenibilidad**, es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.



Figura 2.19: Características de mantenibilidad.

Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)

**Portabilidad**, es la capacidad que tiene el software para ser trasladado de un entorno a otro, es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio.



Figura 2.20: Características de portabilidad.

Fuente: [www.puntoexe.com.co](http://www.puntoexe.com.co)

Para obtener el resultado de usabilidad, eficiencia, mantenibilidad y portabilidad, haremos uso de los test de usuario.

- Test de usuario.

Una gestión total de la calidad asume que cualquier producto contiene debilidades y problemas. El objetivo de un test de usuario es encontrarlos antes de su lanzamiento al mercado.

Los objetivos del test parten al entender qué se quiere saber de un determinado producto y marcan quién debe participar en el test, qué tareas realizar, qué datos recoger, qué equipo utilizar, qué se mide, qué se considera éxito, cómo analizar la información, qué hacer con la información una vez analizada, etc.

Los objetivos ambiguos son una forma de empezar, pero deben concretarse hasta establecer la forma de medirlos:

Interés general. Ejemplo: ¿Serán los usuarios capaces de seleccionar correctamente las opciones del menú?

Interés concreto. Ejemplo: ¿Serán capaces de crear una factura? ¿De modificar una factura ya creada? ¿De enviar la factura en formato PDF por correo electrónico a un cliente?...

Forma de medirlo. Ejemplo: Errores en la selección. Tiempo para completar la tarea.

## CAPITULO III

### DESARROLLO DE LA APLICACIÓN

El desarrollo de la aplicación se enmarcó en la metodología Mobile-D, una metodología orientada exclusivamente al desarrollo de aplicaciones móviles, en este capítulo se describe el desarrollo del prototipo en cada una de sus fases.

#### 3.1.Exploración.

En esta fase se hace una planificación inicial del sistema en donde se menciona:

##### 3.1.1. Establecimientos de involucrados.

La aplicación móvil será usada por un usuario general que tendrá acceso a todas las opciones de la aplicación y por el administrador que tendrá acceso a todas las opciones del sistema web.

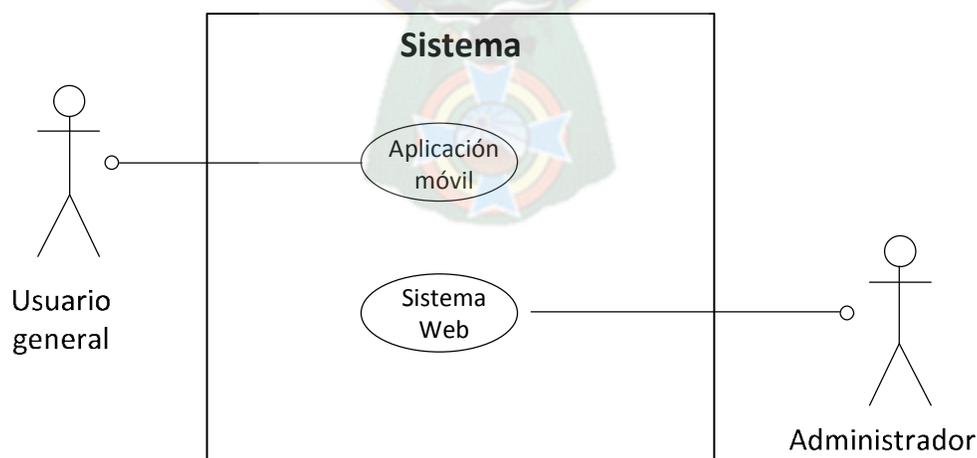


Figura 3.1: Caso de uso – Elementos involucrados

Fuente: Elaboración propia

Descripción del caso de uso:

Nombre	Elementos involucrados
Actores	Usuario general, administrador
Descripción	Usuario general, Se encarga de administrar la aplicación móvil y tiene acceso a visualizar todas las opciones de la aplicación. Administrador, se encarga del registro y actualización de datos y tiene acceso a todas las funcionalidades del sistema web.

**Tabla 3.1: Elementos involucrados**

**Fuente: Elaboración propia**

También visualizaremos a los integrantes para el desarrollo de la aplicación.

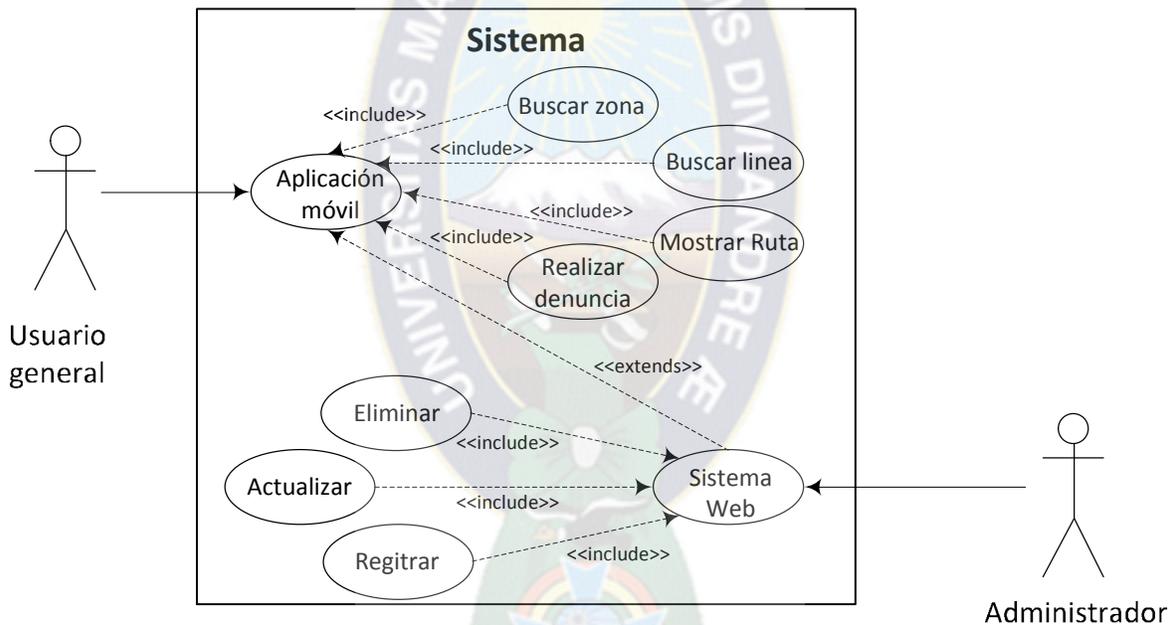
Rol	Responsabilidad	Recurso Humano
Jefe de Proyecto	Cuenta con habilidades de gestión de proyectos y conduce un proyecto desde su inicio hasta su lanzamiento.	Este trabajo recae sobre el autor, es decir, Freddy Lopez
Programador	Escribe las pruebas unitarias y produce el código del sistema.	Este trabajo recae sobre el autor, es decir, Freddy Lopez
Testing	Ejecuta las pruebas que escribe el programador, difunde los resultados al equipo y es el responsable de soporte para pruebas.	Asimismo por la naturaleza de la realización del proyecto, recae sobre Freddy Lopez
Cliente	Proporciona la información para el asentamiento de requerimientos funcionales como no funcionales y estima la implementación para el prototipo desarrollado.	La responsabilidad cae sobre la representación del gobierno autonomo municipal de El Alto, Lic. Rodrigo Calle, Responsable del proyecto

**Tabla 3.2.: Roles y responsabilidades.**

**Fuente: Elaboración propia.**

### 3.1.2. Alcance del proyecto.

La visión del presente proyecto es desarrollar un sistema de información de las rutas del transporte público vehicular de la ciudad de El Alto e implementar un módulo de denuncias en caso de que el conductor infrinja en el recorrido de la línea de transporte público, esto se hará mediante el uso de teléfonos móviles con sistema operativo android a efecto de maximizar el alcance del sistema.



