

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



TESIS DE GRADO

“MODELO DE REALIDAD AUMENTADA PARA EL CONTROL DE TRAMOS DEL TRANSPORTE PÚBLICO BASADO EN ANDROID Y CÓDIGO QR”

CASO: SINDICATO “SIMON BOLIVAR”

Para optar al título de Licenciatura en Ingeniería de Sistemas

Mención: INFORMÁTICA Y COMUNICACIONES

Postulante: Maribel Ana Cruz Sirpa

Tutor Metodológico: Ing. Enrique Flores Baltazar

Tutor Especialista: Lic. Norma Mamani Quispe

Tutor Revisor: Lic. María Magdalena Aguilar Guanto

EL ALTO-BOLIVIA

2020

DEDICATORIA

A Dios por estar conmigo en cada momento de mi vida.

A mis padres por haberme forjado como la persona que soy, mis logros se los debo a ustedes entre los que se incluye este.

A mis hermanos por haberme tenido mucha paciencia.

AGRADECIMIENTOS

A Dios por bendecirme con salud, por guiarme a lo largo de mi existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad, a mis padres por su apoyo incondicional, a mis hermanos por la entera disponibilidad y confianza que tuvieron en mí.

A mi Tutor Especialista la Lic. Norma Mamani Quispe por su tiempo y atención en las repetitivas revisiones de la tesis, tanto técnica como teóricamente, brindándome su colaboración con toda su capacidad y conocimiento para la elaboración de esta tesis de Grado.

Al Ing. Enrique Flores Baltazar por la dedicación, asesoramiento, por sus consejos, amplio conocimiento, y predisposición para la elaboración del presente trabajo.

A la Lic. María Magdalena Aguilar Guanto por la revisión de mi perfil.

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.

RESUMEN

La problemática del Transporte Público en la ciudad de El Alto fue creciendo paulatinamente, el denominado “trameaje” ha sido uno de los principales problemas que obstaculiza brindar un servicio de calidad a los usuarios, el cual perjudica de gran manera a la población que hace uso del servicio de transporte público en la Ciudad de El Alto por esta razón se desarrolló el presente trabajo titulado: “Modelo de Realidad Aumentada para el control de Tramos del Transporte Público basado en Android y Código QR ” donde fue analizado e implementado para poder mejorar la problemática, permitiendo que el usuario pueda realizar un control del cumplimiento de la ruta asignada.

El modelo tiene bases de nuevas tecnologías emergentes en este caso se recurre a la Realidad Aumentada, donde se combina tanto elementos del mundo real como virtual en una misma experiencia, además con una metodología para el desarrollo de aplicaciones móvil la misma es Mobile-D, 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, además recurrimos a librerías de libre distribución para el llamado de mapas el cual es Leaflet.

La implementación del “Modelo de Realidad Aumentada para el Control de Tramos del Transporte Público basado en Android y Código QR” busca eliminar el denominado “trameaje” brindando información en tiempo real, para después mostrar las infracciones que se presenten coadyubando en la toma de decisiones dentro de la organización en este caso el Sindicato “Simón Bolívar”, con el fin de brindar un servicio de calidad a la población que usa diariamente el servicio de transporte público.

PALABRAS CLAVE: QR, móvil, Android, tramos, Realidad Aumentada.

ABSTRACT

The problem of Public Transport in the city of El Alto was gradually growing, the so-called "trameaje" to be one of the main problems that hinders providing a quality service to users which greatly harms the population that makes use of the public transport service in the city of El Alto for this reason the present work was developed entitled : "Augmented Reality Model for the control of Public Transport Routes based on Android and QR Code" where it was analyzed and implemented to be able to improve the problem, allowing the user to carry out a control of the compliance of the assigned route.

The model has bases of new emerging technologies in this case is used to Augmented Reality, where both real and virtual world elements are combined in the same experience, in addition with a methodology for the development of mobile applications the same is Mobile-D, in terms of the tools was Android Studio that allowed us to develop the user interface and integrate the different technologies and tools that were used in this work , we also use free distribution libraries for the so-called maps which is Leaflet.

