

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



TESIS DE GRADO

**“MODELO DE CONTROL DE TRAMOS DEL TRANSPORTE
PÚBLICO USANDO CÓDIGO QR Y APLICACIONES MÓVILES”**

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

POSTULANTE: JUAN CARLOS CHUQUIMIA BUSTENCIO

TUTOR METODOLÓGICO: Ph.D. YOHONI CUENCA SARZURI

ASESOR: M.Sc. CARLOS MULLISACA CHOQUE

**LA PAZ – BOLIVIA
2016**



**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 Dios por estar conmigo en cada momento de mi vida.
A mi madre Zenobia por el constante apoyo incondicional.
A mis hermanos Miguel, Limber, José, Romer.*

AGRADECIMIENTOS

A Dios por darme salud y permitirme culminar esta tesis, a mi madre por su apoyo, a mis hermanos por la confianza que tuvieron en mí y su apoyo.

A mi Tutor Ph.D. YOHONI CUENCA SARZURI por su tiempo y atención en las repetitivas revisiones de la tesis, brindándome su colaboración con toda su capacidad y conocimiento para la elaboración de esta tesis de Grado.

A mi revisor M. SC. CARLOS MULLISACA CHOQUE por el asesoramiento, por sus concejos, amplio conocimiento, para la elaboración del presente trabajo.

Al Lic. GROVER ALEX RODRIGUEZ RAMIREZ por la revisión de mi perfil en Taller 1 de la primera parte.

A mis compañeros por su orientación sobre las diferentes utilidades de la tecnología escogida.

A los docentes de la Carrera de Informática, por su enseñanza.

A los señores bibliotecarios por sus concejos y su apoyo.

RESUMEN

Las nuevas tecnologías emergentes nos permiten crear nuevas soluciones a los problemas que existen en nuestra sociedad, uno de estos problemas es el denominado “trameaje” el cual perjudica de gran manera a la población que hace uso del servicio de transporte público en la ciudad de La Paz por esta razón se desarrolló el presente trabajo titulado: “Modelo de Control de Tramos del Transporte Público usando Código QR y Aplicaciones Móviles” la cual ha sido implementada pensando en subsanar este problema, permitiendo al usuario realizar un control del vehículo acorde a su ruta y así de esta manera mejorar el servicio de transporte público para llegar a sus destinos.

El modelo fue desarrollado con la metodología Mobile-D, orientado al desarrollo de aplicaciones móviles, en lo referente a las herramientas fue Android Studio que nos permitió desarrollar la interfaz de usuario e integrar las diferentes tecnologías y herramientas que se utilizó en este trabajo, para el desarrollo del Modelo de Control de Tramos se utilizaron librerías como ZXing y Barcode para crear y reconocer códigos QR, Google Maps el cual proporciona las APIs necesarias que nos permiten mostrar el mapa, trazar la trayectoria de la línea vehicular en nuestra aplicación móvil, debemos mencionar también que recurrimos a la Geolocalización, el cual nos permite mostrar la posición exacta del vehículo en la ruta, con este proceso se busca el control de tramos.

La implementación del “Modelo de Control de Tramos del Transporte Público usando Código QR y Aplicaciones Móviles” busca eliminar el denominado “trameaje” brindando información primero de manera offline para después mostrarnos la información completa en tiempo real de las rutas de un número de línea vehicular, para esto se recurre al uso de códigos QR en los cuales se almacena información acerca del conductor, vehículo y sindicato los cuales serán reconocidos por medio de la aplicación “Mi Ruta” a través de la cámara del dispositivo móvil para luego mostrarnos el mapa con la ruta de línea vehicular y la posición del vehículo de transporte público en la ruta, colaborando de esta manera al control de tramos por parte del usuario.

ABSTRACT

Emerging technologies allow us to create new solutions to the existing problems our society, one of these problems is called "trameaje" which harms greatly the population that makes use of public transport in the city of La Paz for this reason this paper entitled developed: "Control Model stretches of Public Transport using QR Code and Mobile Applications" which has been implemented planning to remedy this problem by allowing the user to control the vehicle according to your route and and thus improve public transport to reach their destinations.

The model was developed with the Mobile-D methodology, oriented mobile application development, in terms of the tools was Android Studio that allowed us to develop the user interface and integrate the different technologies and tools used in this work, for the development of Control Model stretches libraries like ZXing and Barcode was used to create and recognize QR codes, Google Maps which provides the necessary APIs that allow us to show the map, tracing the path of the vehicle line in our mobile application, we also mention that resort to geolocation, which allows us to show the exact position of the vehicle on the road, with the process control sections is sought.

The implementation of "Control Model Stretches Public Transport using QR Code and Mobile Applications" seeks to eliminate the so-called "trameaje" providing information first so offline and then show us the complete information in real time the routes of a number of vehicular online for this is resorted to using QR codes in which information is stored about the driver, vehicle and union which will be recognized through the "My Route" application through the mobile device camera and then show us the map vehicular route line and the position of public transport vehicle on the road, thus helping to control sections by the user.

Índice general

CAPÍTULO 1. MARCO REFERENCIAL	1
1.1 INTRODUCCIÓN	1
1.2. PROBLEMA	2
1.2.1 ANTECEDENTES.....	2
1.2.2 LISTA DE PROBLEMAS.....	5
1.2.3 FORMULACIÓN DEL PROBLEMA.....	5
1.3. OBJETIVOS	5
1.3.1. Objetivo General.....	5
1.3.2 Objetivos Específicos.....	6
1.4. HIPÓTESIS	6
1.5. JUSTIFICACIÓN	6
1.5.1 Justificación Social.....	6
1.5.2 Justificación Económica.....	6
1.5.3 Justificación Tecnológica	7
1.5.4 Justificación Científica	7
1.6. ALCANCE	7
1.7. METODOLOGÍA.....	8
1.7.1 Observación.....	8
1.7.2 Formulación de hipótesis	8
1.7.3 Experimentación	8
1.7.4 Emisión de conclusiones.....	9
CAPÍTULO 2. MARCO TEÓRICO	10
2.1 PLATAFORMA ANDROID.....	10
2.1.1 Características.....	11
2.1.2 Arquitectura	12
2.1.3 La máquina virtual.....	13
2.2 CÓDIGO QR “QUICK RESPONSE CODE”	14
2.2.1 Características generales	14
2.2.2 Conceptos básicos de una estructura de código QR.....	15
2.2.3 Corrección de errores	17
2.2.4 Área de impresión reducida	17
2.2.5 Alta capacidad de codificación de los datos.....	18
2.2.6 Resistentes a suciedad o daño parcial.....	18
2.2.7 Legibles desde cualquier dirección en 360°	19
2.2.8 Obteniendo datos de un código QR	19
2.3 MOBILE D	20
2.3.1 Exploración	21

2.3.2	Iniciación.....	21
2.3.3	Producto	22
2.3.4	Estabilización.....	23
2.3.5	Pruebas	23
2.4	GEOLocalIZACIÓN	24
2.4.1	Geolocalización, localizar y GPS.	25
2.5	TECNOLOGÍAS A UTILIZAR	26
2.5.1	Google Maps	26
2.5.2	Google Maps API.....	27
2.5.3	La Google Places API Web Service.....	28
2.5.4	Google Maps Directions API	29
2.5.5	Waypoints.....	30
2.5.6	Java Barcode Generation	31
2.5.7	Librería ZXing	32
CAPÍTULO 3.	MARCO APLICATIVO	33
3.1	INTRODUCCIÓN	33
3.2	FASE DE EXPLORACIÓN	33
3.2.1.	Establecimiento de los Usuarios	33
3.2.2	Requerimientos del producto	34
3.2.3	Arquitectura	34
3.3	FASE INICIALIZACIÓN	35
3.3.1	Establecimientos de recursos	36
3.3.4	Plan de Iteraciones del desarrollo	44
3.4	FASE DE PRODUCCIÓN.....	44
3.4.1	Primera Iteración	45
3.4.2	Segunda Iteración.....	47
3.4.3	Tercera Iteración.....	50
3.4.4	Cuarta Iteración	52
3.4.5	Quinta Iteración	53
3.5	FASE DE ESTABILIZACIÓN.....	54
3.5.1	Workshop de post Iteración	54
3.6	FASE DE PRUEBAS.....	55
3.7	RESULTADOS.....	59
3.7.1	Resultados Primera Prueba.....	59
3.7.2	Resultados Segunda Prueba	60
3.7.3	Resultados Tercera Prueba	60
3.7.4	Resultado Total.....	61
CAPÍTULO 4.	PRUEBA DE HIPÓTESIS	62
4.1	Prueba De Hipótesis	62
4.2	Pasos A Seguir En La Prueba De Hipótesis	64
4.3	Evaluación De Resultados	64
4.4	Determinación De La Región Crítica	65

4.5 Toma de Decisión	66
4.6 Evaluación de la variable independiente	67
4.7 Aceptación de la aplicación	67
4.8 Resultados de las Encuestas	67
CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES	72
5.1 CONCLUSIONES.....	72
5.2 RECOMENDACIONES	73
 BIBLIOGRAFÍA.....	 75
 ANEXO A	 78
ANEXO B	79
ANEXO C	81

ÍNDICE DE FIGURAS

Figura 2.1 Logo Android	10
Figura 2.2 Arquitectura de Android	13
Figura 2.3 Estructura Genérica del Símbolo QR	15
Figura 2.4 Versiones de los Códigos QR	16
Figura 2.5 Comparativa espacio de impresión código de barras vs. QR	17
Figura 2.6 Vista general del Código QR.....	18
Figura 2.7 Diferentes tipos de daño en un código QR impreso	19
Figura 2.8 Patrones de detección de posición	19
Figura 2.9 Proceso de “mobile tagging” ó “etiquetado móvil”	20
Figura 2.10 Ciclo de Desarrollo Mobile D.....	21
Figura 3.1 Arquitectura del Modelo de Control de Tramos	35
Figura 3.2 Modelo de entidad relación	36
Figura 3.3 Pantalla Crear Código QR	45
Figura 3.4 Pantalla Principal	48
Figura 3.5 Pantalla información obtenida del Código QR.....	49
Figura 3.6 Visualización del mapa y la ruta.....	51
Figura 3.7 Pantalla Denuncia.....	53
Figura 3.8 Pantalla crear QR	55
Figura 3.9 Código QR.....	56
Figura 3.10 Minibús Línea 360.....	56
Figura 3.11 Escaneando el Código QR.....	57
Figura 3.12 Información de línea vehicular	57
Figura 3.13 Ruta de Línea Vehicular.....	58
Figura 3.14 Ventana Denuncia	59
Figura 4.1 Región Crítica para la hipótesis	65
Figura 4.2 Resultado tabla de función de Distribución normal	66
Figura 4.3 Distribución de Z en el grafico para la toma de decisión	67
Figura 4.4 Reconocimiento de Código QR.....	68
Figura 4.5 Ubicación en la Ruta.....	68
Figura 4.6 Distancia de escaneo de Código QR	69
Figura 4.7 Información de línea vehicular	69
Figura 4.8 Desempeño aplicación Mi Ruta.....	70
Figura 4.9 Volvería a utilizar la aplicación	70
Figura 4.10 Recomendaría la aplicación Mi Ruta.....	71

ÍNDICE DE TABLAS

Tabla 2.1 Capacidades máximas de datos de Código QR	18
Tabla 3.1 Descripción de roles de los usuarios	34
Tabla 3.2 Historia de Usuario 1	37
Tabla 3.3 Tarea 1.1 Diseño de Interfaz para crear códigos QR	37
Tabla 3.4 Tarea 1.2 Generar una Imagen QR.....	38
Tabla 3.5 Historia de Usuario 2.....	38
Tabla 3.6 Tarea 2.1 Diseño de Interfaz de la Ventana Principal.....	39
Tabla 3.7 Tarea 2.2 Reconocimiento del símbolo QR	39
Tabla 3.8 Historia de Usuario 3.....	40
Tabla 3.9 Historia de Usuario 4.....	40
Tabla 3.10 Tarea 4.1 Diseño de la Interfaz para Mostrar el Mapa.....	41
Tabla 3.11 Tarea 4.2 Api de Google Maps.....	41
Tabla 3.12 Tarea 4.3 Google Place API Web Service.....	41
Tabla 3.13 Historia de Usuario 5.....	42
Tabla 3.14 Tarea 5.1 Integrar GPS	42
Tabla 3.15 Historia de Usuario 6.....	43
Tabla 3.16 Tarea 6.1 Tarea 6.1 Diseño de Interfaz Pantalla Denuncia	43
Tabla 3.17 Plan de Iteraciones del Desarrollo.....	44
Tabla 3.18 Información contenida en el código QR.....	46
Tabla 3.19 prueba de aceptación 1	46
Tabla 3.20 Prueba de Aceptación 2	50
Tabla 3.21 Prueba de Aceptación 3	51
Tabla 3.22 Prueba de Aceptación 4	52
Tabla 3.23 Prueba de Aceptación 5	54
Tabla 3.24 Tamaños y distancia de escaneo de un código QR.....	60
Tabla 3.25 Resultados Primera Prueba	60
Tabla 3.26 Resultado Segunda Prueba	60
Tabla 3.27 Resultado Tercera Prueba	61
Tabla 3.28 Resultado Total.....	61

Capítulo 1.

MARCO REFERENCIAL

1.1 INTRODUCCIÓN

Actualmente los dispositivos móviles como los teléfonos móviles juegan un papel muy importante en la sociedad, ya que permite la constante interrelación y comunicación entre personas, sociedades, empresas y los demás actores del mundo moderno, es por esta razón que los dispositivos móviles con sistemas operativos como Android, iOS, Windows Phone, BlackBerry, etc. se han adoptado a nuestras vidas, con estos dispositivos móviles podemos comunicarnos, tomar fotos, escuchar música, descargar aplicaciones, etc.

Esto ha provocado el desarrollo de aplicaciones orientadas al uso y explotación de las capacidades de los dispositivos móviles. Dentro de estas aplicaciones, destacan aquellas en las cuales se han hecho aportaciones singulares que explotan las capacidades de los dispositivos móviles como son los códigos QR, (Quick Response code), son códigos bidimensionales también conocidos como "códigos de respuesta rápida", los cuales son capaces de contener información legible mediante una cámara y un software de lectura por lo que se utilizan como recurso de control. En otros países tanto en ámbitos de publicidad como en contextos comerciales.

Otro aspecto que se toma en cuenta es la geolocalización es un concepto relativamente nuevo, que ha surgido en los últimos dos años a esta parte y que hace referencia al conocimiento de la posición geográfica de modo automático, también denominada georreferenciación. La geolocalización implica el posicionamiento que define la localización de un objeto en un sistema de coordenadas determinado. Este proceso es generalmente empleado por los sistemas de información que se encuentra diseñado especialmente para capturar, almacenar, manipular y analizar en todas sus posibles formas la información geográfica referenciada.

Existen diferentes alternativas para conocer nuestra ubicación pero son los dispositivos móviles los que por su portabilidad con nosotros mismos nos permiten más

fácilmente conocer nuestra ubicación y actualizarla a medida que nos vamos movilizandoy por tanto, cambiando de ubicación geográfica, estos teléfonos móviles cuentan con funciones especializadas, traen integrados receptores de GPS, los cuales y gracias a la red de satélites que rodea al planeta podrán ubicarnos en cualquier parte del globo terráqueo en el cual nos encontremos.

Entonces lo que se pretende es usar los códigos QR como una tecnología que puede incorporarse al control de tramos en el transporte público de la ciudad de La Paz, además gracias al uso de dispositivos móviles y la geolocalización, se busca dar solución al problema denominado “tramos” en el transporte público, ya que pese a normativas y reglamentos “tramos” por parte de algunos choferes continúan, principalmente en lugares alejados, como las Villas, debido a que los choferes usan el denominado disco, en el cual se observa el número de pasajeros. Esta situación perjudica en gran manera a los pasajeros que deben esperar un vehículo para llegar hasta su hogar”. El incumplimiento de las tarifas todavía afecta a las personas que viven principalmente en las Villas.

1.2. PROBLEMA

1.2.1 ANTECEDENTES

El problema de transporte en la ciudad de La Paz es un tema que ha ido creciendo con el tiempo. El transporte público inicialmente fue una respuesta a la necesidad de crear un medio de transporte para un grupo de la población que requería una ocupación y un ingreso. Con el tiempo el servicio se fue miniaturizando y la competencia del centavo se incrementó, lo que resultó en un servicio muy deficiente.

Una expresión ya común es “tramo”, mucha gente se queda un tanto azorada cuando los propios comunicadores utilizan esta palabra para referirse a los trayectos cada vez más cortos de las distintas líneas de transporte público.

Recordemos que la palabra “tramo”, según diccionario de la Real Academia de la Lengua Española, se refiere a: “Parte en que está dividida una cosa que se desarrolla linealmente, como un camino, calle, etc.”. Suponemos que por ello a algún ingenioso dirigente de los transportistas públicos se le ocurrió ante la imposibilidad de lograr elevar el precio de las tarifas, prestar servicios solamente por tramos, lo que hoy hace que el

ciudadano de a pie tenga que gastar hasta más de tres veces, por concepto de pasaje, para trasladarse desde su barrio hasta cualquier otro punto de la urbe, particularmente en el servicio nocturno de los “minibuses”.

La Guardia Municipal de Transporte el año 2014 a través de las denuncias recibidas por los diferentes canales habilitados, realizó el operativo de captura de placas a los vehículos cuyas rutas prestan servicio en diferentes sectores de la ciudad de La Paz, las placas decomisadas son devueltas a los conductores previo pago de las infracciones emitidas por la Guardia Municipal de Transporte por trameo, alteracion de ruta, incremento de tarifas y otros (GAMLP, 2015).

Según GAMLP (2015) para detectar irregularidades en el transporte público el 17 abril de 2014 el Gobierno Autónomo Municipal de La Paz a través de la Guardia Municipal de Transporte iba realizando los operativos de control de las paradas en los sectores de San Antonio, San Antonio, Cotahuma, Max Paredes, Centro y Mallas hasta el final del recorrido de las rutas observadas. Debido a la existencia de rutas fantasmas, este control se realizó con la Ordenanza Municipal OM/109/2010 de rutas y recorridos del Servicio de Pasajeros. La Guardia Municipal de Transporte recorrió la ciudad haciendo un relevamiento de paradas llenando de formularios, para un cambio del transporte público libre, eficiente y sin

Según Tapia (2016) el trameo consiste en la ruptura de rutas y recorridos con un cobro adicional por parte de los transportistas, es considerado como una infracción gravísima del transporte de pasajeros. Así lo establece la Ley para el Control de Servicio de Transporte que fue aprobada por el Concejo Municipal de La Paz.

La norma es complementaria a la ley Municipal de Transporte y Tránsito Urbano, cuyas incorporaciones tienen origen en las preocupaciones que expresaron representaciones vecinales respecto al transporte público, se sancionara como infracción gravísima el trameaje, modificar las rutas y los recorridos, así como aquellos vehículos que no cumplan los parámetros técnicos y de calidad (Tapia, 2016).



En la actualidad se encuentra vigente la Ordenanza Municipal 109/2010 de 26 de abril de 2010, misma que aprobó 540 rutas del servicio público de transporte colectivo de pasajeros, sin embargo el municipio considera necesario actualizar las rutas y recorridos del servicio público de transporte colectivo de pasajeros para satisfacer las necesidades de desplazamiento de la población en general (Calvo, 2016).

Según AMN (2016) en la ciudad de La Paz existen 34 puntos de trameaje, la mayor cantidad se ubica en Mallasa y la zona Sur, según datos manejados por la Alcaldía paceña. En Mallasa y la zona Sur se encuentran los puntos donde los choferes incurren en el trameaje. En el centro existen 10 puntos, de los cuales se fueron identificados cinco y en la Periférica, Cotahuma y Maipú existen 19 puntos en total.

A continuación se presentan algunos trabajos de investigación que se relacionan de alguna manera con el trabajo que se está realizando en la biblioteca de la Carrera de Informática (FCPN-UIBAP).

Título: "Sistema de Información Geográfica para las administraciones de internet, caso Telfo"

Autor: Fernando

Año: 2013

Resumen: Este proyecto de grado fue desarrollado para la empresa TELFO "INGENIERIA", que se dedica a la instalación de internet en diferentes empresas y ciudades. El objetivo del proyecto es realizar un Sistema de Información Geográfica que permita administrar las conexiones de internet que la empresa realiza dando un mapa de las ubicaciones de las conexiones de internet que la empresa realiza.