**Figura 3.2: Caso de uso – Visión del proyecto**  
**Fuente: Elaboración propia**

- Descripción del caso de uso.

<b>Nombre</b>	Visión del proyecto
<b>Actores</b>	Usuario general, administrador
<b>Descripción</b>	<p>Usuario general, tiene las opciones de buscar rutas mediante el nombre de la zona y el número de línea, tiene la opción de visualizar las rutas y realizar denuncias.</p> <p>Administrador, se encarga de la administración de la base de datos (registrar, modificar y eliminar) y también de migrar los datos del sistema web a la aplicación móvil.</p>

**Tabla 3.3: Elementos involucrados**

**Fuente: Elaboración propia**

### **3.1.3. Establecimiento del proyecto.**

En esta parte mencionaremos los requerimientos iniciales del sistema y la planificación del sistema.

#### **3.1.3.1. Definición de requerimientos iniciales.**

- **Requerimientos funcionales.**

RF1. Visualización de zonas y líneas de transporte de la ciudad de El Alto.

RF2. Visualización las rutas del transporte público de El Alto.

RF4. Obtener métodos de búsqueda para ver el trazo de rutas de transporte.

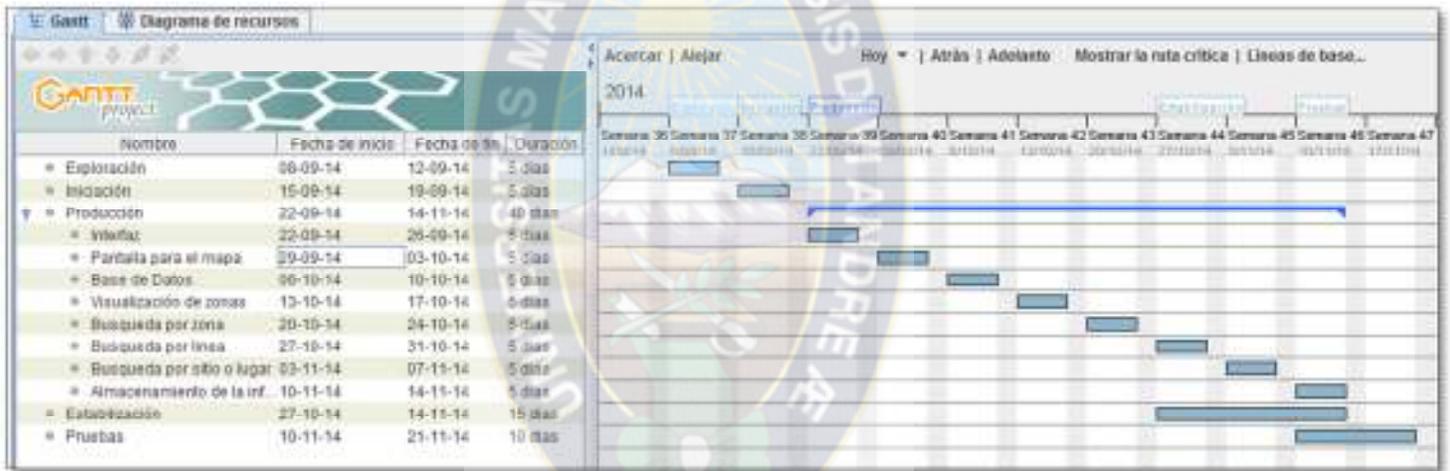
RF3. Enviar denuncias al gobierno autónomo de El Alto.

- **Requerimientos no funcionales.**

Para que un usuario pueda acceder a la visualización del mapa deberá tener una conexión a Internet a través de su dispositivo móvil, este deberá contar con un sistema operativo Android mínimo versión 2.3.

### 3.1.3.2. Planificación del proyecto.

#### Cronograma del proyecto.

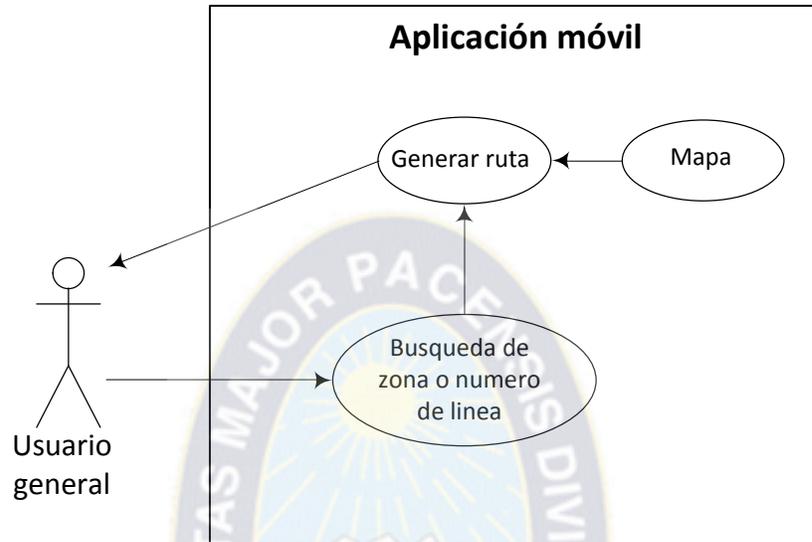


**Figura 3.3.: Cronograma del Proyecto de Producción.**  
Fuente: Elaboración propia.

### 3.1.3.3. Diagramas de casos de uso.

En los siguientes casos de uso se mostrara el proceso de generar una ruta y generar una denuncia de la aplicación móvil:

- Visualización de rutas.



**Figura 3.4: Caso de uso – Visualización de rutas**

**Fuente: Elaboración propia**

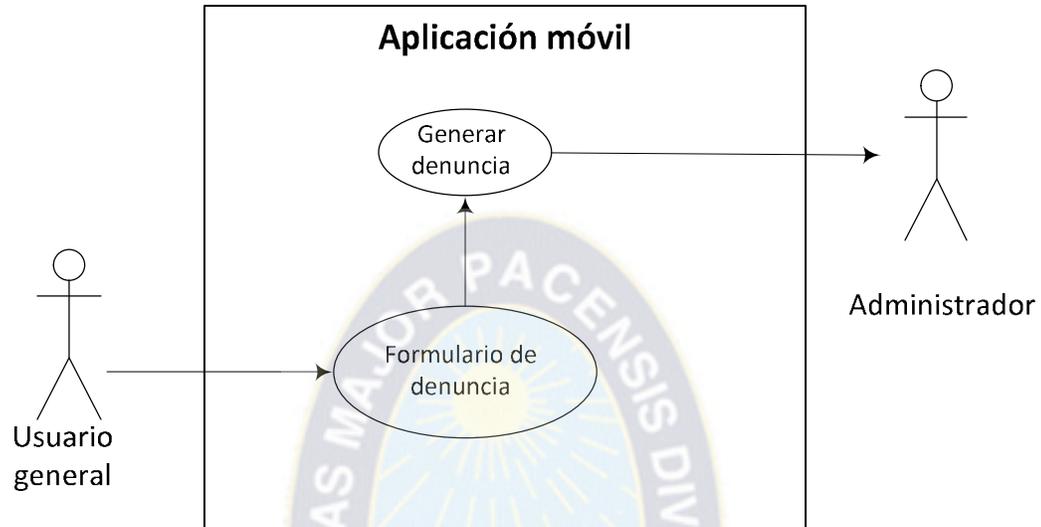
Descripción del caso de uso:

<b>Nombre</b>	Visualización de rutas
<b>Actores</b>	Usuario general
<b>Descripción</b>	La aplicación recibe el nombre de la zona o el número de la línea de transporte, seguidamente con el servicio de google maps la aplicación hace la visualización del trazo de la ruta para el usuario.

**Tabla 3.4: Visualización de rutas**

**Fuente: Elaboración propia**

- Generar denuncias.