The implementation of the "Augmented Reality Model for the Control of Public Transport Sections based on Android and QR Code" seeks to eliminate the so-called "trameaje" providing real-time information, and then show the violations that are presented in the decision-making within the organization in this case the Syndicate "Simón Bolívar", in order to provide a quality service to the population that uses the public transport service daily.

KEY WORDS: QR, mobile, Android, sections, Augmented Reality.

ÍNDICE

1. MARCO PRELIMINAR	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	3
1.3. PLANTEAMIENTO DEL PROBLEMA	4
1.3.1. PROBLEMA PRINCIPAL	4
1.3.2. PROBLEMA SECUNDARIOS	4
1.4. OBJETIVOS	5
1.4.1. OBJETIVO GENERAL	5
1.4.2. OBJETIVOS ESPECÍFICOS	5
1.5. HIPÓTESIS	5
1.5.1. IDENTIFICACIÓN DE VARIABLES	5
1.5.2. OPERACIONALIZACIÓN DE VARIABLES	6
1.5.3. CONCEPTUALIZACIÓN DE VARIABLES	6
1.6. JUSTIFICACIÓN	7
1.6.1. CIENTÍFICA	7
1.6.2. TÉCNICA	7
1.6.3. ECONÓMICA	7
1.6.4. SOCIAL	8
1.7. METODOLOGÍA	8
1.7.1. Método Científico	8
1.7.2. Método de Ingeniería	9
1.8. MÉTRICAS DE CALIDAD DE SOFTWARE	10
1.8.1. Modelo de QSOS	10
1.9. MÉTRICAS DE COSTOS	11
1.9.1. COCOMO	11
1.10. HERRAMIENTAS	12
1.10.1. Sistema operativo	12
1.10.2. Base de Datos	12
1.10.3. IDE	12
1.10.4. IDE Android	12
1.10.5. Lenguaje de Programación	12
1.10.6. Mapas	13
1.10.7. Librerías de desarrollo	13
1.11. LÍMITES Y ALCANCES	13
1.11.1. Límites	13
1.11.2. Alcances	13
1.12. APORTES	14
2. MARCO TEÓRICO	15
2.1. INTRODUCCIÓN	15
2.2. MODELO	15
2.3. REALIDAD AUMENTADA	16
2.3.1. Realidad Aumentada con Marcadores	16
2.3.2. Realidad Aumentada con Geolocalización	17

2.4. GEOLOCALIZACIÓN	18
2.4.1. Hardware	18
2.4.2. Software	18
2.4.3. Conexión a Internet	18
2.4.4. Geolocalización, localizar y GPS.	19
2.5. CONTROL	19
2.6. SISTEMA DE CONTROL	20
2.7. TRAMO	20
2.8. TRANSPORTE PUBLICO	20
2.8.1. Características del Transporte Publico	21
2.8.2. Componentes Físicos del Transporte Publico	21
2.9. PLATAFORMA ANDROID	22
2.9.1. Características	23
2.9.2. Arquitectura	23
2.9.3. La máquina virtual Dalvik	25
2.10. CÓDIGO QR “QUICK RESPONSE CODE”	25
2.10.1. Características Generales	26
2.10.2. Conceptos básicos de una estructura de código QR	26
2.10.3. Corrección de errores	28
2.10.4. Área de impresión reducida	29
2.10.5. Alta capacidad de codificación de los datos	29
2.10.6. Resistentes a suciedad o daño parcial	30
2.10.7. Legibles desde cualquier dirección en 360°	31
2.10.8. Obteniendo datos de un código QR	31
2.11. INGENIERÍA DE SOFTWARE	32
2.11.1. Características	32
2.11.2. El Proceso del Software	33
2.12. METODOLOGÍA MOBILE-D	35
2.12.1. Exploración	36
2.12.2. Inicialización	36
2.12.3. Producto	37
2.12.4. Estabilización	37
2.12.5. Pruebas	38
2.13. MODELO DE QSOS	38
2.13.1. Definición	39
2.13.2. Evaluación	39
2.13.3. Calificación	41
2.13.4. Selección	41
2.14. COCOMO	41
2.14.1. Modos de Desarrollo	42
2.14.2. Modelos de Estimación (Básico, Intermedio y Detallado)	43
2.15. HERRAMIENTAS A UTILIZAR	47
2.15.1. Sistema operativo	47
2.15.2. Base de Datos	48
2.15.3. IDE	49