Título: "Sistema de Información Geográfica para el control de rutas del transporte público vehicular de la ciudad del Alto, vía móvil"

Autor: Fredy López Gómez

Año: 2014

Resumen: En este proyecto de grado se implementó un sistema de información con todas las rutas de la ciudad del Alto, la búsqueda de rutas se realiza mediante zona y número de línea del vehículo, la forma en que se muestra la ruta son los puntos más referentes de la misma, los cuales son consultados a través de



telefonía móvil, esto con el fin de ayudar a controlar el recorrido por tramos del transporte público vehicular de la ciudad del Alto.

Título: "Software Movil de Geolocalizacion para la Banca en la Ciudad de La Paz"

Autor: Alex Perez Corimayta

Año: 2014

Resumen: En esta tesis se desarrolló un software centrado en un servicio de georeferenciación con respecto a la ubicación de servicios bancarios en la ciudad de La Paz, este software tiene como finalidad proporcionar información confiable y precisa sobre la ubicación y la ruta hacia el punto de destino de una entidad financiera usando tecnología de georeferenciación y de geolocalización en un dispositivo móvil.

1.2.2 LISTA DE PROBLEMAS

Para el estudio del problema se realizó un análisis de la situación actual identificándose diversos problemas de cumplimiento de rutas por parte del transporte público. En la continuación se detallan los problemas identificados.

- Falta de control de rutas por parte de los conductores.
- Vehículos del transporte público que no cumplen con las rutas.
- Cobro por tramo por parte de los conductores.
- Líneas de transporte público que no cumplen con la frecuencia habitual de vehículos.
- Cambian de ruta en el momento de la salida para la posibilidad de mayor número de pasajeros.

1.2.3 FORMULACIÓN DEL PROBLEMA

¿La aplicación móvil basada en geolocalización para el control de tramos del transporte público, será capaz de cooperar al usuario en el control de tramos del transporte público denominado "trameaje", utilizando códigos QR y Geolocalización?

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar un modelo de control de tramos del transporte público para dispositivo móvil, que permita al usuario conocer el recorrido de un vehículo del transporte público mediante código QR, verificando la localización del usuario en la ruta.

1.3.2 Objetivos Específicos

- Obtener información acerca de las líneas de transporte público.
- Plantear el modelo de control de tramos con la información adquirida.
- Diseñar una aplicación Android basada en el modelo de control de tramos.
- Implementar un algoritmo para generar códigos QR con una información única.
- Desarrollar un lector de QR en la aplicación móvil.
- Elaborar el sistema de control de ruta utilizando geolocalización.

1.4. HIPÓTESIS

El modelo de control de tramos de transporte público aplicando códigos QR y Geolocalización, colaborando con las autoridades, denominado “trameaje” para el cumplimiento de ruta de vehículos.

1.5. JUSTIFICACIÓN

1.5.1 Justificación

Debido al transporte público que se realiza en la ciudad de La Paz, el incumplimiento de rutas, por parte de algunos choferes de vehículos, las autoridades, para evitar arbitrariedades en el servicio, mientras tanto que el municipio de La Paz prepara una ley para solucionar esta situación. En consecuencia lo que se busca con este proyecto beneficie a la sociedad brindándole información sobre la ruta que debe seguir una determinada línea de transporte vehicular e identificar los puntos de control en la misma, permitiendo decidir al usuario proceder con la denuncia en caso de incumplimiento de ruta.

1.5.2 Justificación Económica

Los usuarios del transporte público deben tomar entre dos y hasta tres vehículos para llegar a su destino, debido a que los transportistas no cumplen las rutas establecidas, erogando de esta manera más de lo debido en el uso de transporte público por parte de los usuarios, es la razón del desarrollo del presente modelo que pretende aminorar los denominados “trameajes” colaborando al pasajero. Por otra parte el costo del desarrollo del modelo de control de tramos será mínimo, se necesitara la creación e impresión de códigos QR y el desarrollo de la aplicación móvil basado en software libre, por lo que el desarrollo del modelo es económicamente factible.



1.5.3 Justificación Tecnológica

El uso de dispositivos móviles ha ido incrementándose a través de los años, hoy por hoy con todas las utilidades que nos proporcionan, nos obligan a no apartarnos de ellos, por lo cual recurrimos a tecnologías como los códigos QR, Servicios de Google, Geolocalización que serán aplicados en el desarrollo de una aplicación móvil codificada en Android Studio que formara parte del modelo de control de tramos.

Gracias a la accesibilidad de los dispositivos móviles y la tecnología que se aplicara, se realizara el modelo de control de tramos de forma útil para el usuario ofreciéndole usabilidad, accesibilidad y rapidez en tiempo real.

1.5.4 Justificación Científica

El desarrollo de aplicaciones para dispositivos operativos como Android, está cobrando mayor importancia debido a la demanda de dispositivos móviles de las diferentes plataformas como Android, iOS, etc. Por tal motivo se pensó en explotar las características de los dispositivos móviles a través del desarrollo del modelo de control de tramos de transporte público, utilizando tecnologías, siendo así este un instrumento alternativo para brindar información acerca de las rutas del transporte público, aprovechando el avance de la informática en el área de sistemas.

1.6. ALCANCE

La aplicación móvil del modelo de control de tramos desarrollada en la plataforma Android, se encargara del reconocimiento de los códigos QR, los cuales contendrán información del conductor, vehículo y ruta del transporte público, estos serán impresos en papel adhesivo, y se colocaran en las ventanas de los minibuses del transporte público, posteriormente se adquirirá la información que estos contengan por medio de la cámara del dispositivo móvil, para luego mostrarnos la ubicación en la ruta de una determinada línea de minibús transporte público.

Esto se llevara a cabo si el usuario tiene acceso a internet ya que el modelo de control de tramos mostrara la información acerca de la ubicación del minibús que recorre una determinada ruta en tiempo real.



El modelo de control de tramos abarcara solamente minibuses del transporte público de una línea determinada de la ciudad de La Paz.

1.7. METODOLOGÍA

La metodología que se utilizara para la presente tesis es el método científico esto debido a las etapas que esta presenta y que son necesarias como:

- Observación
- Formulación de hipótesis
- Experimentación
- Emisión de conclusiones

1.7.1 Observación

Cuando se encuentra un hecho interesante lo primero que hace es observarlo con atención.

La Observación consiste en examinar los hechos y fenómenos que tienen lugar en la naturaleza por los sentidos.

1.7.2 Formulación de hipótesis

Después de las observaciones se plantea el cómo y el porqué de lo que ha ocurrido y formula una hipótesis.

Formular una hipótesis es elaborar una explicación provisional de los hechos observados y de sus posibles causas.

1.7.3 Experimentación

Una vez formulada la hipótesis se comprueba si es cierta. Para ello realizará múltiples experimentos modificando las variables que intervienen en el proceso y comprobará si se cumple su hipótesis.

Experimentar consiste en reproducir y observar varias veces el hecho o fenómeno que se quiere estudiar, modificando las circunstancias que se consideren convenientes.



1.7.4 Emisión de conclusiones

La emisión de conclusiones consiste en la interpretación de los hechos observados de acuerdo con los datos experimentales.

Por otra parte para el tipo de investigación que se utilizara será la cualitativa esto porque la presente investigación es un tipo de investigación aplicada, destinada a encontrar soluciones a problemas que tenga un grupo, una comunidad, una organización.

La metodología que se utilizará será **Mobile-D** por las características fundamentales del método.

El objetivo de este método es el desarrollo de productos de desarrollo muy rápidos en equipos muy pequeños. Este método fue desarrollado por un proyecto finlandés en 2005, pero sigue estando vigente. Las etapas de este método son las mismas pero aplicadas de forma estricta como: extremo de desarrollo, desarrollo iterativo y Rational Unified Process.

Se compone de las siguientes etapas: fase de planificación, fase de producto, fase de estabilización y la fase de entrega. Cada etapa tiene un día de planificación y otro de entrega.



Capítulo 2.

MARCO TEÓRICO

2.1 PLATAFORMA ANDROID

Android es un sistema operativo móvil basado en el kernel de Linux, con una interfaz de programación Java, diseñado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros. Desarrollado por la Open Handset Alliance la cual es formada por Google, Android permite programar aplicaciones en una variedad de lenguajes de programación, la Dalvik. El sistema operativo proporciona todas las herramientas necesarias para desarrollar aplicaciones que accedan a las funciones de los dispositivos como llamadas, sms, agenda, entre otras, de una forma sencilla y accesible. La información muy popular como es Java, en la figura 2.1 (Google, 2013).



Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de software y operadores de servicio. Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 700.000 aplicaciones de las cuales, dos tercios son gratuitas disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android, como pueden ser la App Store de Amazon o la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software

externamente. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto (Pedrozo, 2012).

2.1.1 Características

Según Aranaz (2012) Android es un sistema para dispositivos móviles que contiene una pila de software que incluye un sistema operativo, *middleware* y aplicaciones básicas. Entre otras, cuenta entre otras, con las siguientes características:

- Busca el desarrollo de aplicaciones que sean reutilizables y verdaderamente portables.
- Los componentes de software pueden sustituir fácilmente por otros.
- Cuenta con su propio *Dalvik* que interpreta y ejecuta código escrito en Java.
- Permite la representación 2D y 3D.
- Posibilita el uso de bases de datos.
- Soporta un elevado número de formatos multimedia.
- Servicio de localización.
- Controla los diferentes dispositivos: Bluetooth, Wi-Fi, cámara fotográfica o de vídeo, GPS, acelerómetro, micrófonos, etc., siempre y cuando el dispositivo móvil lo contemple.
- Cuenta con un entorno de desarrollo muy cuidado mediante un SDK disponible de forma gratuita.
- Ofrece un *plug-in* para uno de los entornos de desarrollo más populares, Eclipse, y un emulador integrado para ejecutar las aplicaciones.



2.1.2 Arquitectura

Según Pedrozo (2012) los componentes principales del sistema operativo de Android se describen continuación a detalle:

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs que se encuentran usados por las aplicaciones base. La arquitectura está diseñada para permitir la reutilización de componentes; cualquier aplicación puede luego hacer uso de las capacidades y cualquier otra aplicación puede ser reemplazada (sujeto a reglas de seguridad del framework). Esto permite que los componentes sean reemplazados.
- **Bibliotecas:** las bibliotecas de C/C++ usadas por varios componentes del sistema se exponen a los desarrolladores de aplicaciones de Android; algunas son: biblioteca C estándar), bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** un set de bibliotecas base que proporcionan la mayoría de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación de Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr muchas máquinas virtuales de forma eficiente. Dalvik ejecuta archivos Dalvik Executable (.dex), el cual está optimizado para memoria. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

En la figura 2.2 se muestra arquitectura de Android, donde se observa una visión global por capas de cuál cada una de estas capas utiliza servicios ofrecidos por las anteriores, y ofrece a su vez los suyos propios a las capas de niveles superiores.

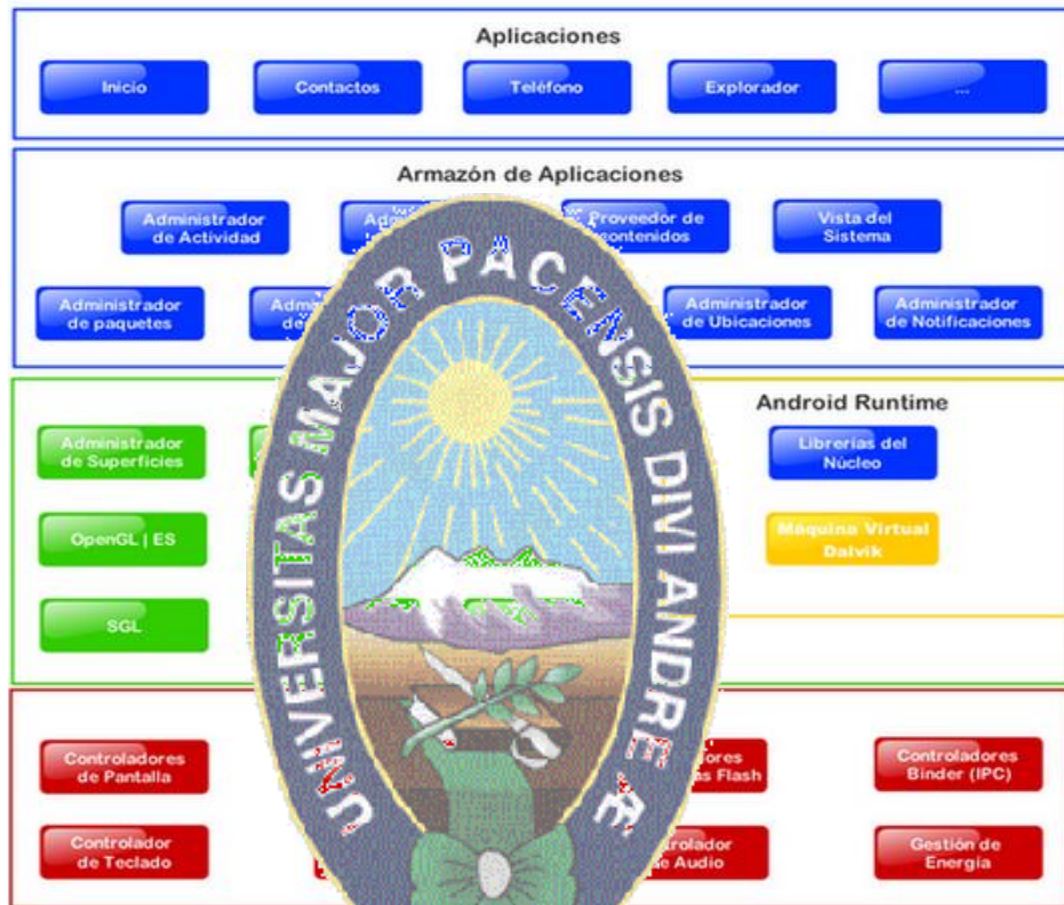


Figura 2.2. Arquitectura de Android (González, 2009)

2.1.3 La máquina virtual

En Android, todas las aplicaciones se programan en el lenguaje Java y se ejecutan mediante una máquina virtual de nombre Dalvik, específicamente diseñada para Android. Esta máquina virtual ha sido optimizada y adaptada a las peculiaridades propias de los dispositivos móviles es decir menor capacidad de proceso, baja memoria, alimentación por batería, etc. y trabaja con ficheros de extensión *.dex* (*Dalvik Executables*). Gracias a la herramienta “dx”, esta transformación es posible: los ficheros *.class* de Java se compilan en ficheros *.dex*, de forma que cada fichero *.dex* puede contener varias clases. Después, este resultado se comprime en un único archivo de

extensión .apk. *Android Package* que es el que se distribuirá en el dispositivo móvil. Según los responsables del proyecto, la utilización de esta máquina virtual responde a un deseo de mejorar y optimizar la ejecución de aplicaciones en dispositivos móviles, así como evitar la fragmentación de otras plataformas como Java ME (Tapia, 2013).

2.2 CÓDIGO QR “QUICK RESPONSE CODE”

El **código QR** del inglés *Quick Response code*, "código de respuesta rápida" es un módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional. Fue creado en 1994 por la compañía japonesa Denso Wave, subsidiaria de Toyota. Presenta tres características que permiten detectar la posición del código al lector. El objetivo del equipo de dos personas en Denso Wave, fue que el código QR se leyera a alta velocidad. Los códigos QR son muy comunes y el código bidimensional más popular (Wikipedia, 2015).

2.2.1 Características

Aunque inicialmente se usó en el área de la fabricación de vehículos, hoy los códigos QR se utilizan en una gran variedad de industrias. La administración de inventarios en una tienda que lee códigos QR en teléfonos móviles ha permitido no solo al consumidor, que se manifiestan en comodidades como el de introducir datos de forma manual en los teléfonos. Las direcciones y anuncios volviendo cada vez más comunes en revistas y anuncios. El agregado de códigos QR en tarjetas de presentación también se está haciendo común, y permite reducir en medida la tarea de introducir detalles individuales del nuevo cliente en el teléfono móvil.

Los códigos QR también se pueden leer desde computadores personales, teléfonos inteligentes o tabletas mediante dispositivos de captura de imagen como escáneres o cámaras de fotos, programas que lean los datos QR y una conexión a Internet para las direcciones web.

El estándar japonés para códigos QR (JIS X 0510) se publicó en enero de 1998 y su correspondiente estándar internacional ISO (ISO/IEC18004) se aprobó en junio de 2000, este documento explica a fondo cómo funcionan los Códigos QR y sus características y en los años siguientes sacan varias revisiones.

Un detalle importante sobre el código QR es que, a diferencia de otros formatos de códigos de barras bidimensionales como el BIDI, su código es abierto y sus derechos de patente de propiedad de Denso Wave no se ejercen (Wikipedia, 2015).

2.2.2 Conceptos básicos de una estructura de código QR

La representación bidimensional de un código QR se denomina símbolo. Cada símbolo está formado por cuadros negros o blancos llamados módulos, que representan el 0 y el 1 binario respectivamente. En la figura 2.3 se presenta los módulos que están ubicados en una estructura que se divide en dos grandes bloques de módulos: los patrones de función y la región de codificación. En cada símbolo existen un conjunto de módulos que no contienen información necesaria para su decodificación, son los patrones de función, y existen de varios tipos:



Figura 2.3 Estructura Genérica del Símbolo QR
Fuente. (Luke, 2015)

- Patrón de localización: patrón de función que existe por triplicado en el símbolo, situado en las esquinas superiores y la inferior izquierda. Sirven para calcular la orientación rotacional del símbolo.
- Patrón de alineamiento: secuencia alternada de módulos blancos y negras que ayuda a calcular las coordenadas de los módulos del símbolo.

- Patrón temporizador: patrón de función que permite resincronizar las coordenadas de mapeo del símbolo ante posibles distorsiones moderadas.
- Separador: patrón de función formado por módulos blancos, cuyo ancho es de un módulo y que separa los patrones localizadores del resto del símbolo.

Los datos codificados, por su parte, se agrupan en conjuntos de 8, denominados *codewords*, que adoptan diversas formas según su ubicación en la estructura. La región de codificación es la región del símbolo no ocupada por patrones de función y sí por *codewords* de datos y de corrección de errores, así como por la información de formato y versión. La información de versión está codificada que contiene información sobre el grado de corrección de errores que se han codificado los datos de la región de codificación y el tamaño del símbolo que se ha aplicado. La información de versión, por su parte, también está codificada que contiene información que indica la versión del símbolo.

Asimismo, para facilitar la lectura de los bordes de cada símbolo, se requiere de una banda de silencio de 4 módulos de ancho que debe estar en blanco y en negro si el símbolo está en blanco y negro.

El tamaño de los símbolos QR varía, véase figura 2.4, existen 40 versiones: la versión 1 tiene 21x21 módulos, la versión 2 tiene 25x25 módulos, y así sucesivamente existe incrementando el tamaño en cada versión hasta la versión 40, que contiene 177x177 módulos. Cada versión tiene un número y disposición diferente de *codewords* de datos y corrección de errores, así como de patrones de función.

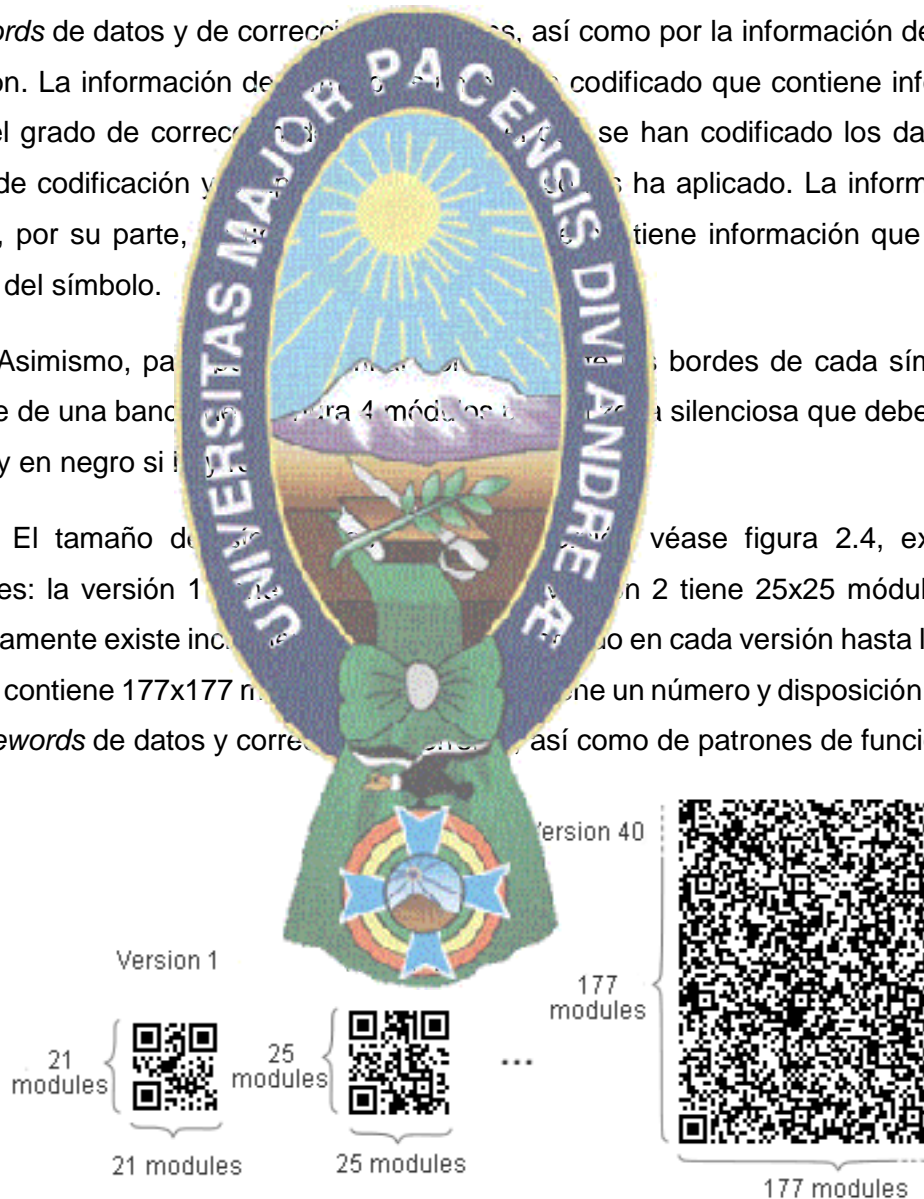


Figura 2.4 Versiones de los Códigos QR
Fuente. (QRCode, 2011)

Así, algunas versiones necesitan de módulos (bits) de relleno y otras no. Las versiones inferiores a la 7 no disponen de información de versión, y la versión 1 no incluye patrón de alineamiento. Sin embargo todas las versiones tienen 3 patrones localizadores, 2 patrones temporizadores, 3 separadores y la formación de formato por duplicado (Luke, 2015).