**Figura 3.5: Caso de uso – Generar denuncia**

**Fuente: Elaboración propia**

Descripción del caso de uso:

<b>Nombre</b>	Elementos involucrados
<b>Actores</b>	Usuario general, administrador
<b>Descripción</b>	Usuario general, se encarga de llenar el formulario de denuncias, la aplicación genera un reporte y lo envía al administrador.

**Tabla 3.5: Generación de denuncias**

**Fuente: Elaboración propia**

### 3.1.3.4. Tarjetas de historias de usuario.

En las siguientes tablas mencionaremos las tarjetas de usuario para la implementación del proyecto:

Historia de Usuario	
<b>Numero :</b> 1	<b>Nombre de Historia de Usuario :</b> Interfaz
<b>Modificación de Historia de Usuario Número :</b>	
<b>Usuario :</b> Usuario móvil	<b>Iteración Asignada :</b> 1
<b>Prioridad en Negocio :</b> Alta	<b>Puntos Estimados :</b> 5
<b>Riesgo en Desarrollo :</b> Alto	<b>Puntos Reales :</b>
<b>Descripción :</b> Ver la interfaz de la aplicación.	
<b>Observaciones :</b>	

Tabla 3.6. Historia de Usuario 1  
Fuente: Elaboración propia.

<b>Historia de Usuario</b>	
<b>Numero :</b> 2	<b>Nombre de Historia de Usuario :</b> Mapa en la pantalla
<b>Modificación de Historia de Usuario Número :</b>	
<b>Usuario :</b> Usuario móvil	<b>Iteración Asignada :</b> 2
<b>Prioridad en Negocio :</b> Alta	<b>Puntos Estimados :</b> 5
<b>Riesgo en Desarrollo :</b> Alto	<b>Puntos Reales :</b>
<b>Descripción :</b> Ver un mapa en la pantalla de la aplicación el cual contendrá el trazo de las distintas rutas del transporte público de El Alto.	
<b>Observaciones :</b>	

**Tabla 3.7. Historia de Usuario 2**  
**Fuente: Elaboración propia.**

<b>Historia de Usuario</b>	
<b>Numero :</b> 3	<b>Nombre de Historia de Usuario :</b> Mostrar la ruta de una línea mediante el nombre de la zona.
<b>Modificación de Historia de Usuario Número :</b>	
<b>Usuario :</b> Usuario móvil	<b>Iteración Asignada :</b> 4
<b>Prioridad en Negocio :</b> Alta	<b>Puntos Estimados :</b> 5
<b>Riesgo en Desarrollo :</b> Alto	<b>Puntos Reales :</b>
<b>Descripción :</b> Ver el trazo de una ruta visualizado en un mapa por medio del nombre de la zona.	
<b>Observaciones :</b>	

**Tabla 3.8. Historia de Usuario 3**  
**Fuente: Elaboración propia.**

<b>Historia de Usuario</b>	
<b>Numero :</b> 4	<b>Nombre de Historia de Usuario :</b> Mostrar la ruta mediante el número de una línea.
<b>Modificación de Historia de Usuario Número :</b>	
<b>Usuario :</b> Usuario Móvil	<b>Iteración Asignada :</b> 4
<b>Prioridad en Negocio :</b> Alta	<b>Puntos Estimados :</b> 5
<b>Riesgo en Desarrollo :</b> Alto	<b>Puntos Reales :</b>
<b>Descripción :</b> Ver el trazo de una ruta visualizado en un mapa por medio del número de una línea del transporte público.	
<b>Observaciones :</b>	

**Tabla 3.9. Historia de Usuario 4**  
**Fuente: Elaboración propia.**

<b>Historia de Usuario</b>	
<b>Numero :</b> 5	<b>Nombre de Historia de Usuario :</b> Mostrar la ruta mediante el nombre del sitio o lugar conocido.
<b>Modificación de Historia de Usuario Número :</b>	
<b>Usuario :</b> Usuario móvil	<b>Iteración Asignada :</b> 4
<b>Prioridad en Negocio :</b> Alta	<b>Puntos Estimados :</b> 5
<b>Riesgo en Desarrollo :</b> Alto	<b>Puntos Reales :</b>
<b>Descripción :</b> Ver el trazo de una ruta visualizado en un mapa por medio del nombre del sitio o lugar mas conocido.	
<b>Observaciones :</b>	

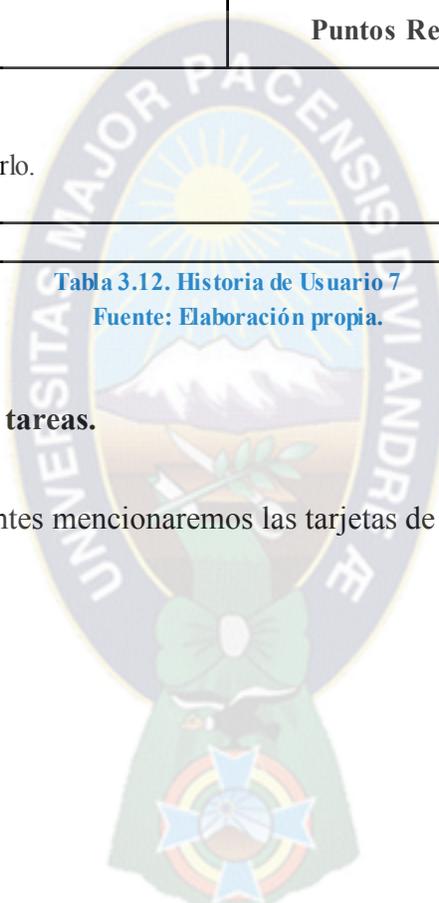
**Tabla 3.10. Historia de Usuario 5**  
**Fuente: Elaboración propia.**

<b>Historia de Usuario</b>	
<b>Numero :</b> 7	<b>Nombre de Historia de Usuario :</b> Módulo de denuncias
<b>Modificación de Historia de Usuario Número :</b>	
<b>Usuario :</b> Usuario móvil	<b>Iteración Asignada :</b> 5
<b>Prioridad en Negocio :</b> Alta	<b>Puntos Estimados :</b> 5
<b>Riesgo en Desarrollo :</b> Alto	<b>Puntos Reales :</b>
<b>Descripción :</b> Ver el un formulario y luego enviarlo.	
<b>Observaciones :</b>	

**Tabla 3.12. Historia de Usuario 7**  
Fuente: Elaboración propia.

### 3.1.3.5. Tarjetas de tareas.

En las tablas siguientes mencionaremos las tarjetas de tareas de cada tarjeta de historia de usuario ya vista.



Tarea de Ingeniería	
<b>Numero de Tarea :</b> 1.1.	<b>Número de historia de Usuario :</b> 1
<b>Nombre de Tarea :</b> Pantalla de Inicio	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 2
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b> <p>La pantalla de inicio nos mostrara el nombre y el logotipo de la aplicación el cual estara en el centro de la pantalla y tendra una duracion de 5 segundos.</p>	

**Tabla 3.13. Tarea de Ingeniería - Pantalla de Inicio**  
Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 1.2.	<b>Número de historia de Usuario :</b> 1
<b>Nombre de Tarea :</b> Pantalla Principal	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 3
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b> <p>La pantalla principal sera la encargada de la interacción con el usuario y contendra los distintos tipos de busqueda de una ruta del transporte publico.</p> <p>La pantalla estara dividida en cuatro partes, cada parte es para un tipo de busqueda y el ultimo es para la opcion que mostrara la ubicacion de las zonas de la ciudad de El Alto.</p>	

**Tabla 3.14. Tarea de Ingeniería - Pantalla Principal**  
Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 2.1.	<b>Número de historia de Usuario :</b> 2
<b>Nombre de Tarea :</b> Integración de google play services y adt android	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 1
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>  Descargar google play services con el sdk manager. Importar la librería de google play services al proyecto.	