2.15.4. IDE Android	49
2.15.5. Lenguaje de Programación	50
2.15.6. Mapas	51
2.15.7. Librerías de desarrollo	51
3. MARCO APLICATIVO	53
3.1. INTRODUCCIÓN	53
3.2. ESTRUCTURA DEL MODELO	53
3.3. METODOLOGÍA MOBILE-D	54
3.3.1. Fase 1: Exploración	54
3.3.2. Fase 2: Inicialización	55
3.3.3. Fase 3: Producción	63
3.3.4. Fase 4: Estabilización	74
3.3.5. Fase 5: Pruebas.	75
3.4. MÉTRICA DE CALIDAD QSOS	77
3.4.1. Tarjeta de identificación del software “Sistema Generador QR”	77
3.4.2. Tarjeta de identificación del software “RUApp”	78
3.4.3. Evaluación de los criterios de valoración (Sistema Generador QR)	78
3.4.4. Evaluación de los criterios de valoración (RUApp)	81
3.5. ANÁLISIS DE COSTO COCOMO	83
4. PRUEBAS Y RESULTADOS	85
4.1. PRUEBAS	85
4.1.1. Generación de Código QR	85
4.1.2. Implementación de Código QR en el minibús de la línea 643	87
4.2. RESULTADOS	90
4.2.1. Resultados Prueba de escaneo código QR	90
4.2.2. Resultados Prueba visualización de del mapa, geolocalización y trasado de ruta.	91
4.2.3. Resultados prueba funcionamiento general de la aplicación móvil RUApp	91
4.2.4. Resultado Total	92
4.3. PRUEBA DE HIPÓTESIS	92
4.4. EVALUACIÓN DE LOS CASOS DE PRUEBA	92
4.5. PROCEDIMIENTO DE LA PRUEBA DE HIPÓTESIS	93
4.5.1. Definición de la Hipótesis en Investigación	93
4.5.2. Evaluación de Resultados	93
4.5.3. Determinación de la Región Critica	94
4.5.4. Cálculo estadístico de la prueba	95
4.5.5. Toma de decisión	95
5. CONCLUSIÓN Y RECOMENDACIÓN	96
5.1. CONCLUSIONES	96
5.1.1. CUMPLIMIENTO DE LOS OBJETIVOS.	96
5.2. RECOMENDACIONES	97