2.2.3 Corrección de errores

Según Luke (2015) los códigos QR emplean codificación de errores basada en algoritmos de *Reed-Solomon* para el conjunto de *codewords* de corrección de errores (ECC, *Error Correction*). Estos se añaden a los de datos aportando redundancia. Los algoritmos de Reed-Solomon fueron desarrollados inicialmente para combatir el ruido de comunicación en sondas artificiales y sondas espaciales, y hoy día se emplean por ejemplo en la producción de CDs de música. Permiten corrección a nivel de *bits* de ráfaga. Existen 4 niveles de corrección de errores:

- L (Low).** Puede corregir el 7% de los *bits* de datos del símbolo.
- M (Medium).** Puede corregir el 15% de los *bits* de datos del símbolo.
- Q (Quality).** Puede corregir el 25% de los *bits* de datos del símbolo.
- H (High).** Puede corregir el 30% de los *bits* de datos del símbolo.

2.2.4 Área de impresión

Los códigos QR, requieren un área de impresión mucho más reducida como se muestra en la figura 2.5, bien sea por el tamaño de impresión y facilitando su lectura. Son capaces de almacenar hasta 7086 *bits* de datos que un código de barras convencional en un tamaño 10 veces mayor puede almacenar aproximadamente. Incluso, otra versión de código QR es más pequeña a un tamaño similar al código de barras como Micro QR (QRCode, 2011).



Figura 2.5 Comparativa espacio de impresión código de barras vs. QR
Fuente. (QRCode, 2011)

2.2.5 Alta capacidad de codificación de los datos

Mientras que los códigos de barras convencionales son capaces de almacenar un máximo de aproximadamente 20 dígitos, QR es capaz de manejar varias decenas a varios cientos de veces más información ver figura 2.6.

En la tabla 2.1 puede observarse las capacidades máximas de datos de código QR, El código QR es capaz de manejar todos los tipos de datos, tales como caracteres alfabéticos y numéricos, códigos de Kanji, Kana, Hiragana, los símbolos, los binarios y de control. Hasta 7.089 caracteres pueden ser codificados en un símbolo (QRCode, 2011).

Tabla 2.1 Capacidades máximas de datos de Código QR

Capacidad	
Numérico	9780 caracteres
Alfanumérico	4296 caracteres
Binario	11163 bytes
Kanji/Hiragana	2197 caracteres

abcdefghijklmnopqrstuvwxyz
 klmnopqrstuvwxyz1234567890
 tuvwxz1234567890abcdefghijklmnopqrstuvwxyz
 34567890abcdefghijklmnopqrstuvwxyz1234
 klmnopqrstuvwxyz1234567890a
 tuvwxz1234567890abcdefghijklmnopqrstuvwxyz



Figura 2.6 Vista general del Código QR
 Fuente. (QR Code, 2011)

2.2.6 Resistentes a suciedad o daño parcial

Los códigos QR tienen incorporadas funciones para corrección de errores. Los datos pueden ser recuperados incluso si el código se encuentra parcialmente dañado o la suciedad presente no permite su lectura total la figura 2.7 puede verse lo mencionado.

Un máximo de 30% de codewords pueden ser recuperadas. Un codeword es una unidad que construye el área de datos en el caso de QR, es igual a 8 bits (QRCode, 2011).



Figura 2.7 Diferencia de posición en un código QR impreso (QRCode, 2011)

2.2.7 Legibles desde cualquier ángulo

Los códigos QR necesitan una funcionalidad de rápida respuesta. Esto es posible gracias a los patrones de posición presentes en 3 de las esquinas que conforman al código.



Figura 2.8 Patrón de detección de posición (QRCode, 2011)

Estos patrones garantizan lectura del código disminuyendo los efectos negativos de la posible interferencia (QRCode, 2011)

2.2.8 Obteniendo datos de un código QR

El proceso de escanear, decodificar y leer el contenido de un código como el QR utilizando la cámara de un teléfono se conoce como “mobile tagging” ó “etiquetado móvil” traducido al español. Para hacer uso de esta funcionalidad en el teléfono provisto de cámara, es necesario contar con un software que decodifica la imagen, el proceso de etiquetado móvil se observa en la figura 2.9.

Existen diferentes en el mercado y la mayoría son gratuitos para la gran variedad de dispositivos: Nokia, Motorola, Apple, etc.

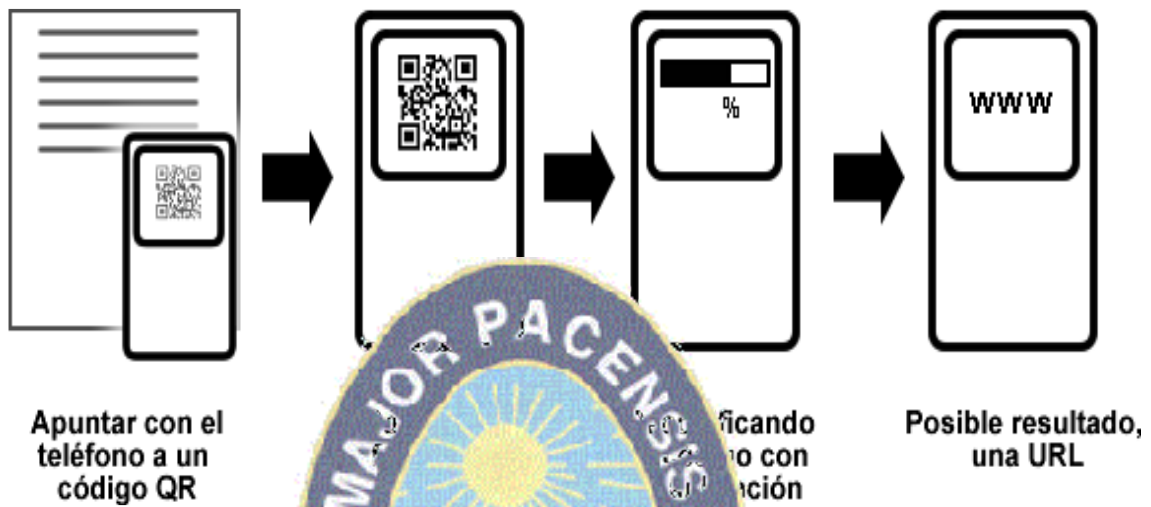


Figura 2.10: Etiqueta de código QR “etiquetado móvil”

Una vez que se toma una fotografía de un código QR con el teléfono celular, el software procesa la imagen y permite determinar una dirección URL, que puede ser una dirección de la industria) que mayor impacto tiene en el uso de tecnología (Bonda, 2014).

2.3 MOBILE D

El desarrollo de aplicaciones móviles enfrenta prácticamente los mismos problemas que la gran mayoría de desarrollos de software. Aunque hay que tener en cuenta sus principales peculiaridades como la fragmentación de sus desarrollos, la gran competencia del sector que obliga a constante innovación, los cambios frecuentes en la plataforma de desarrollo y la simplicidad de algunas aplicaciones. Todo ello influye a la hora de elegir una tecnología concreta de desarrollo.

El objetivo de este método es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: Xtreme Programming, CrystalMethodologies y RationalUnifiedProcess.

El ciclo de desarrollo de Mobile-D se muestra en la figura 2.10, se compone de distintas fases: exploración, inicialización, fase de producto, fase de estabilización y la fase de pruebas. Cada una tiene un día de planificación y otro de entrega.



Mobile D

2.3.1 Exploración

Siendo ligeramente anterior al ciclo de producción, se dedica al establecimiento de un entorno de desarrollo de software ágiles. Por lo tanto, se puede separar del ciclo principal de desarrollo (ya obviarse). Los autores de la metodología ponen acento en la participación de los clientes en esta fase (Koslela, 2004).

Los objetivos de establecimiento son:

- Identificar al cliente para el producto.
- Lograr el compromiso de los stakeholders para la participación del proyecto.
- Definir el modo, tareas y roles del grupo de cliente.

2.3.2 Iniciación

Los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico (incluyendo el entrenamiento del equipo de desarrollo). Los autores de Mobile-D afirman que su contribución al desarrollo ágil se centra fundamentalmente, en la investigación de la línea arquitectónica. Esta acción lleva a cabo durante el día de planificación, se agregan las observaciones, se identifican similitudes y se extraen soluciones viables para su aplicación en el proyecto. Finalmente, la metodología también contempla

algunas funcionalidades nucleares que se desarrollan en esta fase, durante el día de trabajo (Koskela, 2004).

Los objetivos de la fase de iniciación patrón son los siguientes:

- Adquirir un buen conocimiento general del producto para el equipo del proyecto en los requisitos iniciales y descripciones de línea de arquitectura.
- Preparar los recursos físicos, técnicos y humanos, así como de los clientes comunicación, los planes del proyecto y todas las cuestiones fundamentales de desarrollo para que el equipo estén en plena preparación para la implementación de los requisitos seleccionados por el cliente durante las próximas fases de desarrollo.

Los criterios de ingreso a la fase de desarrollo son:

- La fase de exploración de requisitos ha concluido.

El patrón de fase de desarrollo puede llevarse a cabo a través de las siguientes etapas: Proyecto de desarrollo, pruebas de aceptación y pruebas de integración.

2.3.3 Producto

Se repite la programación (se repite la liberación trabajo liberación) se repite iterativamente hasta implementar las funcionalidades. Primero se planifica la iteración de trabajo en términos de tareas a realizar. Se preparan las pruebas de la iteración de antemano. El nombre de esta técnica de Test Driven Development, TDD. Las tareas se llevan a cabo durante el día de trabajo, desarrollando e integrando el código en los repositorios existentes. Durante el último día se lleva a cabo la integración de los equipos en caso de que estuvieran trabajando varios equipos de forma independiente. Se realizan las pruebas de aceptación (Koskela, 2004).

Los objetivos son:

- Implementar la funcionalidad priorizada cliente para el producto.
- Se centran en la funcionalidad del núcleo fundamental de su ejecución a principios de incrementar para permitir múltiples ciclos de mejora.

Después 0 iteración, los criterios de ingreso:



- La fase de iteración precedente ha sido completado.
- Se han identificado los requisitos funcionales más importantes.
- El equipo se ha reunido y ha sido entrenado para el método de desarrollo.
- El entorno de desarrollo ha sido establecido.

Después de iteraciones posteriores, los criterios de ingreso:

- Precediendo productización iteración completa.

2.3.4 Estabilización

Se llevan a cabo las pruebas de integración para asegurar que el sistema completo funcione correctamente. Este es el paso más importante en los proyectos multi-equipo con diferentes equipos de desarrollo por equipos distintos. En esta fase, los desarrolladores se aseguran de que los componentes que debían desarrollar en la fase de productización se integran correctamente. El esfuerzo se dirige a la integración del sistema. Adicionalmente, se puede comenzar esta fase la producción de documentación (Koskela, 2004).

Los objetivos del modelo de estabilización:

- Finalizar la aplicación.
- Mejorar y garantizar la calidad.
- Finalizar la documentación.

Los criterios de ingreso:

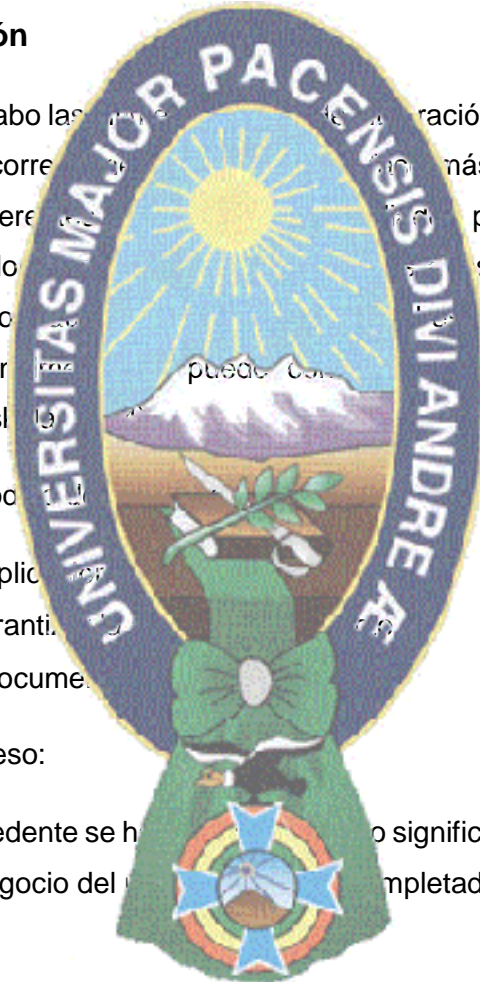
- La fase precedente se ha completado lo que significa que la funcionalidad esencial y valor de negocio del sistema se ha completado y se muestra al cliente.

2.3.5 Pruebas

Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados (Koskela, 2004).

Los objetivos del sistema de prueba y corrección con:

- Pruebe el sistema en base a la documentación presentada en el proyecto.



- Proporcionar información de los defectos encontrados.
- Deje que el equipo de proyecto para planificar solución para los defectos encontrados.
- Corregir los defectos.
- Producir como error de sistema libre como sea posible.

Los criterios de ingreso:

- Las fases anteriores necesitan ser terminados, por lo que la prueba puede llevarse a cabo con un sistema de seguimiento.
- Roles definidos por el sistema.

2.4 GEOLOCALIZACIÓN

La geolocalización es el proceso de asignar coordenadas geográficas a la información por medio de un sistema de seguimiento. La generalización de la tecnología GPS en dispositivos móviles y ordenadores personales ha permitido que cualquier persona, con un teléfono móvil y un ordenador personal, pueda acceder a la información de cualquier ciudadano, y como consecuencia, a los datos de los campos (Rodríguez, 2014).

Inicialmente se partirá de los conceptos de la geolocalización:

La geolocalización es el proceso de asignar coordenadas geográficas a la información por medio de un sistema de seguimiento. Este proceso es utilizado por los Sistemas de Información Geográfica también conocidos en inglés GIS, estos sistemas se encargan de integrar hardware, software y datos geográficos

Existen tres elementos clave que intervienen en el proceso de geolocalización:

Hardware: dispositivo que adquiere los datos en la que se desarrolla el proceso de geolocalización. Puede tratarse de un ordenador de sobremesa, un ordenador portátil, un dispositivo móvil, un navegador GPS, una cámara de fotos, etc. En los casos en que la localización física del dispositivo interviene como un elemento clave del proceso, el dispositivo hardware incorpora los mecanismos necesarios para permitir dicha localización por ejemplo, un receptor GPS.

Software: programa que ejecuta el proceso de geolocalización según su implementación. Este software se ejecutará en la plataforma del dispositivo hardware, y

se apoyará en éste para llevar a cabo la búsqueda de información, la determinación de localizaciones geográficas, y la asociación de ambos elementos.

Conexión a Internet: que actúa como medio de obtención e intercambio de información y, en ocasiones, como sistema de almacenamiento y procesamiento de la misma. Excepcionalmente, pueden ejecutarse procesos de geolocalización sin utilizar una conexión a Internet (modo fuera de línea), cuando los datos necesarios se encuentren cargados con antelación en la memoria del dispositivo (Alcón, 2007).

2.4.1 Geolocalización, localización y GPS

La semejanza entre los términos **Geolocalización** y **Localización** puede dar lugar a confusiones, por lo que a continuación aclararemos sus diferencias.

Geolocalización: consiste en determinar la posición de un objeto en el espacio, este concepto se refiere a la posición que ocupa un objeto en el espacio y que se mide en términos de longitud (x), latitud (y) y altura (z). Este término es un nuevo concepto que ha prestado de nombre a Internet también conocido como georreferenciación.

Localización: es un término que se refiere a la acción de determinar el emplazamiento de algún objeto en un espacio geográfico pero no se trata de una definición en absoluto. Este término se utiliza como un medio de acercamiento a comprender el término.

GPS: El Sistema de Posicionamiento Global (GPS), es un sistema global de navegación por satélite que permite determinar en cualquier punto del mundo la posición de un objeto móvil, podemos alcanzar una precisión de unos pocos metros usando el GPS, pero lo habitual son unos pocos metros, midiendo la distancia entre un grupo de satélites a una posición cualquiera de la Tierra pueden calcular las coordenadas exactas de dicha posición. Los satélites actúan como puntos de referencia precisos sus posiciones se conocen en cada momento, En la práctica los receptores GPS reciben las señales que envían los satélites y las utiliza para determinar la posición del móvil. Empleando las señales de cuatro satélites el receptor GPS puede calcular la posición en el espacio tridimensional (X, Y, Z) y el tiempo de esta señal que tardara en viajar de un satélite orbitando a 20.200 km de la Tierra (Beltran, 2012).



2.5 TECNOLOGÍAS A UTILIZAR

En este punto se hará una descripción de las herramientas utilizadas en la realización de esta aplicación. Estas se expondrán siguiendo el orden en el cual nos hemos ido familiarizando con cada una de ellas.

2.5.1 Google Maps

Google Maps, durante una época llamado *Google Local*, es un servidor de aplicaciones de mapas en web desarrollado con la tecnología de Google.

En sus inicios, ofrecía imágenes de satélite de diversos países como de EEUU, Canadá, Reino Unido, Japón, etc. Además, ofrecía un complemento y ayuda al usuario que realiza búsquedas en Google Maps.

Google Maps fue anunciado oficialmente en el *Google Blog* el 8 de Febrero de 2005. Originariamente, Google Maps era compatible con los navegadores *Internet Explorer* y *Mozilla Firefox*. El soporte para los navegadores *Opera* y *Safari* se añadió el 25 de Febrero de ese mismo año. El software de Google Maps fue desarrollado por los ingenieros de Google antes de convertirse en parte de *Google Local*. Google Maps comenzó a ofrecer imágenes de satélite en abril de 2005 comenzó a ofrecer imágenes de satélite además, imágenes vía satélite de Google Earth. *Keyhole*, adquirida por Google. De esta manera, pueden verse imágenes de satélite de todo el planeta de mayor o menor resolución dependiendo de la zona. En las zonas de los núcleos urbanos.

Todas las imágenes de satélite de Google Maps son las mismas que las que podemos encontrar en otra famosa aplicación de Google llamada *Google Earth*.

Del mismo modo que Google Maps es una aplicación web desarrolladas por Google, para implementar Google Maps, se utiliza un lenguaje de ficheros *JavaScript*. Como el usuario puede mover el mapa, la información del mismo se descarga desde el servidor.

Cuando este busca un punto determinado, la ubicación está marcada por un indicador en forma de pin, el cual es una imagen *PNG* transparente sobre el mapa.

Para conseguir la conectividad sin sincronía con el servidor, y así proporcionar al usuario mayor interactividad con el mapa, mediante la realización de peticiones asíncronas a la red con *JavaScript* y *XMLHttpRequest*, se usa una técnica reciente, conocida como *AJAX*, sobre la que ofreceremos información más adelante.