**Tabla 3.15. Tarea de Ingeniería - Integración con google play services**  
Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 2.2.	<b>Número de historia de Usuario :</b> 2
<b>Nombre de Tarea :</b> Obtención del api key de google	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 1
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>  Obtener el código SHA1 del adt android. Obtener el api key de google para mapas en aplicaciones en android.	

**Tabla 3.16. Tarea de Ingeniería - Obtención del Api key de Google**  
Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 2.3.	<b>Número de historia de Usuario :</b> 2
<b>Nombre de Tarea :</b> Integracion del mapa en la pantalla de la aplicacion	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 3
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>  Mostrar en una pantalla completa el mapa que nos brinda google.	

**Tabla 3.17. Tarea de Ingeniería - Integracion del mapa en la pantalla**  
Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 3.1.	<b>Número de historia de Usuario :</b> 3
<b>Nombre de Tarea :</b> Diseño de la interfaz	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 2
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>  Esta pantalla estara compuesta por una campo de texto al centro, el cual servira para la inserción del nombre de la zona, ademas de tener un boton para realizar la verificación de la existencia de la zona. En caso de que la zona sea correcta, se mostrara el numero de las lineas que pasan por esa zona.	

**Tabla 3.18. Tarea de Ingeniería - Diseño de la Interfaz - Busqueda por zona**  
Fuente: Elaboración propia.

Tarea de Ingenieria	
<b>Numero de Tarea :</b> 3.2	<b>Número de historia de Usuario :</b> 3
<b>Nombre de Tarea :</b> Visualización de la ruta	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 3
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>	
<p>Esta pantalla esta compuesta por un mapa obtenido de google (ver Historia de Usuario 2), en el cual se visualizara el trazo y los puntos mas referentes de la ruta de la linea de transporte ademas de la ubicacion de la zona elejida (Ver Tarea 3.1.) en el mapa.</p>	

**Tabla 3.19. Tarea de Ingenieria - Visualizacion de la ruta - Busqueda por zona**

Fuente: Elaboración propia.

Tarea de Ingenieria	
<b>Numero de Tarea :</b> 4.1.	<b>Número de historia de Usuario :</b> 4
<b>Nombre de Tarea :</b> Diseño de la interfaz	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 2
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>	
<p>Esta pantalla estara compuesta por una campo de texto y un boton al centro, el cual servira para la inserción del numero de la linea, al momento de llenar el campo de texto, nos tendra que aparecer automaticamente los numeros de las lineas de transporte almacenados en la base de datos.</p>	

**Tabla 3.20. Tarea de Ingenieria - Diseño de la Interfaz - Busqueda por linea**

Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 4.2.	<b>Número de historia de Usuario :</b> 4
<b>Nombre de Tarea :</b> Visualización de la ruta	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 3
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>  Esta pantalla esta compuesta por un mapa obtenido de google (ver Historia de Usuario 2), en el cual se visualizara el trazo y los puntos mas referentes de la ruta de la linea de transporte.	

**Tabla 3.21. Tarea de Ingeniería - Visualizacion de la ruta - Busqueda por linea**

Fuente: Elaboración propia.

Tarea de Ingeniería	
<b>Numero de Tarea :</b> 5.1.	<b>Número de historia de Usuario :</b> 5
<b>Nombre de Tarea :</b> Diseño de la interfaz	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 2
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>  Esta pantalla estara compuesta por una campo de texto al centro, el cual servira para la inserción del nombre del sitio o lugar mas conocido, ademas de tener un boton para realizar la verificación de la existencia de la misma. En caso de que el sitio o lugar mas conocido sea correcta, se mostrara el numero de las lineas que pasan por esa zona.	

**Tabla 3.22. Tarea de Ingeniería - Diseño de la Interfaz - Busqueda por sitio o lugar**

Fuente: Elaboración propia.

<b>Tarea de Ingeniería</b>	
<b>Numero de Tarea :</b> 5.2	<b>Número de historia de Usuario :</b> 5
<b>Nombre de Tarea :</b> Visualización de la ruta	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 3
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>	
<p>Esta pantalla esta compuesta por un mapa obtenido de google (ver Historia de Usuario 2), en el cual se visualizara el trazo y los puntos mas referentes de la ruta de la línea de transporte ademas de la ubicación del sitio o lugar mas conocido (Ver Tarea 5.1.) en el mapa.</p>	

**Tabla 3.23. Tarea de Ingeniería - Visualizacion de la ruta - Busqueda por sitio o lugar**

Fuente: Elaboración propia.

<b>Tarea de Ingeniería</b>	
<b>Numero de Tarea :</b> 7.1.	<b>Número de historia de Usuario :</b> 7
<b>Nombre de Tarea :</b> Diseño de la interfaz	
<b>Nombre de Tarea :</b> Desarrollo Desarrollo / Corrección / Mejora / Otra	<b>Puntos Estimados :</b> 5
<b>Fecha Inicio :</b>	<b>Fecha Fin :</b>
<b>Programador Responsable :</b>	Freddy López Gómez
<b>Descripción :</b>	
<p>Esta pantalla estara compuesta por dos campos de texto, dos campos seleccionables y un boton el cual se encargara de enviar la información almacenada al administrador.</p>	

**Tabla 3.25. Tarea de Ingeniería - Diseño del modulo de denuncias**

Fuente: Elaboración propia.

### 3.1.3.6. Planificación del desarrollo.

En la tabla a continuación se describe la planificación hecha para cada una de las historias de usuario en el punto anterior.

<b>Iteración</b>	<b>Tarea(s)</b>	<b>Historia de usuario</b>
Primera iteración.	Armado de la arquitectura del proyecto. Desarrollo de la interfaz.	H. U. 1
Segunda iteración.	Integración del servicio de google maps con el proyecto.	H. U. 2
Tercera iteración.	Implementación del modelo de datos.	H. U. 6
Cuarta iteración.	Desarrollo de métodos de búsqueda y visualización de rutas.	H. U. 3,4,5
Quinta iteración.	Desarrollo del módulo de denuncias.	H. U. 7

**Tabla 3.26: Planificación por iteraciones.**

Fuente: Elaboración propia

### **3.2.Inicialización.**

En esta fase se determinara los recursos primordiales para la generación del producto.

#### **3.2.1. Configuración del proyecto.**

Para la construcción de la aplicación se vio conveniente usar las siguientes herramientas:

- IDE Eclipse Juno.
- SDK Manager V23.
- Google Maps.

#### **3.2.2. Día de planeación (iteración 0).**

Haremos mención de las primeras planificaciones para la inicialización del proyecto:

##### **3.2.2.1.Nombre de entidades.**

Para el caso de las tablas de la base de datos se utilizará el nombre escrito en letras minúsculas. El nombre de la tabla estará asociado al sustantivo al que hace referencia; si son varias palabras se debe de intercalar entre mayúsculas y minúsculas.

El nombre para las columnas empezara con un sustantivo también que representará el atributo seguido con el nombre de la tabla en el caso de que estos se repitan. El nombre escogido para la columna debe hacer referencia al tipo de información que almacenará.

##### **3.2.2.2. Clases.**

Para todo nombre de clase, la primera letra debe de ser Mayúscula. Si son varias palabras se debe de intercalar entre mayúsculas y minúsculas.