ÍNDICE DE TABLAS

Tabla 1. 1 Descripción de operacionalización de variables -----	6
Tabla 2. 1 Descripción de capacidad Código QR -----	30
Tabla 2. 2 Descripción de capacidad Código QR -----	40
Tabla 2. 3 Hoja de Evaluación de software -----	40
Tabla 2. 4 Ecuaciones del Modelo Básico de COCOMO. -----	44
Tabla 2. 5 Ecuaciones del Modelo Intermedio de COCOMO -----	46
Tabla 3. 1 Identificación de Actores-----	54
Tabla 3. 3 Requerimientos definidos –Tomar foto al código QR -----	57
Tabla 3. 4 Requerimientos definidos – Georreferenciar Ruta-----	57
Tabla 3. 5 Requerimientos definidos – Consultar información reservado -----	58
Tabla 3. 6 Requerimientos definidos – Denunciar el incumplimiento de la ruta	58
Tabla 3. 7 Descripción de alto nivel – ingresar a la aplicación -----	59
Tabla 3. 8 Descripción de alto nivel – Escanear Código QR-----	59
Tabla 3. 9 Descripción de alto nivel – Mostrar Ruta-----	60
Tabla 3. 10 Descripción de alto nivel – Denunciar el Incumplimiento de la Ruta -----	60
Tabla 3. 11 Descripción de alto nivel - Agregar Nuevo Conductor -----	61
Tabla 3. 12 Descripción de alto nivel-Generar QR-----	61
Tabla 3. 13 Descripción de alto nivel - verificar denuncias -----	61
Tabla 3. 14 Plan de Iteraciones del Desarrollo-----	64
Tabla 3. 15 Diseño de Interfaz – primera iteración-----	65
Tabla 3. 16 Implementación de la base de datos -----	66
Tabla 3. 17 Creación de código QR-----	66
Tabla 3. 18 Diseño de Interfaz de la App “RUApp” - Segunda iteración -----	69
Tabla 3. 19 Reconocimiento de QR - Integrar ZXing-----	70
Tabla 3. 20 Pantalla información obtenida del código QR -----	71
Tabla 3. 21 Diseño de interfaz y Mapa con Leaflet - Tercera iteración-----	72
Tabla 3. 22 Unificación de GPS - Cuarta iteración-----	72
Tabla 3. 23 Diseño de interfaz pantalla Denuncia - quinta iteración -----	73
Tabla 3. 24 Prueba de Aceptación Sistema Gestor de QR -----	75
Tabla 3. 25 Prueba de Aceptación - funcionalidad de la aplicación móvil -----	76
Tabla 3. 26 Tarjeta de identificación del software “Sistema Generador QR” ---	77
Tabla 3. 27 Tarjeta de identificación del software “RUApp” -----	78
Tabla 3. 28 Evaluación Criterios Sistema Generador QR-----	79
Tabla 3. 29 Valoración Criterios Sistema Generador QR -----	80
Tabla 3. 30 Evaluación Criterios RUApp -----	81
Tabla 3. 31 Valoración Criterios RUApp-----	82
Tabla 3. 32 Coeficiente ab y cb y los exponentes bb y db-----	83
Tabla 4. 1 Dimensión y distancia de escaneo QR -----	90
Tabla 4. 2 Resultados de la primera prueba-----	90
Tabla 4. 3 Resultados de la segunda prueba -----	91
Tabla 4. 4 Resultados de la tercera prueba-----	91
Tabla 4. 5 Resultados de la Prueba Total -----	92
Tabla 4. 6 Resultado tabla de la función de distribución normal -----	94

ÍNDICE DE FIGURAS

Figura 2. 1 Marcador (Estebanell,2012) -----	17
Figura 2. 2 Geolocalización (Estebanell, 2012)-----	17
Figura 2. 3 Logo Android (M.Tapia,2013) -----	22
Figura 2. 4 Arquitectura de Android (Jaime Aranaz, 2009)-----	24
Figura 2. 5 Estructura Genérica de Símbolo QR (Luque,2015) -----	27
Figura 2. 6 Versiones de los Códigos QR (QR Code, 2011) -----	28
Figura 2. 7 Comparativa espacio de impresión código de barras vs QR (QR Code, 2011) -----	29
Figura 2. 8 Vista general Código QR (QR Code, 2011) -----	30
Figura 2. 9 Daños en un Código QR impreso (QR, 2011) -----	30
Figura 2. 10 Patrones de detección de posición (QR Code, 2011)-----	31
Figura 2. 11 Proceso de Etiquetado Movil (Ramonda, 2014)-----	31
Figura 2. 12 Capas de la Ingeniería de Software (Pressman, 2002) -----	33
Figura 2. 13 Ciclo de desarrollo Mobil-D (Vique, 2012 -----	35
Figura 2. 14 Proceso QSOS (Pilot, 2006 -----	39
Figura 2. 15 Conceptualización básica COCOMO (Garita, 2016) -----	42
Figura 2. 16 Esquema de modos de desarrollo de software (Boehm, 1981 -----	43
Figura 2. 17 Valores constantes por modo de desarrollo (Boehm,1981 -----	43
Figura 2. 18 Ecuaciones por tipo de modelo (Boehm, 1981) -----	47
Figura 3. 1 Estructura del Modelo General (Elaboración propia) -----	53
Tabla 3. 2 Estructura de la aplicación (Elaboración propia) -----	56
Figura 3. 2 Caso de uso Usuario (Elaboración propia) -----	59
Figura 3. 3 Caso de uso - Administrador (Elaboración propia) -----	60
Figura 3. 4 Prototipo Inicial - Pantalla de Inicio (Elaboración propia) -----	62
Figura 3. 5 Prototipo Inicial - Escaneado de Código QR (Elaboración propia) --	62
Figura 3. 6 Prototipo Inicial – Lista de Información (Elaboración propia) -----	62
Figura 3. 7 Prototipo Inicial – Visualización del mapa y el trazado de la Ruta (Elaboración Propia) -----	63
Figura 3. 8 Prototipo Inicial – Registro de la Denuncia (Elaboracion propia) ----	63
Figura 3. 9 Modelo Entidad Relación (Elaboración propia)-----	63
Figura 3. 10 Diagrama de Clase SGQR (Elaboración propia) -----	64
Figura 4. 1 Prueba adición de afiliado (Elaboración propia)-----	85
Figura 4. 2 Prueba Guardado de afiliado (Elaboración propia)-----	85
Figura 4. 3 Prueba Generación de QR (Elaboración propia)-----	86
Figura 4. 4 Prueba Impresión de QR (Elaboración propia) -----	86
Figura 4. 5 Minibús Línea 643 (Fotografía propia) -----	87
Figura 4. 6 Escaneado Código QR (Elaboración propia) -----	87
Figura 4. 7 Información de la línea Vehicular (Elaboración propia) -----	88
Figura 4. 8 Ruta de la línea Vehicular (Elaboración propia)-----	88
Figura 4. 9 Ventana de Denuncia (Elaboración propia) -----	89
Figura 4. 10 Portal Web Denuncias (Elaboración propia)-----	89
Figura 4. 11 Región critica para la hipótesis (Elaboración propia) -----	94
Figura 4. 12 Distribución de Z en el grafico para la toma de decisiones (Elaboración propia)-----	95