Además, en Junio de 2005 se presentó la **API de Google Maps** que permite a los programadores crear sus propias aplicaciones web utilizando las imágenes y mapas de este servicio (Vasquez, 2013).

2.5.2 Google Maps API

En los últimos tiempos la creciente demanda por las aplicaciones de geolocalización y mapas ha dado lugar una gran oferta de APIs disponibles entre las que poder elegir según nuestras necesidades. Las funciones que ofrecen todas ellas son muy similares, en general ofrecen: modificación, soporte para marcadores, cálculo de distancias, lugar (Vasquez, 2012).

Google Maps API es una API que permite la visualización de Google Maps en tus aplicaciones web. El API proporciona unas determinadas herramientas para interactuar con los mapas y añadir contenido a los mismos a través de un lenguaje de programación. Llegar a crear aplicaciones con mapas de gran complejidad (Vasquez, 2012).

Para implantar esta API en tu aplicación Android, google te proporciona una clave "single Maps API key" que debes registrar en un sitio web de google. Para obtener esta clave se debe crear una cuenta de desarrollador en google, y la clave que se proporciona estará conectada a dicha cuenta (Vasquez, 2012).

Antes de empezar a trabajar con las APIs de Google es necesario realizar algunas tareas previas, debemos asegurarnos de que tenemos instalado el paquete correspondiente a la versión de Android que desarrollamos enriquecido con las APIs de Google. Estos paquetes se encuentran formalmente *Google APIs by Google*, *Android API x*, esto podemos descargarlo si es necesario desde Eclipse accediendo al *Android SDK manager*. Con todo esto ya tendríamos creado nuestro proyecto de Eclipse y estaríamos preparados para poder ejecutar aplicaciones en el emulador sobre la versión correcta de Android y las APIs necesarias de Google (Hernández, 2013).

Google ofrece dos tipos de documentación: Una de ellas, está diseñada para permitir al nuevo usuario, empezar rápidamente a experimentar y desarrollar sencillas aplicaciones con Google Maps API. El otro tipo de documentación que se ofrece, se trata de una guía de referencia completa y exhaustiva: *Google Maps API Reference*.

Recientemente esta documentación se ha reorganizado para ofrecer una información más conceptual y así focalizar las posibles discusiones en las distintas áreas dominantes (Alcón, 2007).

2.5.3 La Google Places API Web Service

Google Places API Web Service es un servicio que devuelve información sobre sitios, definidos dentro de esta API como establecimientos, ubicaciones geográficas o puntos de interés importantes, mediante el uso de solicitudes de HTTP.

2.5.3.1 Introducción de los servicios

Están disponibles las siguientes funcionalidades:

- **Búsquedas de sitios** que permiten obtener información sobre la base de la ubicación de un usuario.
- Las solicitudes de **datos de sitios** devuelven información más detallada sobre un sitio específico, tales como sus reseñas y fotos.
- **Agregado de sitios** que permite agregar nuevos datos de la base de datos de Google Places.
- **Fotos de sitios** que permiten obtener millones de fotos relacionadas con los sitios almacenados en la base de datos de Google Places.
- **Autocompletado de sitios** que permite usar para completar automáticamente el nombre o la dirección de un sitio que estás escribiendo.
- **Autocompletado de consultas geográficas** que permite usar para proporcionar un servicio de predicción de consultas geográficas basadas en texto, al devolver consultas sugeridas que coinciden con lo que estás escribiendo.

Se puede tener acceso a cada uno de los servicios como una solicitud de HTTP y estos devuelven una respuesta JSON o XML. Todas las solicitudes para un servicio de Places deben usar el protocolo https:// e incluir una clave de API.

Google Places API Web Service usa el id de sitio web para identificar de forma exclusiva un sitio. Para obtener detalles sobre el formato y el uso de este identificador

en toda la Google Places API Web Service y en otras API, consulta la documentación de la id de sitio (Google Developers, 2016).

2.5.4 Google Maps Directions API

Google Maps Directions API es un servicio que calcula indicaciones entre ubicaciones usando una solicitud HTTP.

Puedes buscar indicaciones para diferentes medios de transporte, incluido transporte, manejo, desplazamiento o en bicicleta. Las indicaciones pueden especificar sitios de origen, sitios de destino y waypoints ya sea en forma de cadenas de texto (p. ej., "Chicago, Illinois" o "Australia") o como coordenadas de latitud/longitud. La Directions API puede proporcionar indicaciones en múltiples partes mediante una serie de waypoints.

Este servicio puede usarse para calcular indicaciones para direcciones estáticas que se muestran en el contenido de la aplicación en un mapa; este servicio también puede usarse para proporcionar tiempo real a las entradas del usuario, por ejemplo, para mostrar indicaciones dinámicas por ejemplo, dentro de un elemento de la interfaz de usuario. Para obtener más información para el Servicio de indicaciones de la Google Maps API, consulte la documentación.

El cálculo de indicaciones consume tiempo y recursos. Siempre que sea posible, calcula direcciones con anticipación usando el servicio que se describe aquí y guarda tus resultados en un caché temporal de tu propio diseño (Google Developers, 2016).

2.5.4.1 Solicitudes de indicaciones

Una solicitud de Google Maps Directions API debe respetar la siguiente forma:

<https://maps.googleapis.com/maps/api/directions/output?parameters>

Donde output puede ser cualquiera de los siguientes valores:

- json indica el formato de salida en JavaScript Object Notation (JSON).
- xml indica el formato de salida como XML.

Para acceder a la Google Maps Directions API a través de HTTP, usa:



<http://maps.googleapis.com/maps/api/directions/output?parameters>

Se recomienda el uso de HTTPS para aplicaciones que incluyen en las solicitudes datos privados de los usuarios, como la ubicación de un usuario.

Las direcciones URL de la Google Maps Directions API están restringidas a aproximadamente 2000 caracteres, después de la codificación URL. Dado que algunas direcciones URL de Google Maps Directions API pueden contener muchas ubicaciones en la ruta de acceso, debes tener en cuenta este límite al construir tus direcciones URL (Google Developers, 2016).

2.5.5 Waypoints

Al calcular rutas con la Google Maps Directions API, también puedes especificar *waypoints* para indicar paradas en la ruta. Si especificas un modo de transporte en bicicleta, los waypoints te permiten calcular rutas que incluyen paradas en cada uno de los waypoints. En otro caso la ruta devuelta incluye

Los waypoints se especifican en la URL de la API y consisten en una o más direcciones o ubicaciones separadas por una barra vertical (|).

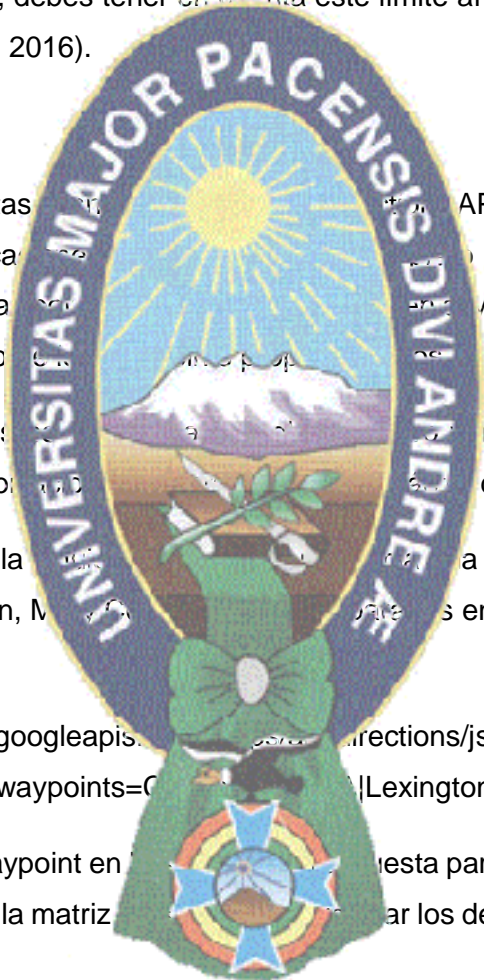
Por ejemplo, la siguiente es una solicitud de indicaciones para una ruta entre Boston, MA y Lexington, MA con paradas en Charlestown y Concord, en este orden:

```
https://maps.googleapis.com/maps/api/directions/json?origin=Boston,MA&destination=Concord,MA&waypoints=Charlestown,MA|Lexington,MA&key=YOUR_API_KEY
```

Para cada waypoint en la solicitud para las indicaciones incluye una entrada adicional en la matriz `legs` para proporcionar los detalles correspondientes a ese tramo del recorrido.

Si quieres alterar la ruta usando waypoints sin agregar una parada, al waypoint agrégale el prefijo `via:` Los waypoints con el prefijo `via:` no agregarán una entrada a la matriz `legs`, sino que, en su lugar, direccionarán el trayecto por el waypoint proporcionado.

La siguiente dirección URL modifica la solicitud anterior de modo que el trayecto se dirija por Lexington sin paradas:



https://maps.googleapis.com/maps/api/directions/json?origin=Boston,MA&destination=Concord,MA&waypoints=Charlestown,MA|via:Lexington,MA&key=YOUR_API_KEY.

El prefijo `via:` es más eficaz cuando se crean rutas en respuesta a que el usuario arrastra los waypoints en el mapa. Al hacer esto, el usuario puede ver cómo se verá la ruta final en tiempo real y lo ayuda a asegurarse de que los waypoints estén ubicados en sitios accesibles para la Google Maps Directions API (Google Developers, 2016).

2.5.5.1 Optimizar waypoints

De forma predeterminada, las aplicaciones calcula una ruta a través de los waypoints proporcionados en el orden en que se proporcionan. Como alternativa, puedes pasar `optimize:true` como el parámetro `optimize` en el parámetro `waypoints` para permitir que el servicio de información de mapas de Google reorganice la ruta proporcionada reorganizando los waypoints de manera que sea la más corta.

Si le indicas a las aplicaciones que respeten el orden de sus waypoints, ese orden se devolverá en el campo `waypoint_order` del objeto `routes`. El campo `waypoint_order` devuelve un índice de waypoint para cada waypoint.

El siguiente ejemplo muestra una ruta desde Adelaide, Australia del Sur, hasta cada una de las principales regiones de Australia del Sur usando la optimización de rutas.

https://maps.googleapis.com/maps/api/directions/json?origin=Adelaide,SA&destination=Adelaide,SA&waypoints=Adelaide,SA|Barossa+Valley,SA|Clare,SA|Connawarra,SA|McLaren+Vale,SA&key=YOUR_API_KEY (Google Developers, 2016).

2.5.6 Java Barcode Generation

Java Barcode Generation es una biblioteca generador de código de barras Java, el uso es para la generación de código de barras lineales y 2D en J2SE, J2EE, Java, esta biblioteca de clases de Java se proporciona para su integración en aplicaciones Java brindando imágenes con calidad con soporte para GIF, PNG, JPEG, BMP y sin distorsión alguna. La biblioteca se ofrece en un solo archivo JAR de Java que puede ser

utilizado en cualquier plataforma con una máquina virtual de Java, incluyendo Windows, Linux, Unix, MacOS, Solaris y HP / UX (Barcolib.com, 2015).

2.5.7 Librería ZXing

La librería ZXing es un proyecto open-source libre de derechos de autor para uso y modificación que ofrece soporte para lectura y decodificación para la gran mayoría de códigos de barras, códigos BIDI o QR en múltiples plataformas.

ZXing procesa imágenes multimedias en 1D/2D. Actualmente es capaz de reconocer los formatos UCC/EAN-13, Códigos 39, 93, 128, ITF, Codabar, RSS-14 en texto, imágenes de datos, Aztec, PDF 417 y por supuesto los populares códigos QR. Este hecho hace que la tecnología móvil esté presente en nuestras



Capítulo 3.

MARCO APLICATIVO

3.1 INTRODUCCIÓN

El modelo de control de tramos que se desarrollara tiene como fin principal ayudar al usuario del transporte público, a controlar que los vehículos sigan su ruta evitando de esta manera el denominado “trameo”. En este capítulo se detalla los elementos de la aplicación móvil definiendo los requisitos del modelo de control de tramos de forma teórica, seguidamente se relaciona el modelo mencionado con los requisitos funcionales conseguidos en los capítulos correspondientes.

La aplicación móvil debe cumplir con las tareas, por lo cual recurrimos a la metodología de desarrollo de aplicaciones móviles en equipos portátiles y rápidos.

Para cumplir con el desarrollo del modelo de control de tramos del transporte público es necesario aplicar la metodología Mobile-D las cuales son exploración, análisis, diseño, implementación y pruebas con el fin de cumplir con el objetivo del presente trabajo.

3.2 FASE DE EXPLORACIÓN

Esta fase se encarga de establecer el establecimiento del modelo de control de tramos, aquí es donde hacemos la definición del alcance del proyecto y su establecimiento con las funciones que queremos tener, dónde queremos llegar en el presente trabajo.

Así mismo es necesario especificar las distintas partes o grupos necesarios para la continuidad del modelo.

3.2.1. Establecimiento de los Usuarios

A continuación se presenta e identifica a los usuarios que pueden afectar o serán afectados por el desarrollo del modelo de control de tramos, la descripción de los usuarios se muestra en la tabla 3.1



Tabla 3.1 Descripción de roles de los usuarios

Usuario	Descripción	Stakeholder
Usuario final	Es la persona que tendrá instalado la aplicación móvil en su dispositivo móvil.	Usuario
Administrador	Persona que administrara la base de datos, este tendrá los permisos para generar los códigos QR para esto se recurrirá a la información de la base de datos como los usuarios, tablas como conductores, vehículos y líneas de transporte público.	Administrador

3.2.2 Requerimientos

Los requerimientos para el sistema de control de tramos serán los siguientes.

- Establecer una base de datos con la información de los conductores, vehículos y sindicatos.
- Con la información de la base de datos generar un símbolo de código QR.
- Diseñar el sistema de control de tramos en una línea de transporte público en particular.
- El lector de códigos QR debe ser capaz de reconocer, procesamiento y escaneo del símbolo QR en tiempo real en la aplicación móvil.
- Recurrir a la geolocalización con el propósito de controlar al vehículo en la ruta, de una línea de transporte público.
- La aplicación desarrollada debe ser accesible de manera automática a diferentes dispositivos móviles.

3.2.3 Arquitectura

La arquitectura del Modelo de Control de Tramos se estructura en la figura 3.1 que servirá para elaborar la planificación del funcionamiento del modelo de control de tramos de este modo podemos prever los puntos débiles y posibles problemas en el desarrollo del modelo.

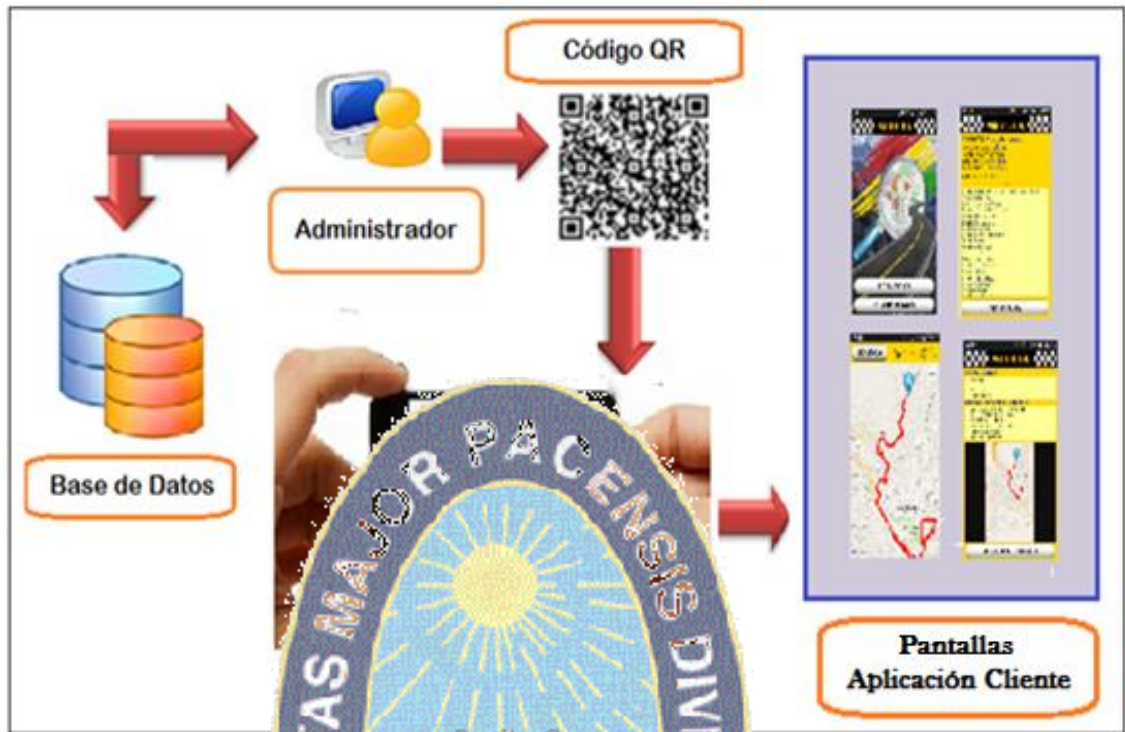


Figura 3.3. Arquitectura del Modelo de Control de Tramos

De lo descrito en el capítulo anterior se puede definir que requiere diferentes componentes el modelo de control de tramos. Para utilizar el administrador gestionara la información de base de datos de la base de datos referida a un número de línea vehicular. Esta información se almacenara en un código QR generado por el administrador. Para acceder a esta información el usuario debe contar con un dispositivo móvil con cámara. El usuario debe tener las aplicaciones necesarias para el reconocimiento del código QR de esta forma visualiza la información contenida en ella, para luego poder hacer uso de esta información y acceder a las diferentes ventanas de la aplicación cliente Mi Ruta, la cual será utilizada en el proceso de control y denuncia del vehículo de tránsito reportados por el usuario.

3.3 FASE INICIALIZACIÓN

Para seguir adelante con el desarrollo del modelo de control de tramos la siguiente fase a seguir, es la de Iniciación, que nos permite configurar el modelo identificando y preparando todos los recursos necesarios como se ha explicado en el anterior capítulo, en esta fase del modelo analizaremos y planificaremos todos los módulos que el sistema tendrá para su funcionalidad preparando los planes para las siguientes fases.

3.3.1 Establecimientos de recursos

En este punto se expone las herramientas necesarias para la implementación de la aplicación móvil.

- IDE Android estudio V 2.1
- SDK Manager
- Google Maps
- Eclipse IDE for Java Developers V Mars.2 Release (4.5.2)
- PostgreSQL V 9.1
- Dispositivo Móvil Samsung
- Computadora Portátil

3.3.2 Modelo de Entidad

Considerando los requisitos se elaboró el modelo Entidad-Relación que se muestra a continuación.

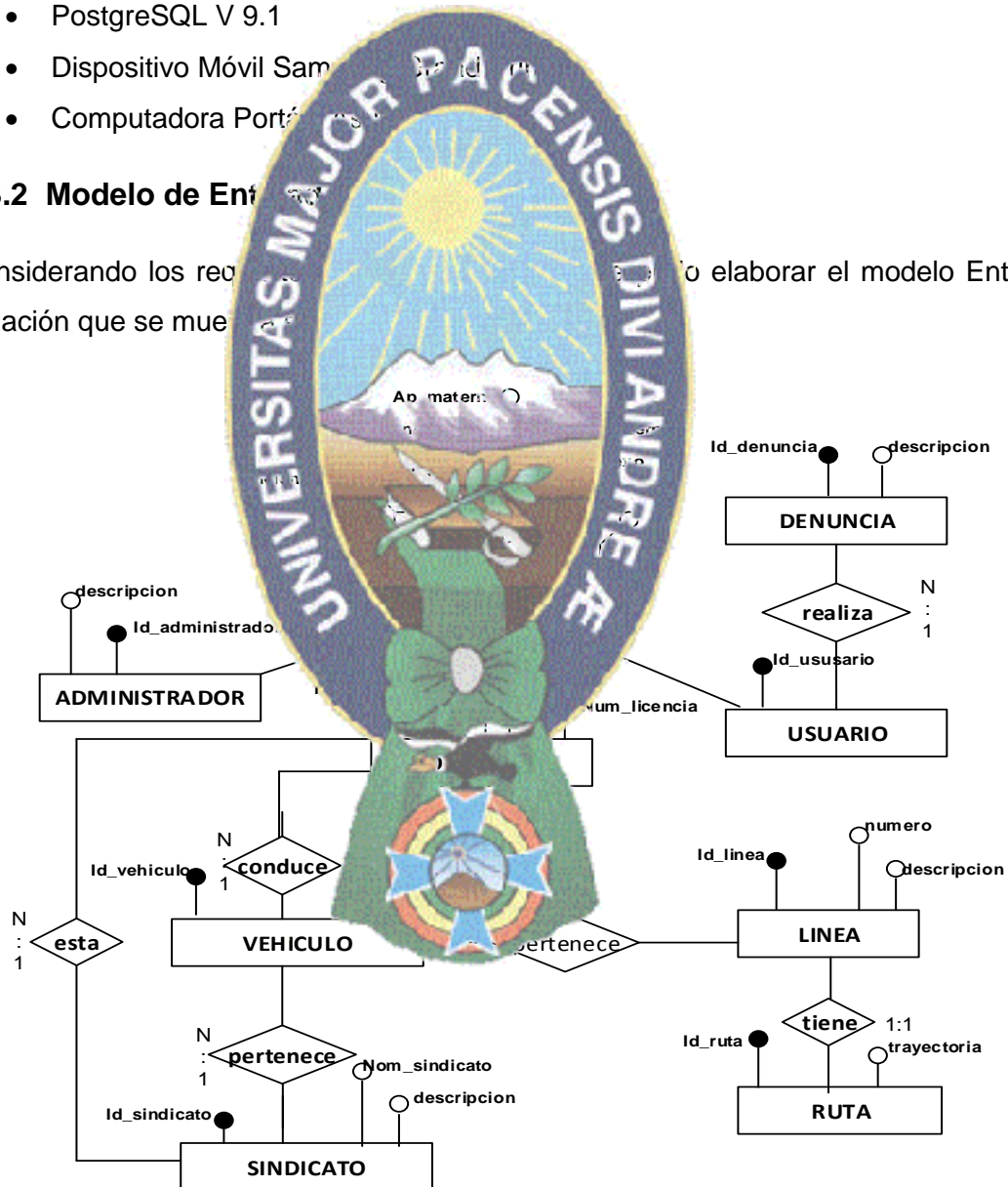


Figura 3.2 Modelo de entidad relación

Se puede ver que cada entidad cuenta con sus atributos respectivos para obtener la información necesaria, además de contar con su cardinalidad correspondiente.