### **3.2.2.3.Métodos.**

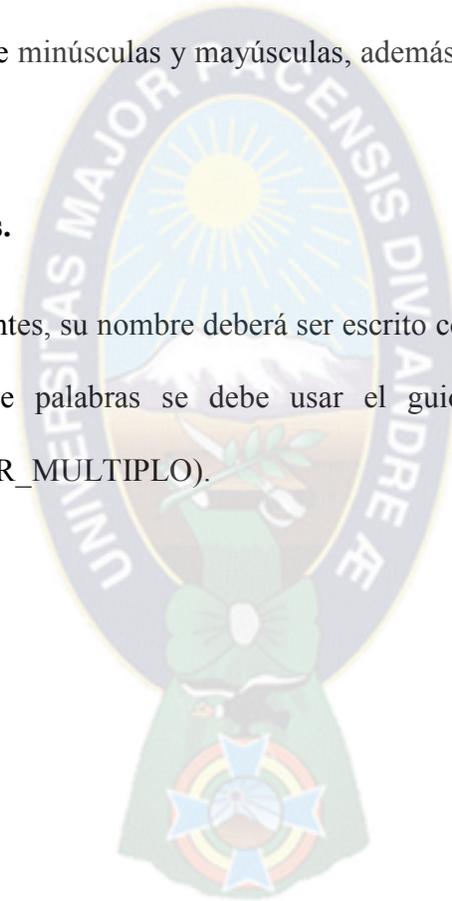
Para los métodos de clases, la primera letra debe ser minúscula. Si son varias palabras, se debe de intercalar entre minúsculas y mayúsculas.

### **3.2.2.4.Variables.**

Para las variables, la primera letra debe ser minúscula. Si son varias palabras, se debe de intercalar entre minúsculas y mayúsculas, además deben ser cortos y descriptivos en sí mismo.

### **3.2.2.5.Constantes.**

Para las constantes, su nombre deberá ser escrito completamente en Mayúsculas, y para la separación de palabras se debe usar el guión bajo ( \_ ). (Por ejemplo MAX\_SUMA, VALOR\_MULTIPLO).



### 3.2.2.6. Modelo de datos.

- Modelo Entidad - Relación.

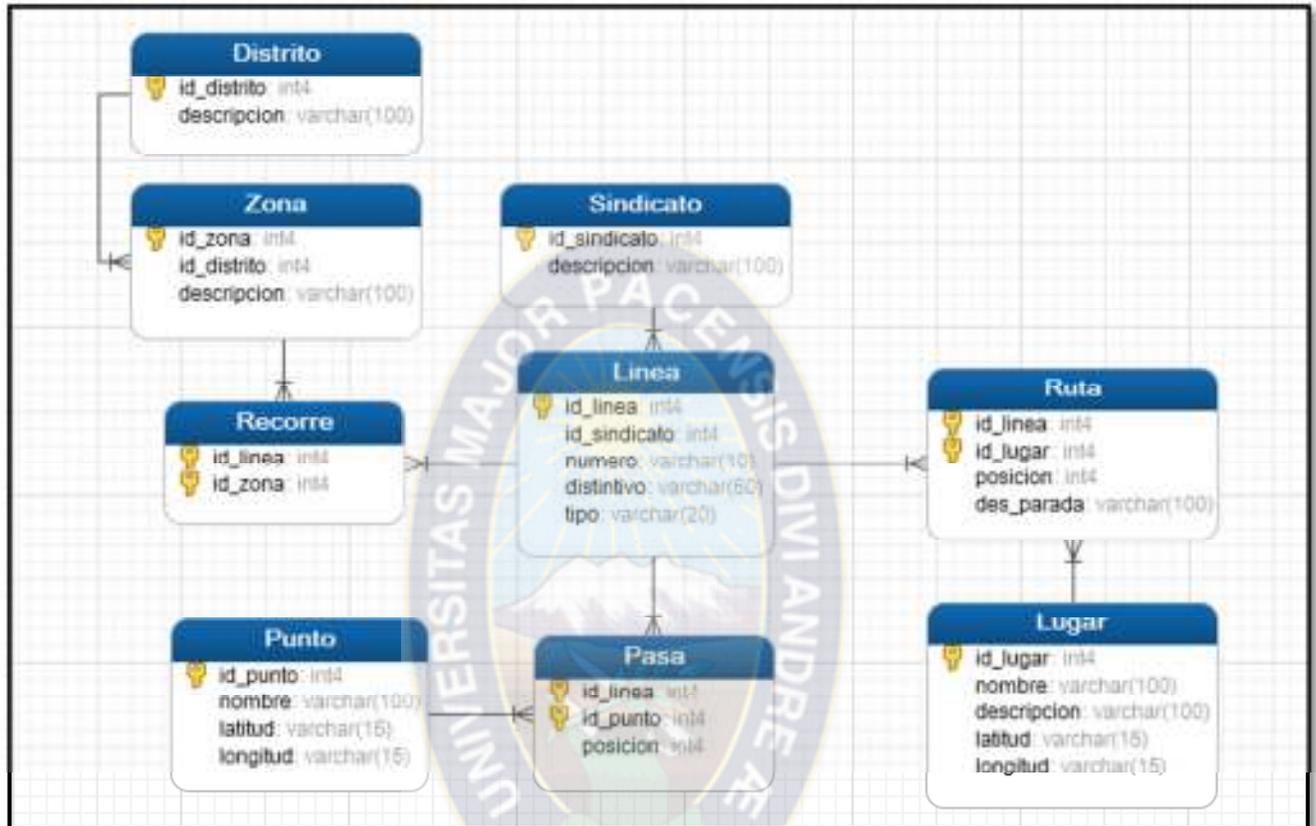


Figura 3.6.: Modelo de Datos  
Fuente: Elaboración propia.

- **Diccionario de datos.**

<b>Nro.</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
1.	Sindicato	Clase	Almacena información de los sindicatos existentes en la ciudad de El Alto.
2.	Línea	Clase	Guarda la información de las líneas que tiene cada sindicato de la ciudad de El Alto.
3.	Punto	Clase	Guarda la ubicación de un punto referente de alguna ruta.
4.	Pasa	Relación	Es la relación que existe entre Línea y Punto.
5.	Lugar	Clase	Almacena la ubicación de un lugar o sitio conocido.
6.	Ruta	Relación	Es la relación que existe entre Línea y Lugar.
7.	Distrito	Clase	Contiene la información de los 14 distritos existentes de la ciudad de El Alto.
8.	Zona	Clase	Guarda la información de las distintas zonas existentes dentro de un distrito.
9.	Recorre	Relación	Es la relación que existe entre Zona y Línea.

**Figura 3.7.: Identificación de identidades**  
Fuente: Elaboración propia.

### 3.2.3. Día lanzamiento (Iteración 0).

Ya definidas los aspectos iniciales para el desarrollo del proyecto damos inicio con el avance del proyecto el cual se describe en la siguiente fase.

### **3.3.Producción o fase de producto.**

La fase de producción de la metodología Mobile D asegura que debe existir un día de planeamiento, un día de programación y un día de liberación para cada iteración. A continuación se detallan las actividades seguidas en cada etapa de esta fase para el prototipo de la aplicación móvil.

Se establecen los requerimientos funcionales y no funcionales detallados en la sección 3.2.1 de este documento para el establecimiento de las historias de usuario y a su vez de las tareas de usuario.

#### **3.3.1. Seguimiento de iteraciones.**

Como se menciona en el apartado 3.3.2 de este documento, la planificación se planteó inicialmente en cinco iteraciones. Cada iteración tiene asignadas historias de usuario que han sido ordenadas por su prioridad, estas historias de usuario como se menciona anteriormente están compuestas por tarjetas de tareas representando las tareas individuales que tendrán que ser completadas para crear valor dentro del proyecto.

##### **3.3.1.1.Primer iteración.**

- **Planeación.**

En la primera iteración se armó la arquitectura del proyecto con las clases que inicialmente se propuso, también la interfaz principal y la pantalla de inicio.

- **Codificación.**

El proyecto se armó inicialmente con cinco clases con sus respectivas vistas, y se desarrolló la pantalla de inicio y la pantalla principal de acuerdo a la historia de usuario 1 y a las tarjetas de tareas 1.1, y 1,2.



Creación de la pantalla principal que nos muestra tres opciones:

- Búsqueda por zona.
- Búsqueda por número de línea.
- Búsqueda por lugar u otro.

**Figura 3.7.: Pantalla principal.**  
**Fuente: Elaboración propia**

### 3.3.1.2.Segunda iteración.

- **Planificación.**

En la segunda iteración se obtuvo el api key de google maps y se desarrolló la ventana para la visualización de los mapas en la aplicación.