CAPÍTULO I

MARCO

INTRODUCTORIO

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Los teléfonos móviles se han convertido en los principales medios de conexión a la red, posibilitándose constante interrelación y comunicación entre personas, sociedades, empresas y los demás actores del mundo moderno, por esta razón los dispositivos móviles se volvieron una necesidad, objetos del diario vivir para los cuales no hay restricción de edad lo usan desde niños hasta ancianos ingresando así la época de la revolución virtual. (Guillermo Contreras, 2013)

En este sentido, las tecnologías disponibles en la actualidad como los QR¹, códigos bidimensionales también llamados "códigos de respuesta rápida", los cuales son capaces de contener información de distinto tipo y son legibles mediante una cámara y un software de lectura para códigos QR, nos evidencia la interactividad en los denominados "nuevos espacios electrónicos construidos a través de imágenes virtuales con el espacio físico" que se conoce actualmente como Realidad Aumentada. (Nussbaum, 2009)

Otro aspecto que se tomara en cuenta es la Geolocalización es un concepto relativamente nuevo, que ha proliferado de unos dos años a esta parte y que hace referencia al conocimiento de la propia ubicación 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. (Kevin F., 2009)

Asumiendo las herramientas que se nos proporciona en la actualidad se presenta el problema que viene en perjuicio de la población en la Ciudad de El Alto, el cumplimiento de los "tramos" asignados a cada sindicato de transporte público,

¹ (Quick Response Code), Códigos de Respuesta Rápida

La Alcaldía de El Alto identificó los puntos de “trameaje” en la ciudad, donde además existen atolladeros y desorden que afectan a los usuarios. Los dirigentes se comprometieron a eliminar los recorridos por tramos, pero reconocen que hay dificultades en el control de sus afiliados. (Miguel Rivas, 2017)

Entonces se pretende usar los códigos QR como una tecnología que puede incorporarse al control de tramos en el transporte público de la ciudad de El Alto, además gracias al uso de dispositivos móviles y la geolocalización, se busca dar solución al problema denominado “trameaje” en el transporte público, ya que pese a normativas y reglamentos municipales, los “trameajes” por parte de algunos choferes continúan, principalmente en horarios nocturnos y en lugares alejados, debido a que los choferes cambian su número de línea o el denominado disco, en el cual se observa el número de línea y rutas del minibús. “Esta situación perjudica en gran manera a los pasajeros, quienes deben abordar más de un vehículo para llegar hasta su hogar”. El incumplimiento de rutas todavía persiste para las personas que viven principalmente en las ciudades El Alto y La Paz.

Así el punto de investigación en el presente trabajo será el Sindicato “Simón Bolívar”, una entidad que tiene como finalidad brindar servicio de autotransporte público en la Ciudad de El Alto distrito 5, esto en un horario continuo.