3.3.3 Historias de usuario y tareas

Siguiendo con la fase de inicialización, es necesario realizar una recopilación e identificación clara de los requerimientos funcionales con los que se debe cumplir para realizar el desarrollo del modelo de software y así tener una visión general de lo que se desea crear, para esto se recurre a las historias de usuario los cuales expresaran los requisitos funcionales y las tareas que se deben realizar, cada una de estas tareas nos servirán para cumplir los requisitos de usuario.

En la tabla 3.2 se describe la tarea 1 Generar el Código QR donde se creara en base a la información de usuario.

Historia de Usuario	U1
Numero: 1	Generación de código QR
Modificación de Historia de Usuario	
Usuario : Administrador	Asignada: 1-2
Prioridad en negocio	Estimados : 3
Riesgo en Desarrollo	Reales : 5
Descripción:	
El sistema verifica que el conductor exista en la base de datos mediante el código del mismo, posteriormente se despacha el conductor, vehículo y sindicato para crear el código QR con toda la información necesaria.	
Observación :	

La siguiente tabla 3.3 describe la tarea 1 Diseño de la interfaz, la interfaz será desarrollado para uso del Administrador, necesario para crear y obtener los códigos QR.

Tabla 3.3 Tarea 1.1 Diseño de Interfaz para crear códigos QR

Tarea de Ingeniería	
Numero de Tarea: 1.1	Numero de Historia de Usuario: 1
Nombre de Tarea: Diseño de Interfaz	

Nombre de Tarea: Desarrollo Desarrollo/ Corrección / Mejora/otra	Puntos Estimado: 1
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio
Descripción: En esta pantalla existirá un botón con la opción “Generar QR” el cual una vez digitado codificara los datos seleccionados en un código QR.	

En la tabla 3.4 se describe la tarea de Ingeniería de Usuario 1.

Tarea de Ingeniería de Usuario 1: Generar una Imagen QR

Tarea de Ingeniería	Generar una Imagen QR
Numero de Tarea: 1	Nombre de Historia de Usuario: 1
Nombre de Tarea: Desarrollo/ Corrección / Mejora/otra	Puntos Estimado: 2
Fecha Inicio :	Programador Responsable: Juan Carlos Chuquimia Bustencio
Descripción: Para poder codificar la información seleccionada en un código QR y tener como resultado una imagen con el código QR, se debe integrar la librería Barcode.jar en la aplicación que usara el dispositivo móvil.	

En la tabla 3.5 muestra la historia de Usuario 2. Reconocer el símbolo QR con el propósito de mostrar la información de la librería.

Historia de Usuario 2

Historia de Usuario	
Numero: 2	Nombre: Reconocer el símbolo QR por medio de la cámara del dispositivo móvil.
Modificación de Historia de Usuario numero :	
Usuario : Usuario móvil	Iteración Asignada: 1-2
Prioridad en negocio : Alto	Puntos estimados : 3
Riesgo en Desarrollo : Medio	Puntos Reales : 5

Descripción: En esta instancia la aplicación móvil deberá escanear el código QR que contiene información codificada.
Observación :

La siguiente tabla 3.6 describe la Tarea 1 Diseño de la interfaz de la historia de usuario 2.

Tabla 3.6 Tarea 2.1 Diseño de Interfaz de la Ventana Principal

Tarea de Ingeniería	
Numero de Tarea: 2.1	Numero de Historia de Usuario: 2
Nombre de Tarea: Diseño de Interfaz de la Ventana Principal	
Nombre de Tarea: Desarrollo/ Corrección / Mejora/otro	Puntos Estimado: 2
Fecha Inicio :	
Programador Responsable :	Chuquimia Bustencio
Descripción: Esta es la ventana principal de la aplicación móvil, en esta ventana se encuentran dos botones una con la opción "IDENTIFICAR QR" y otra con la opción "REGISTRAR USUARIO".	

En la tabla 3.7 se describe la Tarea 2 Diseño de Interfaz de la Ventana de Historia de Usuario 2.

Tabla 3.7 Tarea 2.2 Diseño de Interfaz de la Ventana de Historia de Usuario 2

Tarea de Ingeniería	
Numero de Tarea : 2.2	Numero de Historia de Usuario: 2
Nombre de Tarea: Integrar Zonas de Interés	
Nombre de Tarea: Desarrollo/ Corrección / Mejora/otro	Puntos Estimado: 3
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio
Descripción: Esta librería nos permitirá procesar imágenes 2D, obteniendo y reconociendo de esta manera la información contenida en el código QR por medio de la cámara del dispositivo móvil.	

La tabla 3.8 describe la historia de usuario 3, Pantalla información obtenida del código QR.

Tabla 3.8 Historia de Usuario 3

Historia de Usuario	
Numero: 3	Nombre: Pantalla información obtenida del código QR
Modificación de Historia de Usuario número:	
Usuario : Usuario Móvil	Asignada:
Prioridad en negocio : Alto	Esfuerzos : 1
Riesgo en Desarrollo :	
Descripción: En esta pantalla se muestra información obtenida en el Código QR, además una lista de calles y avenidas con una línea vehicular.	
Observación :	

En la siguiente tabla se describe la tarea 4. Mostrar el mapa con la ruta trazada de una línea vehicular.

Historia de Usuario	
Numero: 4	Nombre: Mostrar el mapa con la ruta del número de línea vehicular
Modificación de Historia de Usuario número:	
Usuario : Usuario Móvil	Asignada: 3
Prioridad en negocio : Alto	Esfuerzos : 5
Riesgo en Desarrollo : Alto	Esfuerzos : 8
Descripción: En esta ventana se mostrara un mapa con las indicaciones de la ruta que debe seguir el vehículo del transporte público.	
Observación :	

La siguiente tabla 3.10 describe la tarea 1 de la historia de Usuario 4.

Tabla 3.10 Tarea 4.1 Diseño de la Interfaz para Mostrar el Mapa

Tarea de Ingeniería	
Numero de Tarea: 4.1	Numero de Historia de Usuario: 4
Nombre de Tarea: Diseño de Interfaz	
Nombre de Tarea: Desarrollo Desarrollo/ Corrección / Mejora/otra	Puntos Estimado: 2
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio
Descripción: En la ventana se mostrará el mapa de la zona de transporte público. El usuario debe seguir el número de una línea de transporte público. En caso de que el vehículo se encuentre en una zona de riesgo, se debe seleccionar la opción "DENUNCIA" en caso de que el vehículo se encuentre en una zona de riesgo.	

La tabla 3.11 se describe la tarea 4.2 de la historia de Usuario 4.

Tarea de Ingeniería	
Numero de Tarea: 4.2	Numero de Historia de Usuario: 4
Nombre de Tarea: Actualización de Interfaz	
Nombre de Tarea: Desarrollo/ Corrección	Puntos Estimado: 3
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio
Descripción: En la implementación será necesario utilizar los permisos de Google Maps, el cual nos proporciona una API que ayudara a mostrar el mapa en el proyecto de Android Studio.	

En la tabla 3.12 se describe la tarea 3 de la historia de Usuario 4.

Tabla 3.12 Tarea 4.3 Google Place API Web Service

Tarea de Ingeniería	
Numero de Tarea: 4.3	Numero de Historia de Usuario: 4
Nombre de Tarea: Api Key de Google Place API Web Service	

Nombre de Tarea: Desarrollo Desarrollo/ Corrección / Mejora/otra	Puntos Estimado: 3
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio
Descripción: Se requiere permisos de uso de Google Place API web Service, para obtener las peticiones de Google Maps Directions API	

En la siguiente tabla 3.13 se describe la historia de usuario 5. La Geolocalización parte importante para el Modelo de Usuario 5.

Historia de Usuario	
Numero: 5	Geolocalización
Modificación de Historia de Usuario:	
Usuario : Usuario M	Asignada: 4
Prioridad en negocio:	Estimados : 5
Riesgo en Desarrollo:	Requisitos : 2
Descripción: Para el control del vehículo de la línea de transporte, se requiere la Geolocalización, mostrando en cada momento la posición del vehículo que transita el vehículo de la línea vehicular.	
Observación :	

En la tabla 3.14 se describe la historia de usuario 5.

Tarea de Ingeniería	
Numero de Tarea: 5.1	Numero de Historia de Usuario: 5
Nombre de Tarea: Integración de GPS	
Nombre de Tarea: Desarrollo Desarrollo/ Corrección / Mejora/otra	Puntos Estimado: 3
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio

Descripción: Se integrara GPS para poder tener la ubicación exacta en la ruta.
--

En la siguiente tabla 3.15 se describe la historia de usuario 6, pantalla Denuncia importante para el control del vehículo en la ruta.

Tabla 3.15 Historia de Usuario 6

Historia de Usuario	
Numero: 6	Nombre: Denuncia
Modificación de Historia de Usuario:	
Usuario : Usuario Móvil	Asignación Asignada: 5
Prioridad en negocio	Estimados : 3
Riesgo en Desarrollo	Reales : 2
Descripción: Se podrá realizar la denuncia de un vehículo de transporte público no transite por la ruta e	
Observación :	

En la tabla 3.16 se describe la tarea de ingeniería de usuario 6.

Tabla 3.16 Tarea de Ingeniería de Usuario 6: Interfaz Pantalla Denuncia

Tarea de Ingeniería	
Numero de Tarea: 6.1	Numero de Historia de Usuario: 6
Nombre de Tarea: Diseño de Interfaz	
Nombre de Tarea: Desarrollo de Interfaz Pantalla Denuncia	
Estimados: 2	
Fecha Inicio :	Fecha Fin:
Programador Responsable :	Juan Carlos Chuquimia Bustencio
Descripción: Esta ventana estará encargada de enviar los datos del usuario, la información extraída del código QR y una fotografía que se capturara automáticamente como prueba de la falta, toda esta información la enviara el usuario en caso de existir "trameaje".	

3.3.4 Plan de Iteraciones del desarrollo

Para llevar con éxito el desarrollo de prototipo, se definió una serie de tareas que fueron desarrolladas en base a prioridades establecidas en las historias de usuarios, estas tareas se muestra a continuación en la tabla 3.17.

Tabla 3.17 Plan de Iteraciones del Desarrollo

Iteraciones	Nro. De HU	Tareas
Primera iteración	HU1	Diseño de Interfaz
	HU1	Integración de QR code
		Implementación de la base de datos
Segunda iteración	HU1	Diseño de Interfaz
	HU1	Integración de QR code
		Integración obtenida del código QR
Tercera iteración		Integrar y obtener Api Key de Google Place Web
		Integrar y obtener Api Key de Google Place Web
Cuarta iteración		
Quinta iteración		Integración pantalla Denuncia

3.4 FASE DE PRODUCCIÓN

De acuerdo a la metodología de desarrollo de software, esta fase producción se proseguirá con la programación de tres ciclos de desarrollo, en planificación, trabajo y liberación este proceso repetirá iterativamente hasta completar todas las funcionalidades.

Ahora se muestran las actividades del Modelo de Control de Tramos referida a la creación de código para la aplicación móvil Mi Ruta, para verificar su funcionamiento se realizaron pruebas de aceptación en las iteraciones, para que los usuarios puedan observar los diferentes módulos que el modelo de control de tramos posee. Lo primordial en este paso es lograr la funcionalidad del modelo, para esto se utilizó el lenguaje de programación ID Android Studio este cuenta con un emulador virtual de dispositivo Android en el que se puede comprobar la funcionalidad de la aplicación en la pantalla.

3.4.1 Primera Iteración

En la primera iteración se describe el diseño de la interfaz donde se genera el código QR así como la interfaz de inicio de la aplicación móvil.

[Historia de usuario 1, tarea de ingeniería 1.1]. Diseño de interfaz

Planeación

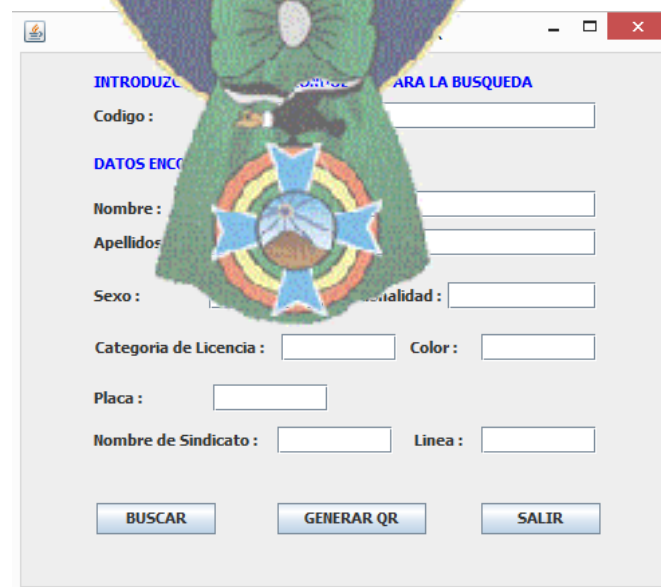
En esta fase se desarrolló la interfaz para la generación de código QR, guiando al administrador en el uso de los botones los cuales son buscar, salir y generar código QR, la búsqueda se realiza introduciendo el código del conductor, una vez hallada la información se procede a codificar la información para luego obtener el resultado.

Trabajo

Esta interfaz se programó en eclipse que consta de varios campos, los cuales se muestran la información buscada en la base de datos.

Liberación

La figura 3.3 se muestra la interfaz de usuario.



INTRODUZCA EL CÓDIGO PARA LA BUSQUEDA

Codigo :

DATOS ENCONTRADOS

Nombre :

Apellidos :

Sexo : Nacionalidad :

Categoría de Licencia : Color :

Placa :

Nombre de Sindicato : Línea :

Figura 3.3 Pantalla Crear Código QR

[Historia de usuario 1, tarea de ingeniería 1.2] Creación de código QR

Planeación

En el símbolo QR se almacenara la información de los atributos más importantes de las entidades que se describen en la tabla 3.18, algunos de estos atributos fueron concatenados para poder mostrar en los campos de la figura 3.3.

Tabla 3.18 Información contenida en el código QR

Entidad	
Conductor	Sexo, nacionalidad,
Vehículo	
Sindicato	Número de línea

En esta iteración será necesario utilizar la librería de java, para poder crear el código QR, esta librería se puede encontrar en el administrador, quien contara con todos los recursos necesarios para ejecutar.

Trabajo

En la codificación se utilizará Eclipse con el fin de crear los códigos QR con la información requerida (ver anexo C).

Liberación

La tabla 3.19 describe la prueba de aceptación para la creación de código QR.

Prueba de Aceptación	
Caso de prueba: Creación de códigos QR	
Numero de caso de prueba: 1	Numero de historia de usuario: 1
Nombre de caso de prueba: Prueba Generar el código QR	
Descripción: El sistema pide el código (id) del conductor con esta información realiza una búsqueda en la base de datos, posteriormente esta información será codificada en un código QR si el Administrador lo requiere.	

<p>Condiciones de Ejecución: las condiciones para esta prueba es que se cuente con una base de datos con las tablas conductor, vehículo, y sindicato además de que el código del conductor se encuentre al momento de la búsqueda.</p>
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresa al sistema (Administrador). 2. Selecciona la opción buscar 3. Verificación del id con la base de datos del sistema 4. Digita la opción GENERAR QR
<p>Resultado Esperado:</p> <ul style="list-style-type: none"> • EL Sistema guardara el código QR en formato jpg <p>La prueba fue exitosa ya que el código QR se guardó en formato jpg con la información de</p>

En esta primera iteración se realizó la implementación de la base de datos el sistema gestionado en PostgreSQL de código abierto, donde se almacenaron los datos importantes como conductor, vehículo, sindicato relacionados al transporte público.

Se procedió a realizar la consulta y la obtención de información de la base de datos, para poder acceder a la información almacenada es necesario para poder consultarla.

3.4.2 Segunda Iteración

[Historia de usuario 2, tareas de desarrollo de software] **Diseño de interfaz**

Planeación

En esta iteración se diseñó la interfaz de inicio de la aplicación móvil, guiando al usuario en el uso de la aplicación con un botón llamado guía de usuario y la otra opción es el botón escanear para el ingreso al reconocimiento de código QR.

Trabajo

En esta fase la interfaz de inicio es una actividad en la programación Android, que tiene una breve explicación del uso de la aplicación y un botón que inicia el escaneo de código QR.



Liberación

El resultado de diseño de la interfaz de la ventana principal se muestra en la figura 3.4



[Historia de usuario 3].

Planeación

Continuando con la primera etapa de diseño y desarrollo la interfaz donde se mostrara la información que se obtiene al escanear el código QR, así como los nombres de calles y avenidas por donde circula el transporte público.

Trabajo

Esta interfaz es una programación Android, la ventana contendrá un botón con la opción que permitirá observar un mapa con la ruta y la ubicación en ella de acuerdo al número de línea vehicular.

Liberación

La figura 3.5 muestra el resultado del desarrollo de la pantalla donde se mostrara la información mencionada.



Figura 3.20. Pantalla informac. de la del Código QR

[Historia de usuario] [Reconocimiento de código QR]

Planeación

La aplicación es un sistema que se accede por medio de la cámara del dispositivo móvil. El usuario debe abrir la aplicación Mi Ruta, con el fin de enfocar un código de naturaleza QR que tiene almacenado, necesarios para el control del vehículo en la ruta.

Trabajo

Para codificación se utilizó un prototipo para el dispositivo móvil la librería ZXing mediante los cuales se realiza el procesamiento y reconocimiento de los códigos QR.

Liberación

La prueba de aceptación 2 se trata del reconocimiento de código QR que se encuentra descrita en la tabla 3.20.

Tabla 3.20 Prueba de Aceptación 2

Prueba de Aceptación	
Caso de prueba: Reconocimiento de Código QR	
Numero de caso de prueba: 2	Numero de historia de usuario: 2
Nombre de caso de prueba: Prueba reconocer el código QR	
Descripción: El sistema espera el reconocimiento de un código QR para mostrar el mensaje almacenado en el símbolo QR.	
Condiciones de Ejecución: Las condiciones para esta prueba es que el usuario cuente con un dispositivo móvil que tenga cámara, podrá reconocer el código QR mostrando el mensaje almacenado en la misma. Además se deberá tener un código QR para poder escanear este.	
Entrada:	
<ol style="list-style-type: none"> 1. Ingresar al sistema 2. Se activa la cámara 3. Escanea el código QR 	
Resultado Esperado	
<ul style="list-style-type: none"> • El sistema muestra el mensaje contenido en el código QR. 	
El reconocimiento del código QR.	

3.4.3 Tercera Iteración

En esta iteración se desarrolló el diseño de rutas de un número de línea vehicular del transporte público.

[Historia de usuario 4, tarea 4.2, 4.3]. Diseño De Interfaz

Planeación

Se desarrollara el diseño de la interfaz donde se mostrara el mapa con la ruta trazada acorde a un número de línea de transporte público, siendo necesarios las APIS de Google Maps mencionados anteriormente.