- **Trabajo.**

Para el uso de mapas en android tenemos que obtener el api key de google masp e importar la librería de google play services, es por el cual que en esta iteración se obtuvo el api key de google y se desarrolló la ventana para visualizar el mapa de google en nuestra aplicación como indica en la historia de usuario 2 y las tarjetas de tareas 2.1 y 2.2.

- **Liberación.**



Se implementó la ventana que muestra el mapa haciendo uso del servicio de google maps.

**Figura 3.8.: Visualización del mapa de google maps en la aplicación**  
**Fuente: Elaboración propia**

### **3.3.1.3.Tercera iteración.**

- **Planificación.**

En la tercera iteración se procedió a la implementación de la base de datos del teléfono móvil (SQLite), en donde se realizó pruebas de almacenamiento.

- **Codificación.**

En esta iteración se procedió a crear la base de datos en la aplicación (SQLite) para luego realizar pruebas de almacenamiento y obtener datos para el trazo de las rutas, esto se realizó con comandos SQL para la inserción.

### **3.3.1.4.Cuarta iteración.**

- **Planificación.**

En la cuarta iteración se diseñó los tipos de búsqueda para obtener como resultado la visualización de una ruta, para el cual es necesario que el usuario indique la zona o el número de línea del transporte público y con ese dato la aplicación busca información en la base de datos para el trazo de la ruta indicada.

- **Codificación.**

De acuerdo con las historias de usuario 1, 2, 3 y con sus respectivas tarjetas de tareas, se desarrolló las ventanas para realizar los distintos tipos de búsqueda el cual coadyuva a la visualización de las rutas.

- Liberación.



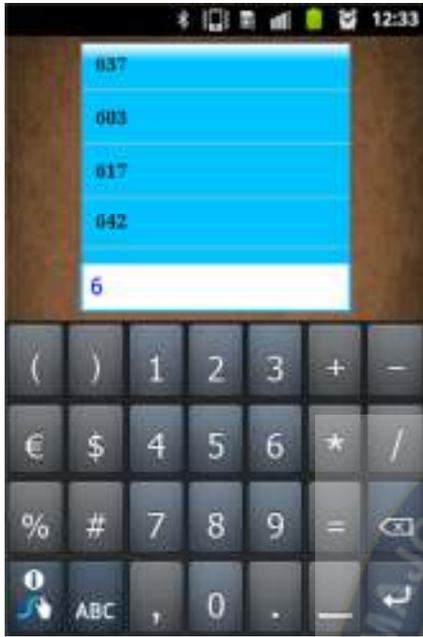
La ventana nos muestra un campo de texto en donde ingresamos el nombre de la zona y la aplicación nos muestra de manera inmediata las opciones de las zonas con ese mismo nombre.

**Figura 3.9.: Búsqueda de una ruta mediante una zona.**  
Fuente: Elaboración propia.



Luego de realizar la inserción del nombre de la zona, la aplicación nos muestra las opciones de las rutas que pasan por dicha zona.

**Figura 3.10.: Obtención de rutas mediante una zona.**  
Fuente: Elaboración propia



**Figura 3.11.: Búsqueda de una ruta mediante el número de una línea de transporte.**  
Fuente: Elaboración propia

Esta ventana contiene un campo de texto en donde se introduce el número de la línea que se desea ver, al momento de ingresar el número de línea la aplicación nos brinda una ayuda mostrando las líneas que están registradas y coinciden con el número ingresado.



**Figura 3.12.: Visualización de una ruta mediante una zona.**  
Fuente: Elaboración propia

Luego de realizar el proceso de introducir datos ya sea por la búsqueda por zonas o por el número de línea de transporte la aplicación nos muestra en la pantalla el trazo de la ruta haciendo uso del servicio de google maps.

### 3.3.1.5. Quinta iteración.

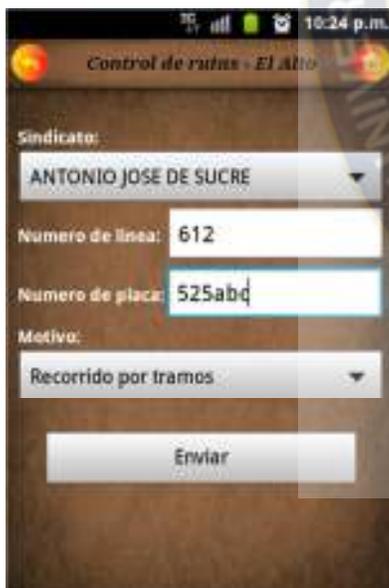
- **Planificación.**

En la quinta iteración se diseñó la ventana para realizar denuncias el cual consiste en un formulario donde el usuario tendrá que llenar la información.

- **Codificación.**

Para el módulo de denuncias se creó dos clases y una actividad para la vista el cual obtiene la información de la denuncia para luego enviar dicha información.

- **Liberación.**



La ventana nos muestra un pequeño formulario en donde el usuario tendrá que llenarlo para luego enviar el formulario a encargados del gobierno autónomo municipal de EL Alto.

**Figura 3.13.: Visualización del módulo de denuncias.**  
**Fuente: Elaboración propia**

### **3.4.Estabilización.**

Luego de realizar la fase de producción se pasó a realizar la documentación para después hacer la presentación del primer prototipo.

#### **3.4.1. Documentación.**

Se creó un manual de usuario con el fin facilitar el manejo a miembros de la alcaldía municipal de El Alto.

#### **3.4.2. Liberación.**

En fecha 26 de noviembre del año en curso se realizó una presentación con el primer prototipo de la aplicación en el cual estaban encargados de la dirección transporte y vialidad del gobierno autónomo municipal de El Alto y el Lic. Rodrigo Calle Peñaloza (encargado del proyecto por parte del GAMEA).

### **3.5.Pruebas del sistema.**

Esta es la última fase del desarrollo de la aplicación según la metodología Mobile-D. El detalle de las pruebas realizadas sobre la aplicación se detalla en el capítulo siguiente pruebas de calidad y resultados.

## CAPITULO IV

### PRUEBAS DE CALIDAD Y RESULTADOS

#### 4. Pruebas de calidad y resultados.

Este capítulo tiene como objetivo, determinar la calidad de la aplicación móvil y los procedimientos de seguridad. La calidad dentro del desarrollo de software es muy importante, la calidad se llega a verificar utilizando parámetros de medición en nuestro caso consideraremos la norma ISO-9126.

En el presente proyecto se evaluó la funcionalidad, fiabilidad, usabilidad, eficiencia y mantenibilidad.

#### 4.1.Métricas de calidad modelo ISO – 9126.

##### 4.1.1. Funcionalidad.

La funcionalidad se obtiene mediante un determinado “Punto Función” que se hacía en la notación empírica de medidas cuantitativas del dominio de información de software, el cual por formula tiene.

$$FP = cuenta\ total \times \left( 0,65 + 0,01x \sum F_i \right)_{(1)}$$

Dónde:

$\sum F_i$ , es el total del valor de ajuste.

Cuenta total, es la suma de todas las entradas del punto de función.

- Cálculo de  $\sum F_i$ .

Para el cual haremos uso de una escala de rangos de los valores de complejidad.