1.2. ANTECEDENTES

Internacionales

[PAULO CORONADO]; “DISEÑO DE APLICACIÓN DE REALIDAD AUMENTADA EN DISPOSITIVOS MOVILES PARA USUARIOS DEL SISTEMA INTEGRADO DE TRANSPORTE PUBLICO BOGOTA”, Generar un prototipo como herramientas tecnológicas de fácil consulta y eficiente comunicación para los dispositivos móviles, mostrando los diferentes puntos de acceso y rutas que se encuentran ubicadas sobre la carrera 13 público mediante Realidad Aumentada (FACULTAD DE INGENIERIA, UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS, 2015).

[JOSÉ LUIS LEIVA OLIVENCIA]; “REALIDAD AUMENTADA BAJO TECNOLOGÍA MÓVIL BASADA EN EL CONTEXTO APLICADA A DESTINOS TURÍSTICOS”, El objetivo del trabajo es definir un soporte teórico para la creación y configuración de un sistema de realidad aumentada para un destino turístico, donde los usuarios puedan disponer de herramientas para planificar individualmente o en grupo visitas o rutas turísticas, teniendo en cuenta sus preferencias y contexto. (INFORMÁTICA, UNIVERSIDAD DE MÁLAGA, 2014).

Nacional

[FREDY LOPEZ GOMEZ]; “SISTEMA DE INFORMACION PARA EL CONTROL DE RUTAS DEL TRANSPORTE PUBLICO VEHICULAR DE LA CIUDAD DEL EL ALTO VIA TELEFONIA MOVIL”, Implementar un sistema de información con todas las rutas de la ciudad del Alto, la búsqueda de rutas se realiza mediante zona y numero de línea del vehículo, para coadyuvar el control del recorrido por tramos del transporte público vehicular de la Ciudad de El Alto. (FACULTAD DE CIENCIAS PURAS INFORMÁTICA, UNIVERSIDAD MAYOR DE SAN ANDRÉS, 2014).

[ELIZABETH CERTAN CHUQUIMIA]; “APLICACIÓN PARA DISPOSITIVOS MÓVILES EN PLATAFORMA ANDROID PARA LA ADMINISTRACION DE PEDIDOS Y CONTROL DE RUTAS DE VENDEDORES. CASO: EMPRESA IMPORTADORA MEGA VIT”, Desarrollar una aplicación para dispositivos móviles en plataforma Android que permita automatizar la gestión de pedidos y controlar rutas a vendedores de la Empresa Importadora Mega Vit. (FACULTAD DE CIENCIAS PURAS INFORMÁTICA, UNIVERSIDAD MAYOR DE SAN ANDRÉS, 2016).

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. PROBLEMA PRINCIPAL

El recorrido por tramos denominado “trameaje” en el transporte público, hace necesario recurrir a las nuevas tendencias tecnológicas, ya que cada vez se tienen un número mayor de rutas que no se llegan a cumplir, las cuales perjudican a la población en general al no poder llegar a su destino.

1.3.2. PROBLEMA SECUNDARIOS

- Falta de control en las líneas de transporte público.
- Cambian de ruta en función del horario y la posibilidad de mayor número de pasajeros.
- Vehículos del transporte público recorren otras rutas.
- Cobro por tramos por parte de los dueños de vehículos.
- Líneas de transporte público no tienen la cantidad habitual de vehículos.
- La falta de aplicación de las TIC al problema de control de transporte.

¿La aplicación móvil basada en el modelo de realidad aumentada para el control de tramos del transporte público, será capaz de cooperar al usuario en el problema denominado “trameaje”, utilizando códigos QR y Geolocalización?

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un modelo de realidad aumentada para el 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.

1.4.2. OBJETIVOS ESPECÍFICOS

- ✓ Analizar el cumplimiento de las rutas del transporte público.
- ✓ Plantear el modelo de realidad aumentada para el control de tramos en base al análisis de la información recaba.
- ✓ Diseñar una aplicación Android basada en el modelo de realidad aumentada para de control de tramos.
- ✓ Implementar métodos para generar códigos QR con una información única.
- ✓ Desarrollar un lector de QR en la aplicación móvil.
- ✓ Desarrollar un módulo por el cual el usuario puede realizar la queja correspondiente.
- ✓ Realizar las pruebas de control de calidad del software.