Trabajo

Para añadir el mapa se utilizó la API de Google Maps para posteriormente realizar la codificación de la ruta se procedió a integrar Google Maps Directions API, Google Place API Web Service (ver anexo C), además utilizando Waypoints para poder trazar la ruta por los calles y avenidas exactas por donde recorre el vehículo de un numero de línea de transporte público.

Liberación

La figura 3.6 es la ventana resultante donde se mostrara el mapa y la ruta



Para esta parte será en la visualización de la ruta y la ruta de un determinado número de línea vehicular del transporte público. La descripción que describe en la tabla 3.21 a la tercera parte de la aplicación móvil Mi Ruta.

Caso de prueba: Desarrollo de la aplicación móvil	Visualización de rutas
Numero de caso de prueba: 3	Numero de historia de usuario: 3
Nombre de caso de prueba: Prueba de Visualización de rutas	
Descripción: La aplicación móvil muestra la ruta de un determinado número de línea vehicular del transporte público.	
Condiciones de Ejecución: las condiciones para esta prueba es que el usuario introduzca el número de línea vehicular y tener acceso a internet	
Entrada:	
	<ol style="list-style-type: none"> 1. Ingresar al sistema (usuario) 2. Introducir el número de línea vehicular 3. Mostrar la ruta de la línea vehicular.
Resultado Esperado:	
	El sistema muestra la ruta de un número de línea vehicular en el mapa.
	La visualización de la ruta en el mapa de un número de línea vehicular es exitosa.

3.4.4 Cuarta Iteración

[Historia de usuario 5, tarea de ingeniería 5.1]. Integración de GPS

Planeación

Para poder tener control del vehículo en la ruta de un número de línea de transporte público se recurre a la Geolocalización, mediante el receptor GPS del dispositivo móvil nos permitirá mostrar la localización del usuario y el vehículo de transporte público en la ruta de número determinada.

Trabajo

Se realizó el desarrollo de métodos que funcionaron adecuadamente en Android.

Liberación

La tabla 3.22 describe los criterios de aceptación respecto a la localización del vehículo de transporte público.

Caso de prueba: Integración de GPS	
Numero de caso de prueba: 1	de historia de usuario: 4
Nombre de caso de prueba: Integración de GPS	
Descripción: El sistema requiere el uso de GPS para que nos muestre el punto exacto en la ruta del número de una línea de transporte público.	
Condiciones de Ejecución: las condiciones para esta prueba es que el usuario cuente con acceso a internet y cuente con un dispositivo móvil. En este caso será un dispositivo móvil.	
Entrada:	
<ol style="list-style-type: none"> 4. Ingresa al sistema(usuario, 5. Se activa GPS 6. Muestra la ubicación exacta. 	
Resultado Esperado:	
<ul style="list-style-type: none"> • El sistema muestra nuestra posición en la ruta. 	
	La geolocalización se realiza exitosamente



3.4.5 Quinta Iteración

[Historia de usuario 6, tarea de ingeniería 6.1]. Diseño de interfaz

Planeación

En la quinta iteración se diseñó la ventana para realizar la denuncia en caso de que el vehículo no transite por la ruta que le fue asignada, en esta ventana el usuario tendrá que añadir sus datos personales en caso de querer realizar la denuncia al digitar la opción Enviar el sistema envía el correo electrónico del usuario más la información contenida en el código QR y una imagen de satélite para la generación automática del lugar exacto donde este altera su ruta.

Trabajo

En la codificación se realizó la implementación de la opción registrar denuncia.

Liberación

El resultado del desarrollo de esta pantalla se muestra en la figura 3.7



Figura 3.7 Pantalla Denuncia

3.5 FASE DE ESTABILIZACIÓN

En esta fase del proyecto como nos indica la metodología Mobile D llegamos a ensamblar todos los módulos del sistema en uno solo para su completa funcionalidad para continuar a realizar las diferentes pruebas al modelo de control de tramos.

En esta fase se finaliza el desarrollo de la aplicación móvil del modelo de control de tramos y se asegura la calidad del producto desarrollado, para asegurar la calidad del producto en el desarrollo del mismo y según la metodología usada se realizó mejoras ensamblando todos los componentes de control.

3.5.1 Workshop de post-entrega

- **Mejoras:** Se mejoró la interfaz de usuario de la aplicación móvil Mi Ruta dándole una mejor experiencia al usuario final y al administrador respectivamente.
- **Fortalezas:** la aplicación móvil Mi Ruta puede ser utilizada de manera offline parcialmente, para la función de verificación de rutas es necesario contar con conexión a internet.
- **Debilidades:** la aplicación móvil Mi Ruta puede ser utilizada solamente si se cuenta con una conexión a internet, el funcionamiento de la aplicación también depende de la buena conexión a internet.

La siguiente tabla 3.23 muestra los resultados de la prueba de aceptación 5 las cuales fueron realizadas a la funcionalidad de la aplicación móvil Mi Ruta.

Tabla 3.23 Resultados de Aceptación 5

Prueba de Aceptación 5	
Caso de prueba: Funcionalidad de la aplicación móvil Mi Ruta	
Numero de caso de prueba: 5	Numero de historia de usuario:
Nombre de caso de prueba: Prueba Funcionalidad Móvil	
Descripción: El sistema requiere mejora en los estilos de interfaz de usuario, se necesita verificar la trayectoria de la línea vehicular en el mapa, validación de campos y revisar si los nombres de los botones son los apropiados.	
Condiciones de Ejecución: las condiciones para la prueba es que se cuente con acceso a internet.	

<p>Entrada:</p> <ol style="list-style-type: none"> 1. Se Ingresó al sistema 2. Se ingresa a la ventana principal 3. Se ingresa a la ventana donde ve se la información de línea vehicular 4. Se ingresa a la ventana donde se muestra la ruta 5. Se ingresa a la ventana denuncia
<p>Resultado Esperado:</p> <ul style="list-style-type: none"> • El sistema muestra los cambios efectuados en la visualización de interfaz de usuario y demás cambios realizados.
<p>Las modificaciones realizadas en las bases de datos</p>

3.6 FASE DE PRUEBA

Esta fase tiene como objetivo verificar el funcionamiento de una versión de la aplicación estable plenamente funcional y su correcta implementación y continuación

Primero será necesario acceder a la base de datos para esto se tendrá acceso a la base de datos del Ministerio de Transportes y Obras Públicas que tiene tablas como conductor, vehículo y sindicato, para la generación de un QR se ingresará el código del conductor, esta ventana se ve representada en la siguiente imagen 8.



Figura 3.8 Pantalla crear QR

Como se observa en la figura 3.8 se introdujo el código del conductor (C.I.), en base al dato introducido el sistema realiza una selección de la base de datos con el botón

buscar, de esta manera se obtuvo la información reflejada en la figura mencionada con información acerca del vehículo, conductor y sindicato.

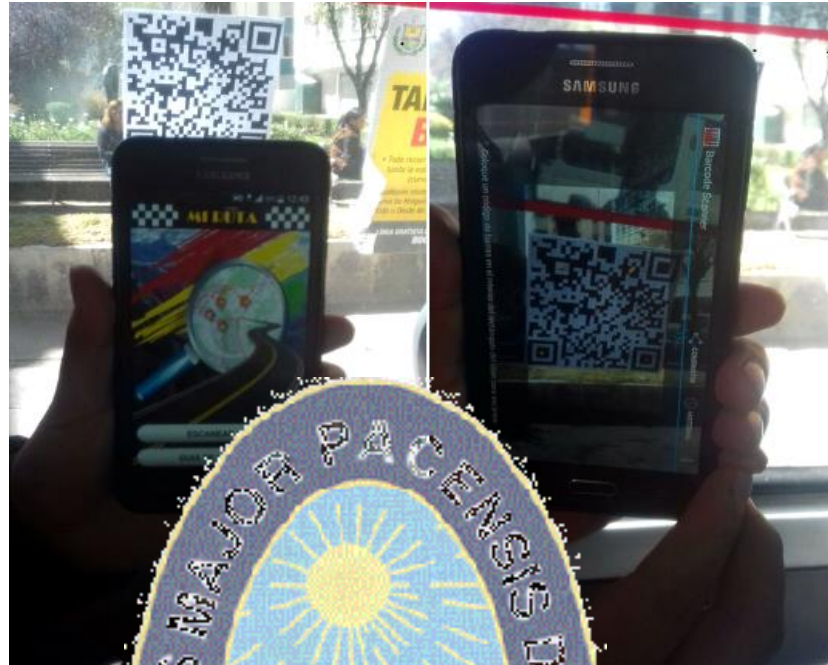
Para crear el código QR con la información obtenida de la base de datos se eligió el botón Generar QR, esta opción nos permitió codificar la información obteniéndose como resultado una imagen en formato jpg el cual se imprimió para la aplicación móvil en papel adhesivo de 120 gr, el símbolo QR es impreso en un tamaño de 9x9 cm el cual tienen una distancia de escaneo, la máxima es de 110 cm y la mínima es de 25 cm, en la figura 3.9 se muestra el código QR.

Estos códigos QR se instalaron en las ventanas laterales del minibús del Sindicato Mini Sur línea 360, como se muestra en la figura 3.10. La ubicación de los códigos QR en las ventanas de los minibuses permite que los usuarios puedan tener fácil acceso al momento de abordar el vehículo.



Figura 3.10 Minibús Línea 360

Posteriormente se instaló la aplicación móvil Mi Ruta, en la ventana principal de la aplicación móvil Mi Ruta existe dos opciones la primera es la opción “Escanear QR” y la segunda “Inicio de Guía”, con la primera opción se procedió a escanear el código QR como muestra la figura 3.11.



Este procedimiento para la siguiente información se halla representada en la figura 3.12, la ventana de información de línea vehicular. Al escanear el código QR, estos datos son acerca del conductor y la información de la línea vehicular, toda esta información la obtenemos.

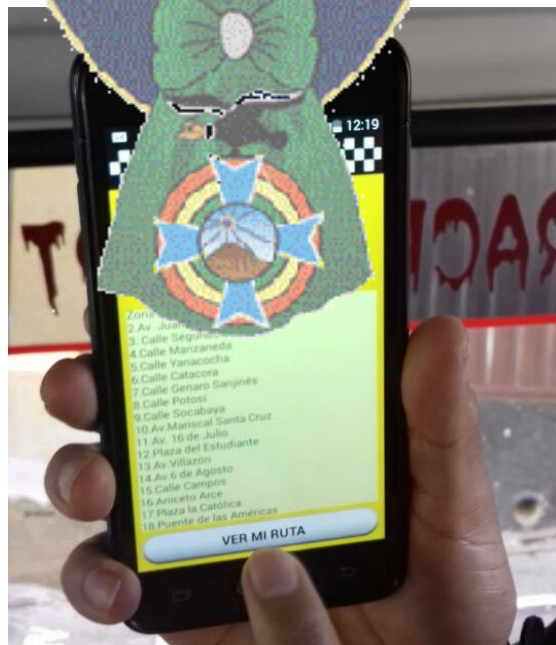


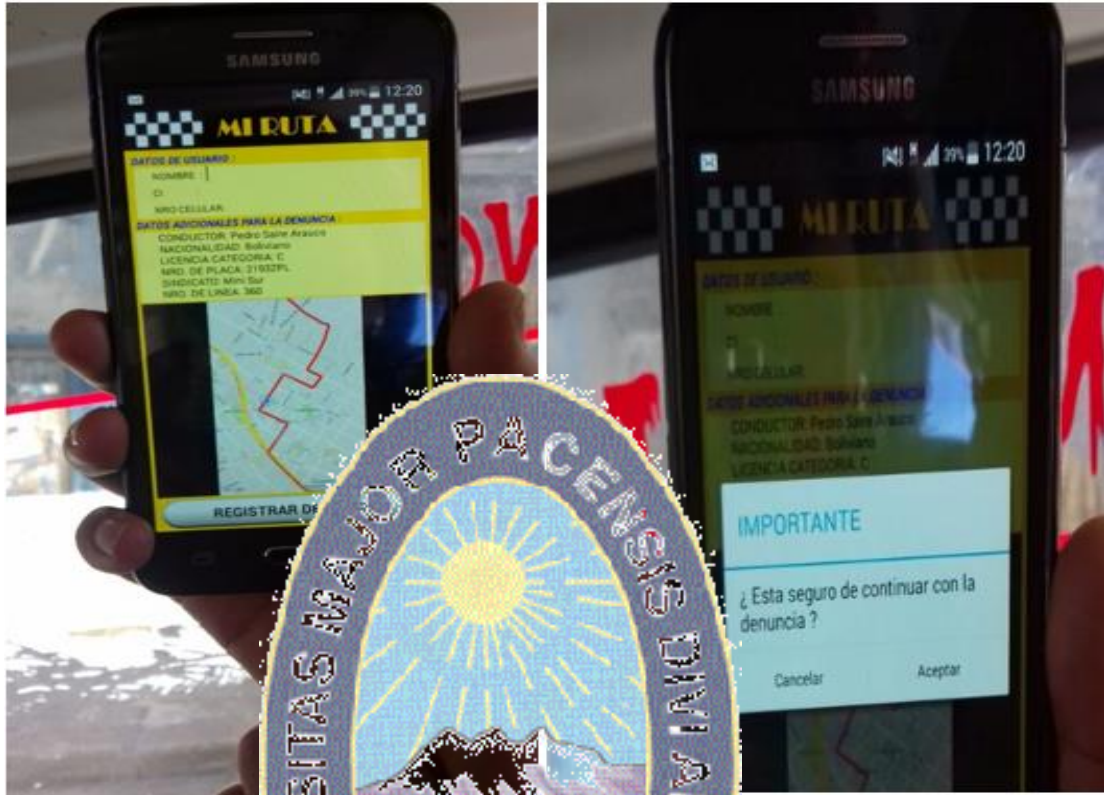
Figura 3.12 Información de línea vehicular

Además debemos mencionar que en esta ventana existe la opción “VER MI RUTA” la cual despliega otra ventana que contiene la trayectoria de la ruta de la línea vehicular contenida en un mapa y la localización del vehículo en la ruta, esta imagen se ve reflejada en la figura 3.13, esta ruta pasara por calles y avenidas por donde debe circular el vehículo de transporte público.



Figura 3.13 Vista de Línea Vehicular

Como puede observarse en la figura 3.13, existe un botón con la opción denuncia, cuando se digita esta opción se despliega la ventana de la figura 3.14, esta tiene por objetivo controlar el vehículo en la ruta por medio de Geolocalización, esta ventana está encargada para enviar denuncias, en caso de ocurrir “trameaje”, los usuarios envían la denuncia con los datos personales que pide el sistema, la información que se obtuvo del código QR además se incluye a la denuncia una foto que se genera automáticamente del lugar donde se produjo la infracción. De esta manera se concluye que las pruebas realizadas del modelo de control de tramos resultaron exitosas.



3.7 RESULTADOS

Después de realizar pruebas de aceptación en las diferentes iteraciones y fases, pruebas en dispositivos móviles y ver la funcionalidad de los distintos módulos, como el más importante en el modelo de control de tramos es la aplicación móvil, ante el cual se podrá realizar el control del vehículo de forma eficiente es la aplicación móvil. Los resultados obtenidos se enfocaron en el aspecto mencionado, los resultados de las pruebas son los siguientes.

3.7.1 Resultados Primera Prueba

Los resultados realizados en esta primera prueba son los resultados hechos al reconocimiento de códigos QR y la respuesta que tiene la aplicación móvil, para esta prueba se usó un código QR impreso en papel adhesivo, este fue reconocido por la aplicación Mi Ruta a través de la cámara del dispositivo móvil, comprobándose de esta manera el funcionamiento del sistema para esta parte del Modelo de Control de Tramo.

En esta prueba se tomó en cuenta el tamaño y la distancia de escaneo del código QR este detalle se describe en la tabla 3.24.

Tabla 3.24 Tamaños y distancia de escaneo de un código QR

Tamaño código QR [cm]	Distancia mínima [cm]	Distancia máxima [cm]
9x9	25	110
11x11	30	140

Los resultados de la primera prueba acerca del reconocimiento de códigos QR se describen en la tabla 3.25.

Tabla 3.25 Resultados Primera Prueba

	PRUEBAS	PORCENTAJE
Pruebas aceptadas	8	100%
Pruebas rechazadas	0	0%
TOTAL PRUEBAS	8	100%
Pruebas Corregidas	0	100%

3.7.2 Resultados Segundo Prueba

Los resultados de la segunda prueba son pruebas realizadas a la visualización del mapa de la ruta que se muestra en la imagen 3.10. La ruta tiene una trayectoria por calles y avenidas que se muestran en el mapa. El número de línea vehicular del transporte público, además de la ubicación del vehículo en la ruta recurrimos a la geolocalización que nos permite mostrar en el mapa la ubicación del vehículo.

Tabla 3.26 Resultados Segunda Prueba

	PRUEBAS	PORCENTAJE
Pruebas aceptadas	11	80%
Pruebas rechazadas	1	10%
TOTAL PRUEBAS	12	100%
Pruebas Corregidas	0	100%

3.7.3 Resultados Tercera Prueba

Por último se realizaron pruebas al funcionamiento de la aplicación móvil Mi Ruta una de las partes más importantes del modelo de control de tramos, En esta tercera prueba se probaron todos los módulos de dicha aplicación probado las cuatro pantallas

que fueron desarrollados verificando que el funcionamiento sea óptimo, los resultados obtenidos se describen en la tabla 3.27.

Tabla 3.27 Resultado Tercera Prueba

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	9	90%
Pruebas rechazadas	1	10%
TOTAL PRUEBAS	10	100%
Pruebas Corregidas	0	100%

3.7.4 Resultado Total

En la tabla 3.28 se describen los resultados de todas las pruebas que se realizaron, como de la tabla 3.26, para después con los datos que se obtuvieron realizar los resultados.

	NÚMERO DE PRUEBAS	PORCENTAJE
Pruebas aceptadas	28	93%
Pruebas rechazadas	2	7%
TOTAL PRUEBAS	30	100%
Pruebas Corregidas	0	100%



Capítulo 4.

PRUEBA DE HIPÓTESIS

Así como es una parte tan importante la teoría que respalda al presente trabajo de investigación, también lo es la correspondiente prueba de hipótesis y los resultados que derivan de la misma.

En este capítulo el objetivo central es el de demostrar que la hipótesis planteada en el primer capítulo se cumplió al usar como herramientas de tipo estadístico.

Al usar el método estadístico se utilizaron pruebas unitarias al sistema como también pruebas de correlación. Los elementos requeridos para desarrollar las pruebas estadísticas se cuenta con las características y tecnologías necesarias y la información codificada.

También se irán desarrollando pruebas de tener una guía con la cual se respaldara los resultados obtenidos en el presente estudio es de gran importancia, pues en este mismo es que se va a determinar si tiene veracidad en la hipótesis ya planteada la cual es:

“El modelo de control de tráfico vehicular en un espacio público aplicando códigos QR y Geolocalización, colabora con la seguridad vial denominada “trameaje” para el cumplimiento de ruta del vehículo.”

4.1 Prueba De Hipótesis

En el mundo real existe una gran variedad de pruebas, de todo tipo de índole; de la misma manera existen otros tipos de pruebas en las áreas de la ingeniería, ciencia y administración, el punto de todo esto es que se requiere que se tome una decisión entre aceptar o rechazar una proposición sobre algún tipo de parámetro, precisamente a esta proposición ya mencionada se le llama hipótesis.

De esta forma se puede dar una definición más formal y estadística de lo que es una hipótesis, una hipótesis es una proposición o supuesto sobre los parámetros de una o más poblaciones.



Para este tipo de aplicaciones se usan tipos de hipótesis de manera más formal esto se puede expresar como:

- H_0 : que representa hipótesis nula
- H_1 : que representa hipótesis alternativa

Esta última puede ser una hipótesis alternativa bilateral o una hipótesis alternativa unilateral, eso depende de los valores que la misma pueda tomar.

Algo para tomar en cuenta, es que las hipótesis siempre son proposiciones sobre la población o sobre la muestra.

Un procedimiento de prueba de hipótesis sobre una hipótesis en particular recibe el nombre de prueba de hipótesis. Los procedimientos de la prueba de hipótesis dependen del empleo de una muestra aleatoria de la población de interés. Si esta información respalda la hipótesis, se concluye que esta es verdadera; por otra parte, si la información es inconsistente con la hipótesis, se concluye que esta es falsa.

Debe hacerse una advertencia sobre la posibilidad de una hipótesis en particular nunca puede conocerse si es verdadera o falsa. Solo se puede examinar a toda la población. Usualmente se trabaja con muestras en muchas situaciones prácticas. Por tanto, es necesario determinar el resultado de una prueba de hipótesis teniendo en cuenta la probabilidad de llegar a una conclusión equivocada.