0	1	2	3	4	5
Sin influencia	Menor importancia	Moderado	Medio	Significativo	Esencial

Tabla 4.1: Valores de complejidad

Fuente: Elaboración propia

Ahora asignamos un valor a cada pregunta:

Numero	Factores de ajuste	Valor
1	¿Requiere el sistema copias de seguridad?	1
2	¿Se requiere comunicación de datos?	5
3	¿Existe funciones de procesamiento distribuido?	3
4	¿Es crítico el rendimiento?	2
5	¿Se ejecutará el sistema en un entorno operativo existente y utilizado?	3
6	¿Se requiere entrada de datos?	5
7	¿Requiere la entrada de datos que las transacciones de entrada se hagan sobre múltiples pantallas u operaciones?	5
8	¿Se utilizan archivos maestros de forma interactiva?	5
9	¿Son complejos las entradas, salidas, los archivos o las peticiones?	2
10	¿Es complejo el procesamiento interno?	3
11	¿Se ha diseñado el código para ser reutilizable?	3
12	¿Están incluidas en el diseño la conversión y la instalación?	5
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	5
14	¿Se ha diseñado la aplicación para facilitar los cambios y ser fácilmente utilizada por el usuario?	3
<b>TOTAL <math>\sum F_i</math></b>		<b>50</b>

Tabla 4.2: Valores de ajuste de complejidad según Pressman (2002)

Fuente: Elaboración propia

- Cálculo de cuenta total.

Para obtener el valor de la cuenta total requerimos de la siguiente información.

(a) Número de entradas de usuario.

<b>Numero</b>	<b>Entradas de usuario</b>	<b>Cantidad</b>
1	Ingreso a la aplicación	5
2	Ingreso a la pantalla de visualización	5
	<b>Total</b>	<b>10</b>

**Tabla 4.3: Entradas de usuario**

**Fuente: Elaboración propia**

(b) Número de salidas de usuario.

<b>Numero</b>	<b>Salidas de usuario</b>	<b>Cantidad</b>
1	Módulo de denuncias	4
	<b>Total</b>	<b>4</b>

**Tabla 4.4: Salidas de usuario**

**Fuente: Elaboración propia**

(c) Número de peticiones de usuario.

<b>Numero</b>	<b>Peticiones de usuario</b>	<b>Cantidad</b>
1	Lista de zonas	5
2	Lista de sindicatos	3
3	Lista de líneas de transporte	5
	<b>Total</b>	<b>13</b>

**Tabla 4.5: Peticiones de usuario de usuario**

**Fuente: Elaboración propia**

(d) Número de archivos.

<b>Numero</b>	<b>Archivos</b>	<b>Cantidad</b>
1	Base de datos (SQLite)	5
	<b>Total</b>	<b>5</b>

**Tabla 4.6: Archivos de usuario**

**Fuente: Elaboración propia**

(c) Número de interfaces externas.

En la aplicación móvil, no existen interfaces externas, por lo tanto el total es 0.

Posteriormente, recopilando los datos anteriores, calculamos los factores de ponderación.

<b>Parámetro de medición</b>	<b>Cuenta</b>	<b>Factores de ponderación</b>			<b>Resultado</b>
		<b>Simple</b>	<b>Media</b>	<b>Complejo</b>	
Número de entradas de usuario	10		x3		30
Número de salidas de usuario	4			x6	24
Número de peticiones de usuario	13		x3		39
Número de archivos	5		x3		15
Número de interfaces externas	0	x1			0
		<b>Total</b>			<b>108</b>

**Tabla 4.7: Cálculo de puntos de función.**

**Fuente: Elaboración propia**

Reemplazando en la formula (1), tenemos:

$$FP = 108 \times (0,65 + 0,01 \times 50)$$

$$FP = 124,2 \quad (2)$$

Luego obtenemos el FP ideal, eso se calcula usando el valor máximo de  $\sum F_i$

$$FP_{ideal} = 108 \times (0,65 + 0,01 \times 70)$$

$$FP_{ideal} = 145,8 \quad (3)$$

Ahora dividimos (2) y (3) para obtener la funcionalidad deseada:

$$Funcionalidad = \frac{FP}{FP_{ideal}} = \frac{124,2}{145,8} = 0,85 = 85 \%$$

Mediante el uso del punto función se obtuvo una funcionalidad del 85 %, el cual indica que el sistema de información satisface las necesidades de funcionalidad.

#### **4.1.2. Fiabilidad.**

La fiabilidad del software es la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico (Pressman, 2002).

Entonces una sencilla medida de fiabilidad es el tiempo medio entre fallos y está dado por:

$$TMEF = TMDF + TMDR \quad (1)$$

Dónde:

TMEF es el tiempo medio entre fallos.

TMDF es el tiempo medio de fallo.

TMDR es el tiempo medio de reparación.

Reemplazando en (1), tenemos:

$$TMEF = 8 \text{ hrs. de trabajo} * 0.7 \text{ hrs. de reparacion} = 5,6 \text{ hrs.}$$

Ahora debemos calcular la medida de disponibilidad del software, que será la probabilidad de que un programa funcione de acuerdo a los requisitos en un momento dado y se define como:

$$\text{Disponibilidad} = \frac{TMDF}{TMDF + TMDR} = \frac{8 \text{ hrs.}}{8 \text{ hrs.} + 0,7 \text{ hrs}} = 0,9195 = 92\%$$

El sistema obtuvo una fiabilidad de 92% el cual indica que el sistema de información tiene un buen nivel de funcionamiento en caso de errores.

#### 4.1.3. Usabilidad.

La usabilidad es el grado en el que el software es fácil de usar, para comprobar la usabilidad consideramos un método Test de Usuario el cual consiste en realizar una evaluación escrita después de las pruebas finales, así se obtienen los valores respondidos por los usuarios. En el presente se utilizaron las preguntas descritas en la tabla.

<b>Numero</b>	<b>Factor de ajuste</b>	<b>Valor obtenido</b>
<b>1</b>	Es entendible	90
<b>2</b>	Puede ser utilizado fácilmente	90
<b>3</b>	Es atractivo a la vista	80
	<b>Promedio</b>	<b>86.7</b>

**Tabla 4.8: Valores de ajuste en Test de usuario, según Pressman (2002)**

**Fuente: Elaboración propia**

La aplicación móvil obtuvo una usabilidad de 87%, el cual indica que el sistema de información es agradable en cuanto a la interacción con el usuario.

#### 4.1.4. Eficiencia.

La eficiencia es el grado en el que el software hace óptimo el uso de los recursos del sistema, la eficiencia está indicada por los tiempos de uso y recursos utilizados. Para evaluación se utilizó los datos descritos en la tabla 4.9.

<b>Numero</b>	<b>Factor de ajuste</b>	<b>Valor obtenido</b>
<b>1</b>	Es de respuesta rápida al ver la ruta.	80
<b>2</b>	Es de respuesta rápida al realizar una denuncia.	65
<b>3</b>	Responde adecuadamente cuando realiza las consultas.	90
	<b>Promedio</b>	78,3

**Tabla 4.9: Factores de eficiencia.**

**Fuente: Elaboración propia**

El sistema obtuvo una eficiencia de 78%, el cual indica que los tiempos de respuesta del sistema de información y procesamiento de datos son aceptables pero suficientes ya que depende de la conexión a internet de cada dispositivo móvil.

#### **4.1.5. Mantenibilidad.**

No hay forma de medir directamente la facilidad de mantenimiento, por ello se debe utilizar medidas indirectas (Pressman, 2002).

La facilidad con la que una modificación es realizada, está dada por, la facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba. Por lo tanto el programador debe hacerse las siguientes preguntas, ver tabla 4.10.

<b>Numero</b>	<b>Factor de ajuste</b>	<b>Valor obtenido</b>
<b>1</b>	Es fácil de analizar una falla o error	65
<b>2</b>	Se puede identificar las partes que deben ser modificados	70
<b>3</b>	Existe la facilidad de realizar cambios	75
<b>4</b>	Los cambios permiten una mejor estabilidad	80
<b>5</b>	Los cambios mejoran la facilidad de pruebas	80
	<b>Promedio</b>	82

**Tabla 4.10: Factores de ajuste de mantenibilidad.**

**Fuente: Elaboración propia**

El sistema de información obtuvo una mantenibilidad del 74 %, el cual indica que la capacidad de implementación de una modificación es relativamente aceptable.



## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5. Conclusiones y recomendaciones.

##### 5.1. Conclusiones.

Una vez concluida el presente proyecto, y realizado el análisis, desarrollo e implementación del sistema, llegamos a las siguientes conclusiones:

- El objetivo general fue cumplido ya que se implementó una aplicación que almacena las rutas de transporte público vehicular de la ciudad de El Alto, con el cual el usuario podrá controlar las rutas y esto con el fin de evitar, reclamar y denunciar el problema principal, el trameaje.
- Se logró interactuar con el servicio de google maps, esto con el fin de visualizar rutas de manera gráfica y sencilla.
- Se diseñó e implemento la base de datos en SQLite esto con el fin de no interactuar con terceros al momento de realizar consultas.
- Se implementó métodos de búsquedas con el objetivo principal de facilitar las consultas del usuario.

- Se logró distribuir la aplicación a funcionarios y personas externas del gobierno municipal de El Alto.
- Se logró la motivación de implementar más herramientas tecnológicas para tratar de solucionar problemas cotidianos que tiene el transporte en la ciudad de El Alto.
- Se logró hacer correr la aplicación sin conexión a internet, obteniendo las funcionalidades básicas de la aplicación.

## **5.2.Recomendaciones.**

### **5.2.1. A la institución.**

- Implementar mecanismos para adquirir la información actualizada de las rutas del transporte público de El Alto, ya que por el momento el gobierno municipal de El Alto cuenta con información desactualizada del transporte público en El Alto.
- Motivar el uso de la aplicación a la ciudadanía alteña e implementar mecanismos para corregir el mal actuar de algunos encargados del transporte público de El Alto.

### **5.2.2. A futuros proyectos.**

- Desarrollar la aplicación en los distintos tipos de sistemas operativos móviles.
- Implementar el módulo de registro y seguimiento de los conductores del transporte público vehicular, esto para que el control sea mas preciso.

## **Bibliografía.**

Amo, F. A. & Normand, L. M. (2005). *Introducción a la ingeniería del software*. Delta Publicaciones, 2005.

Fernandez, G. (2002). *Introducción a Extreme Programming*. Recuperado el 12 de septiembre de 2013, de <http://www.umedu.ar/>

Sampieri, N.J. (1998). *Metodología de la Investigación*. México: McGraw-Hill INTERAMERICANA EDITORES, S.A. de C.V.

Pressman R.S (1994), *Ingeniería del Software. Un enfoque práctico*. España: Mc Graw Hill.

Software de comunicaciones, (2011). *Programación en dispositivos móviles portables*. Recuperado de <https://sites.google.com/site/swcuc3m/>

Blanco, P., Camarero, J., Fumero, A., Werterski, A., & Rodriguez, P. (2009). *Metodología de desarrollo ágil para sistemas móviles*. Madrid: Universidad Politécnica de Madrid.

Salazar Guerrero A. (2010). *Tecnologías móviles*. Recuperado: <https://cursos.aiu.edu/>

Pedroso Petrazzini G. (2012). *Sistemas operativos en dispositivos móviles*. Argentina: Universidad Nacional del nordeste., Recuperado: <http://www.exa.unne.edu.ar/>

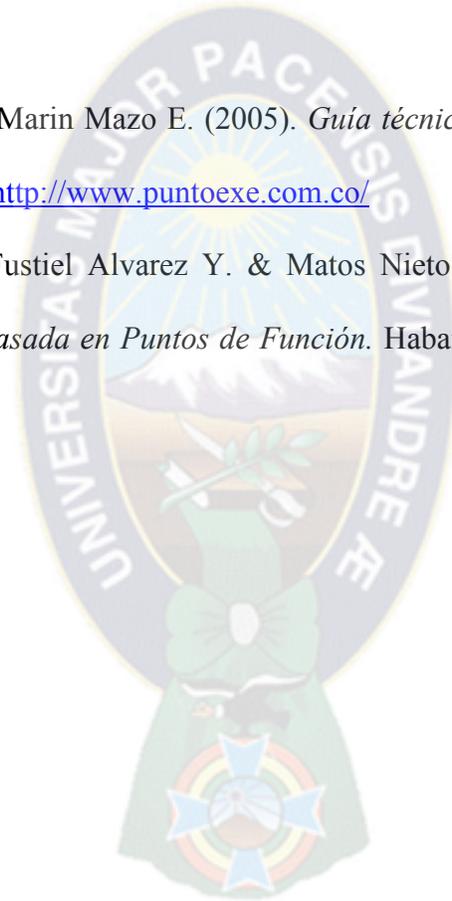
Ortiz Tamayo I. (2013). *Desarrollo de una aplicación móvil para recuento de stocks e inventarios transferido mediante web services*. Quito: Escuela politécnica nacional.

Quenta Carvajal J. (2013). *Aplicación de la realidad aumentada como medio educativo para la “U.E. Daniel Sánchez Bustamante” en el nivel secundario*. La Paz: Universidad mayor de San Andrés.

Alvarado Mallo S. (2013). *Realidad aumentada en aplicaciones móviles para la seguridad ciudadana en el transporte público*. La Paz: Universidad Mayor de San Andrés.

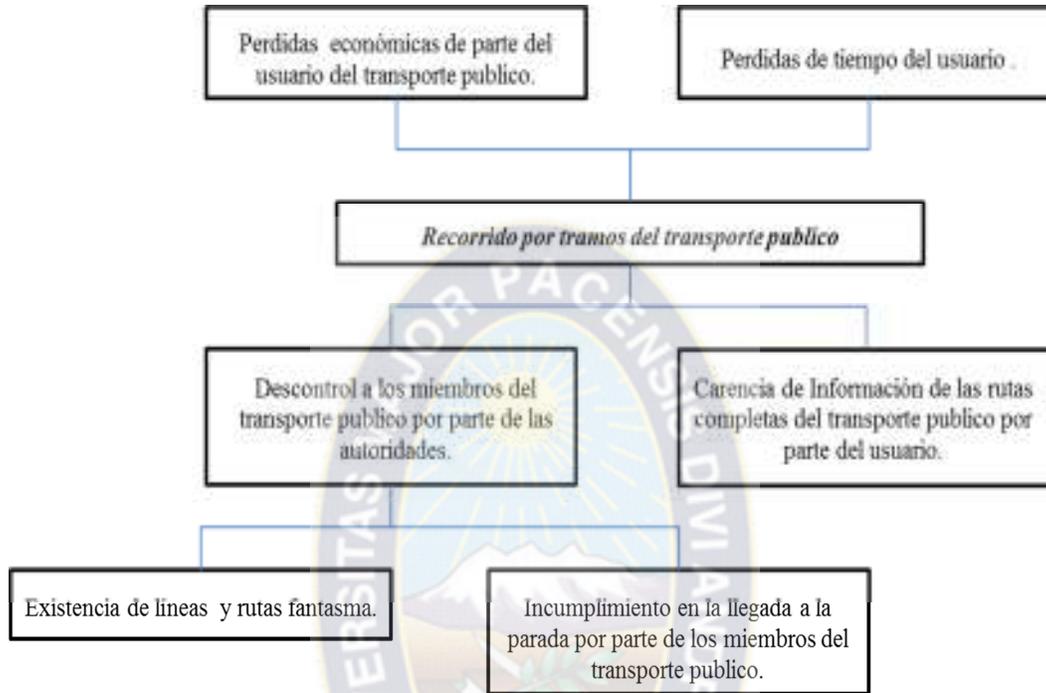
Largo Garcia C.A. & Marin Mazo E. (2005). *Guía técnica para evaluación de software*. Recuperado: <http://www.puntoexe.com.co/>

López Guerra E. & Fustiel Alvarez Y. & Matos Nieto A. & Pérez López E. (2007). *Estimación Basada en Puntos de Función*. Habana: Universidad de las Ciencias Informáticas.



## Anexos.

### A.1. Árbol de Problemas.



### A.2. Árbol de Objetivos.