La hipótesis nula, representada por H_0 , es la afirmación sobre una o más características de poblaciones que se supone es cierta para el investigador.

La hipótesis alternativa, representada por H_1 , es la afirmación contradictoria a H_0 .

La hipótesis nula se rechaza a favor de la hipótesis alternativa, solo si la evidencia muestral sugiere que H_0 es falso. Si la muestra no contradice decididamente a H_0 , se continúa creyendo en la validez de la hipótesis nula. Entonces, las dos conclusiones posibles de un análisis por prueba de hipótesis son rechazar H_0 o no rechazar H_0 .



En el caso de que la hipótesis nula sea verdadera y se la rechazase, esto es llamado error del tipo uno, en caso de que la hipótesis nula sea falsa y no se la rechazase este procedimiento es llamado error de tipo dos.

4.2 Pasos A Seguir En La Prueba De Hipótesis

A continuación se brindara las etapas para una prueba de hipótesis, la cual servirá de gran manera en este capítulo.

- Primero se define la hipótesis, en este caso la hipótesis del investigador.
- Se formula la hipótesis alternativa llamada contra-hipótesis.
- Se elige un nivel de significancia, en este nivel se construye la zona de aceptación.
- Con la zona de aceptación se define la zona crítica la cual se llama región crítica y su área es el nivel de significancia.
- Se verifica la hipótesis.
- Se evalúa con la prueba estadística.
- Mediante el p-valor se rechaza la hipótesis.

4.3 Evaluación De Resultados

Para fines del presente trabajo se realizaron pruebas de la aplicación tanto en diferentes dispositivos móviles. Recordaremos el resumen de los resultados obtenidos mediante las pruebas de aceptación y pruebas finales desarrolladas en el anterior capítulo, ahora el método estadístico que se usara para el presente trabajo es el de proporciones.

Pruebas totales	2
Pruebas aceptadas	1
Pruebas reprobadas:	2
Porcentaje de éxitos:	93%
Porcentaje de fracasos:	7%

Como se observa en los resultados, el porcentaje de éxitos es mayor al 90% esperado al momento de plantearse la Hipótesis, pero para comprobar de manera cuantificable si este valor se asemeja al valor esperado, se realiza una prueba de hipótesis estadística.

Para determinar si el porcentaje de éxitos obtenido en las pruebas puede ser considerado cercano al 95% de nivel de confianza esperado, se hará uso de una prueba de hipótesis para proporciones.

Las variables usadas en dicha prueba serán las mismas mencionadas en la evaluación de casos de prueba:

$$p_0 = 95$$

$$\alpha = 5$$

n es el número total de pruebas realizadas y se conoce además que se tomará un nivel de significancia $\alpha = 5\%$.

Para el caso de la hipótesis alternativa $H_1: p > p_0$:

En este caso se espera un porcentaje de éxitos igual o mayor a 95%:

Es decir:

4.4 Determinación De La Región Crítica

La región crítica para la hipótesis H_1 se representa en la figura 4.1:

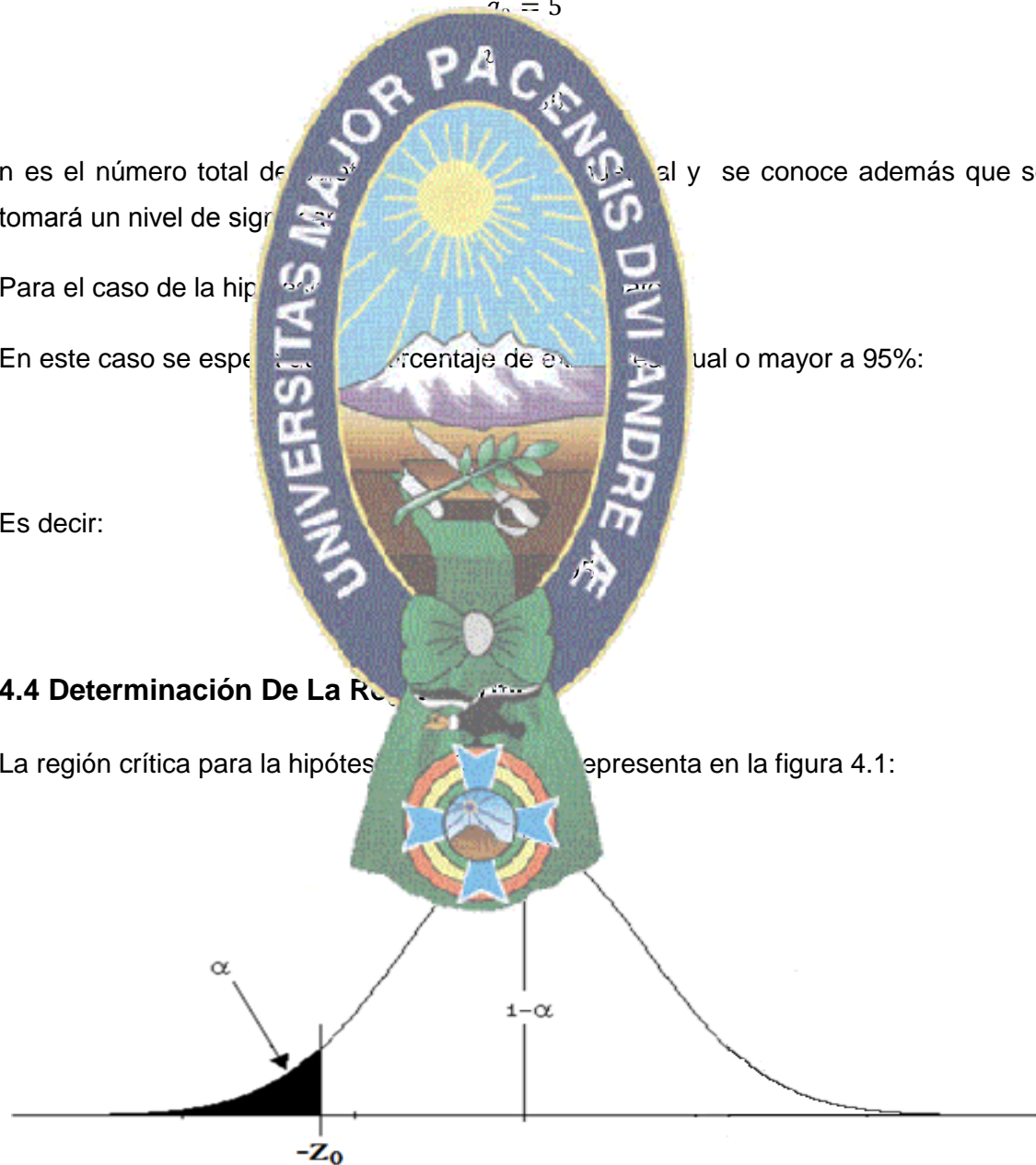


Figura 4.1 Región Crítica para la hipótesis

Como n se refiere en este caso al número de pruebas, en este caso 30, el punto crítico a usar es Z_0 y se determina mediante:

$$-Z_0 = -Z_{1-\alpha} = Z_{1-0.05}$$

$$-Z_0 = Z_{0.95}$$

Este valor se halla de la tabla de la función de distribución normal, la cual se encuentra en el anexo B. Para obtener el valor de z se elige de la tabla mencionada el valor más cercano a 0,95; el cual está ubicado en la fila 1,6 y columna 0,04, este proceso se puede observar en la figura 4.2.

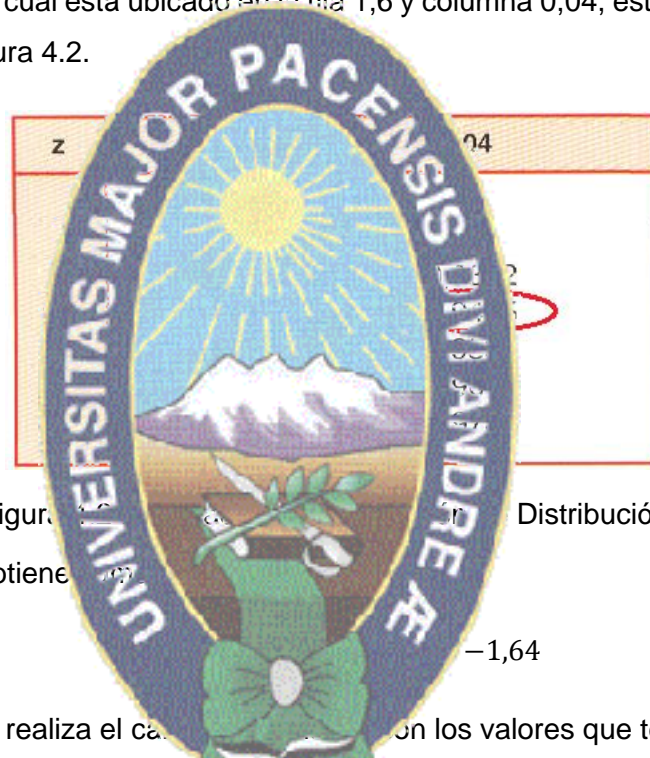


Figura 4.2. Distribución normal

El valor de z se obtiene

-1,64

A continuación se realiza el cálculo con los valores que tenemos:

Como se conoce el número total de pruebas en el espacio muestral, el valor del estadístico de la prueba se obtiene mediante la siguiente fórmula:

$$Z_c = \frac{0.95}{\sqrt{\frac{p_0 q_0}{n}}} = \frac{0.95}{\sqrt{\frac{0.95 * 0.05}{30}}} = -0.503$$

4.5 Toma de Decisión

Al comparar el valor de los estadísticos Z_0 y Z_c , puede observarse en la figura 4.3 que el valor del estadístico de la prueba no se encuentra dentro de la región crítica, por lo tanto se acepta la hipótesis nula H_0 .

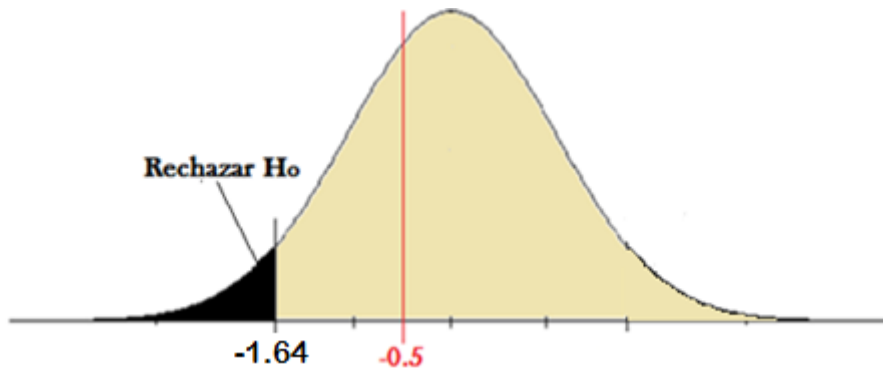


Figura 4.3 Distribución de Z en el grafico para la toma de decisión

El promedio de éxitos de reconocimiento de reconocer las nuestras se acerca al 95%, como ya se mencionó en el capítulo anterior, por lo tanto se acepta H_0 se puede concluir que el modelo de control de tráfico de transporte público aplicando códigos QR y Geolocalización, colabora con el cumplimiento de ruta denominado "trameaje" para el cumplimiento de ruta.

H_0 : "El modelo de control de tráfico de transporte público aplicando códigos QR y Geolocalización, colabora con el cumplimiento de ruta denominado "trameaje" para el cumplimiento de ruta."

4.6 Evaluación de

La variable independiente en este estudio es la aceptación de la aplicación móvil del control de tráfico de transporte público a través de encuestas.

4.7 Aceptación de la aplicación

Se procedió a realizar encuestas a los usuarios del transporte público al "modelo tecnológico de aceptación" o TAM este modelo intenta determinar si los usuarios están dispuestos a aceptar o rechazar una nueva tecnología. En este caso, como se trata de una aplicación móvil, esta encuesta (ver anexo B) nos permitirá evaluar la utilidad y facilidad de uso por parte de los usuarios del transporte público.

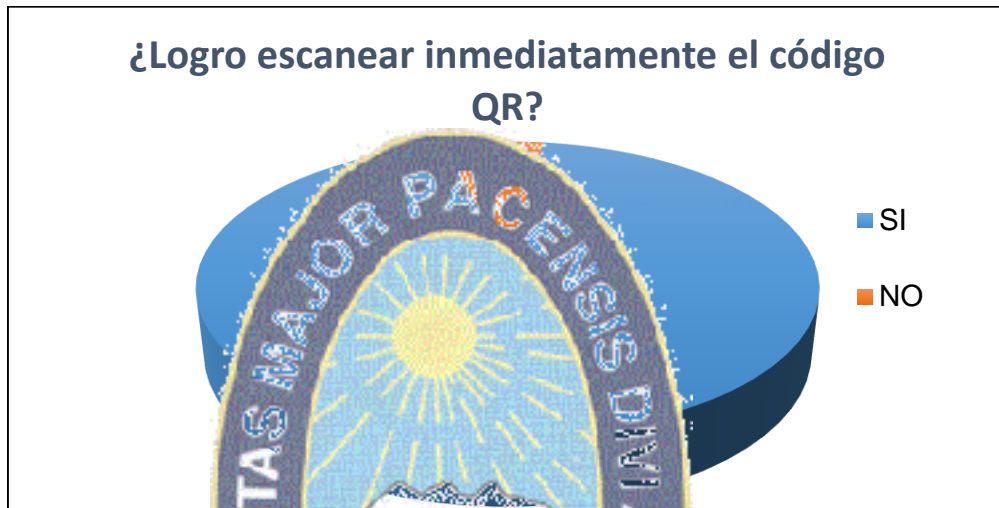
Se procedió a encuestar a un grupo de usuarios de la línea 360 y 207 del sindicato Mini Sur, el número de usuarios encuestados son 30 realizados aleatoriamente, los resultados de la encuesta reflejaron diversas sugerencias para con la aplicación.

4.8 Resultados de las Encuestas

A continuación se mostrara los gráficos de las encuestas realizadas a los usuarios, en los cuales se representa los porcentajes de los resultados obtenidos.

Pregunta Nro. 1

La figura 4.4 Reconocimiento de código QR muestra que los usuarios del transporte público lograron escanear el código QR en un 94% y los usuarios que no lograron fueron en un 6%.



Pregunta Nro. 2

En la figura 4.5 se puede observar que un 100% logro observar la ruta en el mapa.

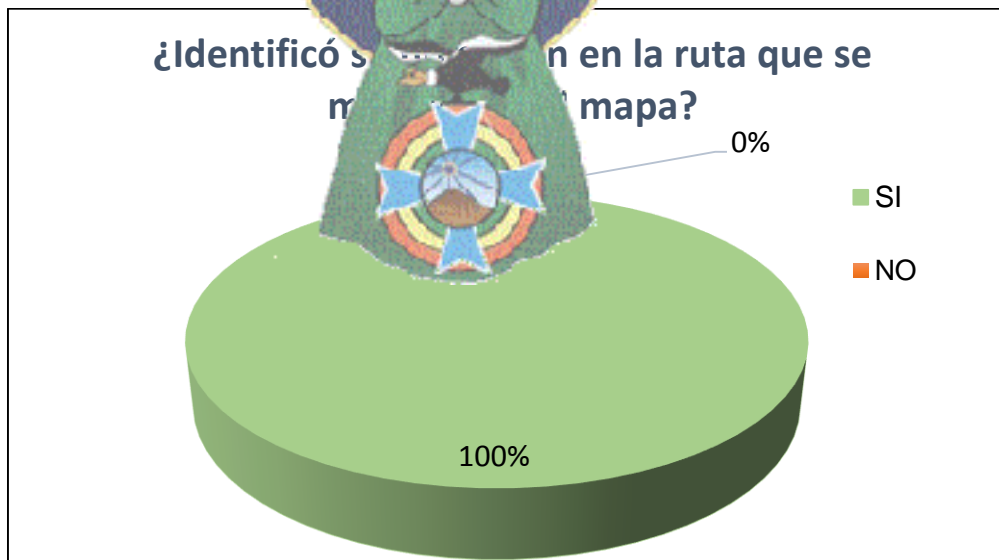
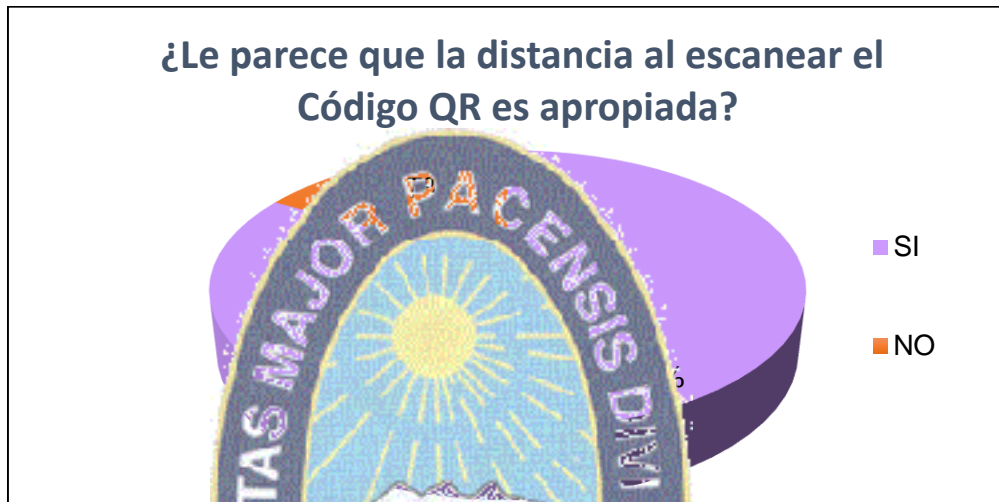


Figura 4.5 Ubicación en la Ruta

Pregunta Nro. 3

En la figura 4.6 a los usuarios que les pareció apropiada la distancia al escanear el código QR fue en un 83% y las personas que no estuvieron de acuerdo con la distancia de escaneo fue en 17%.



Pregunta Nro. 4

La figura 4.7 muestra que el 90% de los usuarios que la información proporcionada acerca de la línea vehicular fue útil y el 10% dijeron que en cierta manera le fue útil son el 10%.

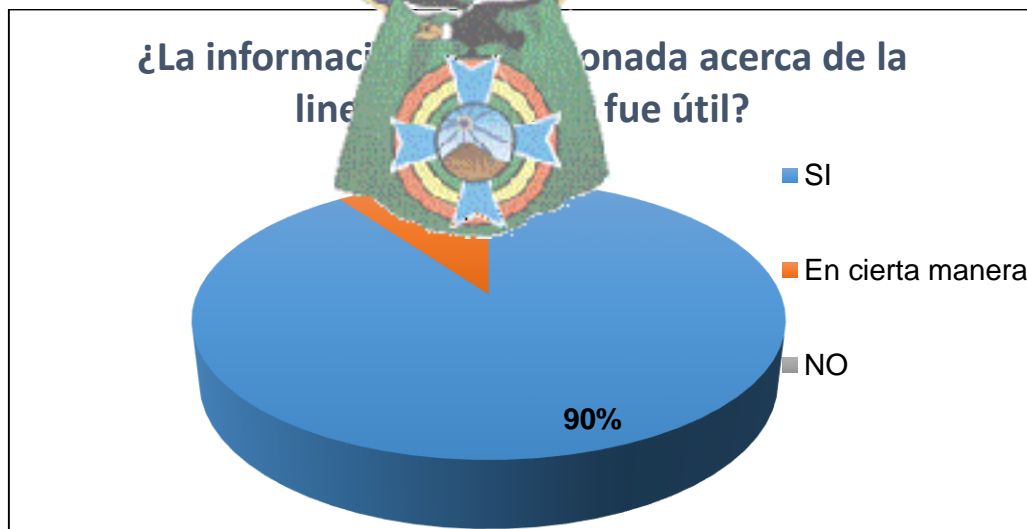
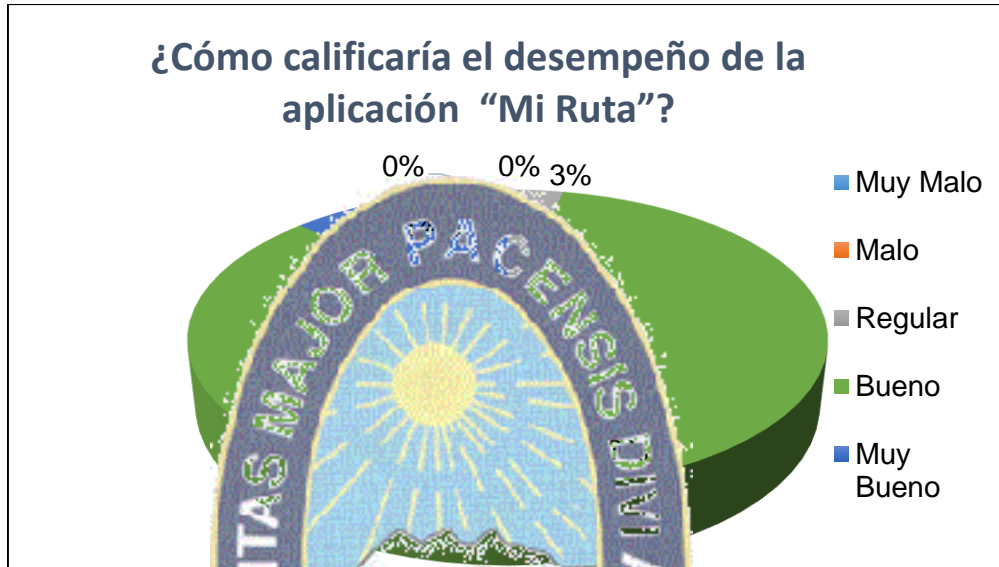


Figura 4.7 Información de línea vehicular

Pregunta Nro. 5

En la figura 4.8 se muestra el resultado del desempeño de la aplicación Mi Ruta donde un 83% les gusto la aplicación.



Pregunta Nro. 6

Según los usuarios ¿volverían a utilizar la aplicación MI Ruta esto puede verificarse en...

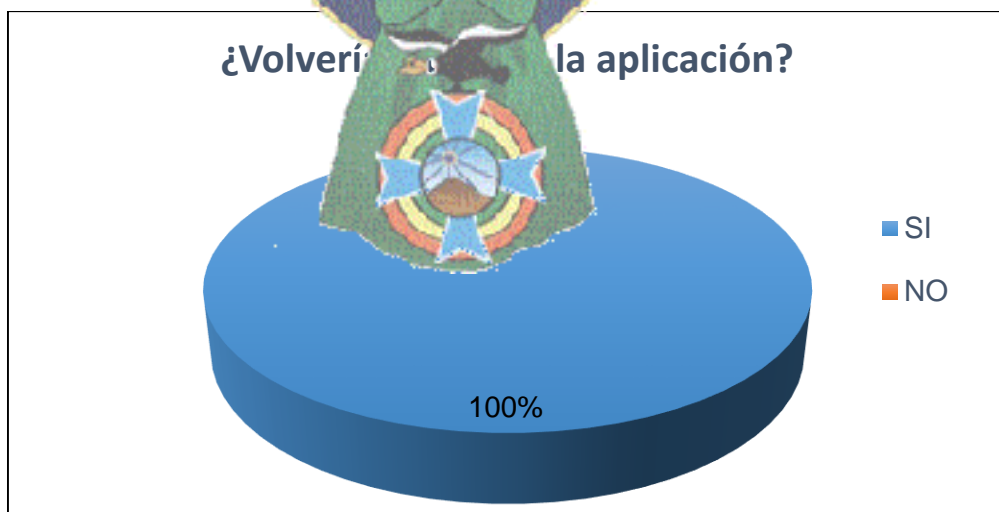
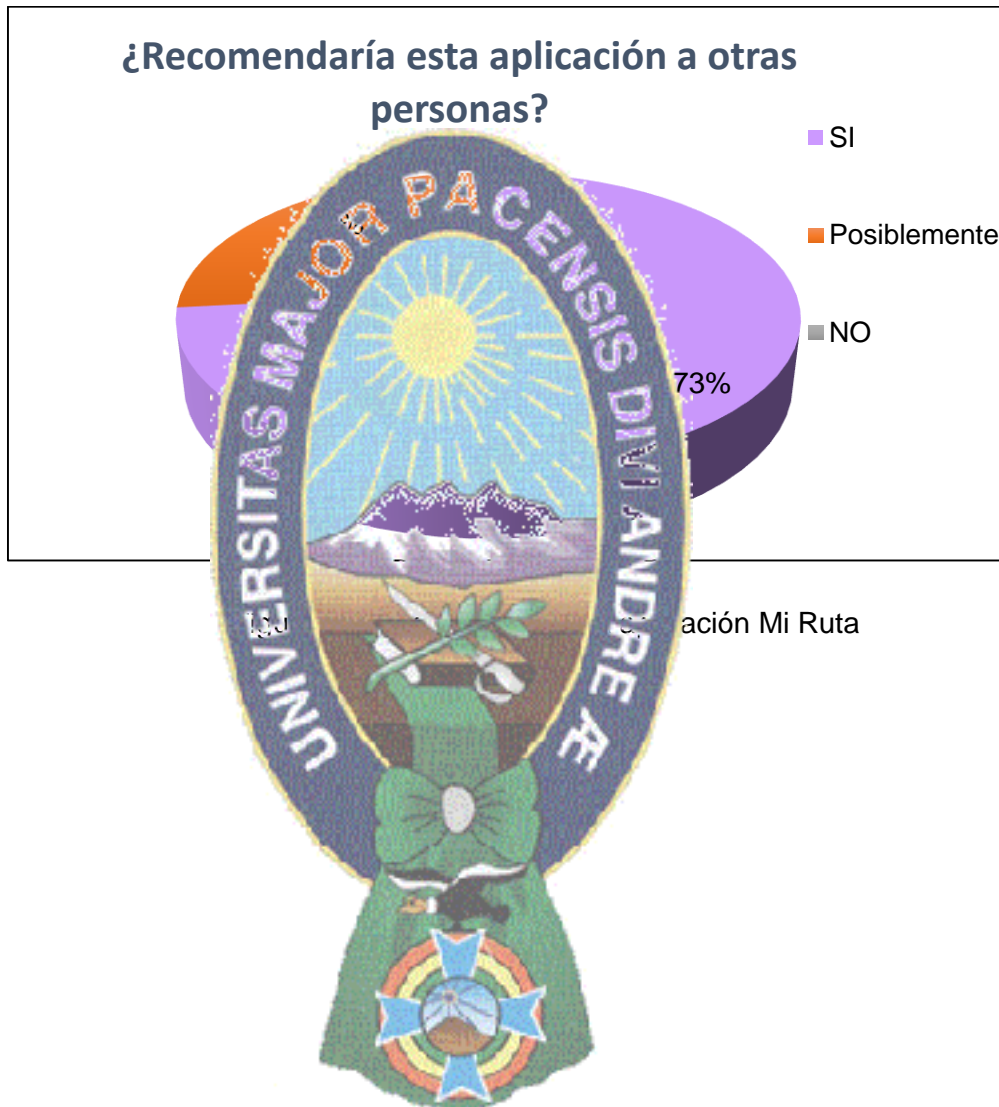


Figura 4.9 Volvería a utilizar la aplicación

Pregunta Nro. 7

Como puede observarse en la figura 4.10 un 73% por ciento de los usuarios encuestados opina que recomendaría la aplicación Mi Ruta, mientras que el 27% de los usuarios eligen la opción posiblemente.



Capítulo 5.

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Dada la problemática que se planteó en el primer capítulo, se pensó en sentar las bases para el Modelo de Control de tramos usando código QR y Aplicaciones Móviles, con este modelo se podrá realizar el control del vehículo en la ruta de una línea vehicular de manera innovadora.

Además de lo mencionado anteriormente, cabe hacer notar que el aporte del presente trabajo permitió demostrar que un código QR y las diferentes utilidades que se generaron de esta manera se podrían resolver problemas de una nueva manera. Para la culminación del presente trabajo de investigación se cumplió con los objetivos específicos por esta razón todo lo expuesto anteriormente permitió afirmar que la hipótesis junto con el objetivo planteado se cumplió. La conclusión de esta afirmación recae en la comprobación del objetivo planteado, la cual es la siguiente solución al problema planteado de la siguiente manera:

Desarrollar un modelo de control de tramos de transporte público para dispositivo móvil, que permita al usuario verificar el pago de un vehículo del transporte público mediante código QR, verificando la ruta del usuario en la ruta.

La solución como objetivo de esta interrogante fue planteada de la siguiente manera:

Obtuve la información de las rutas por donde circula una determinada línea vehicular de transporte público visitando el sitio web de la Gobierno Autónomo del Municipio de La Paz, en donde existe una app con información acerca de las líneas de transporte público.

Se planteó el modelo de control de tramos con la información adquirida acerca de los códigos QR e información obtenida acerca de las líneas de transporte público.

Cumplí con el diseño una aplicación móvil en Android basada en el modelo de control de tramos, el resultado son cuatro pantallas importantes en la aplicación cliente.

Implemente un algoritmo para generar códigos QR, este código tiene información única acerca de la línea de transporte público con datos del conductor, vehículo y sindicato.

Se desarrolló la aplicación móvil con lector de QR el cual decodifica la información contenida en los códigos.

Para el control de vehículo en la ruta se elaboró el sistema utilizando geolocalización recurriendo al receptor GPS del dispositivo móvil.

De esta manera se creó un modelo de control de tramos del transporte público, dicho modelo descriptivo de su correspondiente diseño y análisis fue sometido a diferentes pruebas para la implementación del modelo mediante el uso de dispositivos móviles con geolocalización que permite al usuario un mejor control de la ruta de una determinada línea vehicular realizada de manera eficiente.

Se puede afirmar que como consecuencia del objetivo de este trabajo de investigación, se cumplió.

De acuerdo a las pruebas realizadas se concluye que el modelo es muy útil para los usuarios que requieren un control de transporte público, brindando así un transporte más eficiente y seguro a los usuarios de la línea vehicular para la satisfacción de los usuarios.

Para terminar se vuelve a afirmar que el objetivo del presente trabajo de investigación queda cumplido.

5.2 RECOMENDACIONES

En esta sección del capítulo se harán notar algunas posibles recomendaciones, las cuales son desde la perspectiva y punto de vista del investigador (autor), y están abiertas a cualquier tipo de crítica que el lector note y juzgue, en esta tesis de grado.

Entre las recomendaciones observadas en esta tesis se puede mencionar es la utilización del Modelo de control de tramos usando código QR y aplicaciones móviles en la institución Municipal sugiriendo en la misma que los códigos QR que contendrán información sean pegadas permanentemente para que de esta manera los vehículos del



transporte público conserven el número inicial de vehículos en cada línea vehicular con el objetivo de que no se circulen otras rutas.

Una recomendación acerca del control de tramos es que al momento en que el GPS se activa se debe tener en cuenta que en ocasiones no muestra la posición en la ruta debido a que el internet en Bolivia es lenta ya que en algunos sectores de la ruta la señal de internet se pierde por esta razón sería conveniente desarrollar una alternativa para un mejor control.

Una recomendación dirigida a los alumnos de la carrera de Informática, va sobre el uso de tecnologías que les interesen en las cosas que día a día van desarrollándose y si no les gustan en sus vidas, para facilitarles diferentes tipos de trabajos que el presente trabajo es solo un principio de los diferentes trabajos que se pueden desarrollar utilizando varias tecnologías.



Tapia, M.L. (2013). Estudio Y Desarrollo De Aplicaciones Para Dispositivos Móviles Android. Recuperado el 17 de noviembre 2015 de <http://repositorio.utn.edu.ec/bitstream/123456789/2614/1/04%20ISC%20284%20TESIS.pdf>

Vasquez, D. (2013). Sistema De Administración Y Monitoreo Vía Web de La Posición De Objetos Utilizando Dispositivos Móviles. Recuperado el 20 de octubre de 2015 de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4436/Tesis.pdf?sequence=1>

Wikipedia. (2015). wikipedia. Recuperado el 23 de noviembre de 2015 de http://es.wikipedia.org/wiki/Universidad_Mayor_Pacensis_Divi_Andreae



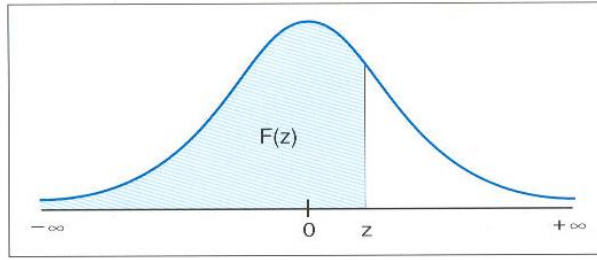
ANEXO A

Árbol de problema



Tabla de distribución normal $N(0, 1)$

$$F(z) = P(Z \leq z)$$



z	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,5000	0,5040	0,5080	0,5120	0,5160	0,5199	0,5239	0,5279	0,5319	0,5359
0,1	0,5398	0,5438	0,5477	0,5517	0,5557	0,5596	0,5636	0,5675	0,5714	0,5753
0,2	0,5793	0,5832	0,5871	0,5910	0,5948	0,5987	0,6026	0,6064	0,6103	0,6141
0,3	0,6179	0,6217	0,6255	0,6293	0,6331	0,6368	0,6406	0,6443	0,6480	0,6517
0,4	0,6554	0,6591	0,6628	0,6665	0,6702	0,6738	0,6772	0,6808	0,6844	0,6879
0,5	0,6915	0,6950	0,6985	0,7019	0,7054	0,7088	0,7123	0,7157	0,7190	0,7224
0,6	0,7257	0,7291	0,7324	0,7357	0,7389	0,7421	0,7454	0,7486	0,7517	0,7549
0,7	0,7580	0,7611	0,7642	0,7673	0,7703	0,7733	0,7764	0,7794	0,7823	0,7852
0,8	0,7881	0,7910	0,7939	0,7968	0,7996	0,8025	0,8051	0,8078	0,8106	0,8133
0,9	0,8159	0,8186	0,8213	0,8241	0,8268	0,8294	0,8315	0,8340	0,8365	0,8389
1,0	0,8413	0,8438	0,8461	0,8485	0,8508	0,8529	0,8554	0,8577	0,8599	0,8621
1,1	0,8643	0,8665	0,8686	0,8708	0,8729	0,8749	0,8770	0,8790	0,8810	0,8830
1,2	0,8849	0,8869	0,8888	0,8907	0,8925	0,8943	0,8962	0,8980	0,8997	0,9015
1,3	0,9032	0,9049	0,9066	0,9082	0,9099	0,9115	0,9131	0,9147	0,9162	0,9177
1,4	0,9192	0,9207	0,9222	0,9237	0,9251	0,9266	0,9279	0,9292	0,9306	0,9319
1,5	0,9332	0,9345	0,9358	0,9371	0,9384	0,9396	0,9406	0,9418	0,9429	0,9441
1,6	0,9452	0,9463	0,9474	0,9484	0,9494	0,9504	0,9515	0,9525	0,9535	0,9545
1,7	0,9554	0,9564	0,9573	0,9582	0,9591	0,9599	0,9608	0,9616	0,9625	0,9633
1,8	0,9641	0,9649	0,9656	0,9664	0,9671	0,9678	0,9686	0,9693	0,9699	0,9706
1,9	0,9713	0,9719	0,9726	0,9732	0,9738	0,9744	0,9750	0,9756	0,9761	0,9767
2,0	0,9772	0,9778	0,9783	0,9788	0,9793	0,9798	0,9803	0,9808	0,9812	0,9817
2,1	0,9821	0,9826	0,9830	0,9834	0,9838	0,9842	0,9846	0,9850	0,9854	0,9857
2,2	0,9861	0,9864	0,9868	0,9871	0,9874	0,9878	0,9881	0,9884	0,9887	0,9890
2,3	0,9893	0,9896	0,9898	0,9901	0,9903	0,9906	0,9909	0,9911	0,9913	0,9916
2,4	0,9918	0,9920	0,9922	0,9924	0,9926	0,9929	0,9931	0,9932	0,9934	0,9936
2,5	0,9938	0,9940	0,9941	0,9942	0,9943	0,9946	0,9948	0,9949	0,9951	0,9952
2,6	0,9953	0,9955	0,9956	0,9957	0,9958	0,9960	0,9961	0,9962	0,9963	0,9964
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9972	0,9973	0,9974
2,8	0,9974	0,9975	0,9976	0,9977	0,9978	0,9978	0,9979	0,9979	0,9980	0,9981
2,9	0,9981	0,9982	0,9982	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986	0,9986
3,0	0,9987	0,9987	0,9987	0,9988	0,9988	0,9989	0,9989	0,9989	0,9990	0,9990
3,1	0,9990	0,9991	0,9991	0,9991	0,9992	0,9992	0,9992	0,9992	0,9993	0,9993
3,2	0,9993	0,9993	0,9994	0,9994	0,9994	0,9994	0,9994	0,9995	0,9995	0,9995
3,3	0,9995	0,9995	0,9995	0,9996	0,9996	0,9996	0,9996	0,9996	0,9996	0,9997
3,4	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9998
3,5	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998
3,6	0,9998	0,9998	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999
3,7	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999
3,8	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999
3,9	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

ANEXO C

CODIGO JAVA HISTORIA DE USUARIO 1 TAREA 1.2

```
282 public void geneQR(String dato)
283 {
284     try
285     {
286         QRCode c=new QRCode();
287         c.setData(dato);
288         c.setDataMode(QRCode.MODE_ALPHANUMERIC);
289         c.setUOM(ucm);
290         c.setLeftMargin(mi);
291         c.setRightMargin(md);
292         c.setTopMargin(ms);
293         c.setBottomMargin(mb);
294         c.setResolution(res);
295         c.setRotate(0);
296         c.setModuleSize(1);
297
298         String a = "user.home" + "/qrdibujo.GIP";
299         c.render(a);
300
301         Desktop d=Desktop.getDesktop();
302         d.open(a);
303
304     }
305     catch (Exception e)
306     {
307         System.out.println(e);
308     }
309 }
310 }
311 }
```

CODIGO JAVA HISTORIA DE USUARIO 2 TAREA 2.2

Método principal para el método de la tarea 1.2.

```
public final Alex (String data, String> desiredBarcodeFormats){
    Intent intents = Intent.ACTION_VIEW;
    intentScan.putExtra(Intent.EXTRA_TEXT, data);

    if(desiredBarcodeFormats != null)
        StringBuilder sb = new StringBuilder();
        for(String format : desiredBarcodeFormats){
            if(!sb.toString().contains(format))
                sb.append(format).append(",");
        }
        intentScan.putExtra(Intent.EXTRA_BURST_FORMATS, sb.toString());

    String targetAppPackage=findTargetAppPackage(intentScan);
    if(targetAppPackage==null){
        return showDownloadDialog();
    }
    intentScan.setPackage(targetAppPackage);
    intentScan.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    intentScan.addFlags(Intent.FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET);
    attachMoreExtras(intentScan);
    startActivityForResult(intentScan, REQUEST_CODE);
    return null;
}
```


Interpretando los datos recibidos de google place web services

```
private void parseJSON(String data) throws JSONException {
    if (data == null)
        return;

    List<Route> routes = new ArrayList<>();
    JSONObject jsonData = new JSONObject(data);
    JSONArray jsonRoutes = jsonData.getJSONArray("routes");
    for (int i = 0; i < jsonRoutes.length(); i++) {
        JSONObject jsonRoute = jsonRoutes.getJSONObject(i);
        Route route = new Route();

        JSONObject overview_polylineJson = jsonRoute.getJSONObject("overview_polyline");
        JSONArray jsonLegs = jsonRoute.getJSONArray("legs");
        JSONObject jsonLeg1 = jsonLegs.getJSONObject(0);
        JSONObject jsonLeg2 = jsonLegs.getJSONObject(jsonLegs.length()-1);
        JSONObject jsonDistance = jsonLeg1.getJSONObject("distance");
        JSONObject jsonDuration = jsonLeg1.getJSONObject("duration");
        JSONObject jsonEndLocation = jsonLeg1.getJSONObject("end_location");
        JSONObject jsonStartLocation = jsonLeg1.getJSONObject("start_location");
        route.distance = new Distance(jsonDistance.getString("text"), jsonDistance.getInt("value"));
        route.duration = new Duration(jsonDuration.getString("text"), jsonDuration.getInt("value"));
        route.endAddress = jsonEndLocation.getString("address");
        route.startAddress = jsonStartLocation.getString("address");
        route.startLocation = new Location("lat", jsonStartLocation.getDouble("lng"));
        route.endLocation = new Location("lat", jsonEndLocation.getDouble("lng"));
        route.points = new ArrayList<>();

        routes.add(route);
    }
    listener.onDirectionalRoutesReady(routes);
}
```

CODIGO DE HISTORIA DE LA UNIVERSIDAD 5 TAREA 5.1

Integrando el GPS

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    LatLng hcmus = new LatLng(-14.233333, -79.016667);
    mMap.moveCamera(CameraUpdateFactory.newLatLng(hcmus));
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.shouldShowRequestPermissionRationale(this)) {
        // TODO: Consider calling
        // ActivityCompat#requestPermissions
        // here to request the missing permissions, and
        // public void onRequestPermissionsResult(int requestCode, String[] permissions,
        // int[] grantResults) to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
    mMap.setMyLocationEnabled(true);
}
```