1.5. HIPÓTESIS

Con la ayuda de la ingeniería de software, herramientas y metodologías se tiene un modelo de realidad aumentada aplicado al control de tramos de transporte publico basado en Android y código QR el cual tendrá una eficiencia del 90%.

1.5.1. IDENTIFICACIÓN DE VARIABLES

Ciertamente, el Modelo de Realidad Aumentada para el control de tramos se sustenta del análisis y la información recababa de un área determinada.

- **VARIABLE DEPENDIENTE:**
Control de tramos
- **VARIABLE INDEPENDIENTE:**
Modelo de Realidad Aumentada

1.5.2. OPERACIONALIZACIÓN DE VARIABLES

Tabla 1. 1

Descripción de operacionalización de variables

VARIABLE	DIMENSIÓN	INDICADORES	ACTIVIDADES
Control de Tramos	<ul style="list-style-type: none">• Kilometro (km)• Monitoreo	<ul style="list-style-type: none">✓ Unidad de medida, superficie de área recorrida✓ Datos de análisis de espacio.	Observación Documentación
Modelo de Realidad Aumentada	<ul style="list-style-type: none">• Pruebas• Algoritmos	<ul style="list-style-type: none">✓ Calcular la eficiencia más cercana a 100%	Observación

Nota. Elaboración propia

1.5.3. CONCEPTUALIZACIÓN DE VARIABLES

1.5.3.1. Control de Tramos

- **Control:** Consiste en verificar si todo ocurre de conformidad con lo adoptado, con las instrucciones emitidas y con los principios establecidos. (Henry Fayol - 2012)
- **Tramo:** Es una porción de superficie o terreno, que aparece de algún modo señalizada como subdivisión del mismo, aplica a un camino o vía transitable que es una porción de un trayecto mayor.

1.5.3.2. Modelo de Realidad Aumentada

Un Modelo de Realidad Aumentada se crea mediante la aplicación de un algoritmo a los datos, pero es algo más que un algoritmo, es un conjunto de datos que combina elementos reales y virtuales, la cual se interactiva en tiempo real. (Tom Caudell - 2009)

1.6. JUSTIFICACIÓN

1.6.1. CIENTÍFICA

La construcción de este tipo de modelos de desarrollo de aplicaciones móviles en sistemas operativos como Android, está cobrando mayor importancia en la actualidad debido a la demanda de dispositivos. Por tal motivo se pensó en explotar las características de los dispositivos móviles a través del desarrollo del modelo de realidad aumentada para el control de tramos que utilizará varias tecnologías, siendo así este un instrumento alternativo, rápido y eficiente, la cual permitirá brindar información acerca de las rutas del transporte público, contribuyendo así al avance de la informática en el área de sistemas.

1.6.2. TÉCNICA.

El uso de dispositivos móviles se ha ido incrementando 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 opta por la metodología Móvil – D, que es una metodología óptima para el desarrollo de aplicaciones móviles con el cual se realizara el modelo de control de tramos de forma útil para el usuario ofreciéndole usabilidad, accesibilidad y resultados en tiempo real.

1.6.3. ECONÓMICA

En la actualidad 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, la presente investigación no tendrá costo alguno ya que se opta por herramientas que son distribuciones de Software Libre.

1.6.4. SOCIAL

La investigación generara un impacto en lo social por que debido al trameaje, la población entera lleva constantemente quejas por el incumplimiento de rutas por parte de algunos choferes, esto conlleva que se pida un mayor control a las autoridades, para evitar arbitrariedades en el servicio de transporte público. En consecuencia, lo que se pretende es que esta investigación beneficie a la sociedad brindándole información sobre rutas que debe seguir una determinada línea de transporte vehicular, permitiendo decidir al usuario proceder con la denuncia en caso de incumplimiento de la ruta.

1.7. METODOLOGÍA

1.7.1. Método Científico

Para realizar el desarrollo de la investigación existe un método que es de vital importancia es el Método Científico. (Luis Bernal, 2012)

- **Observación.** - Consiste en examinar atentamente los hechos y fenómenos que tienen lugar en la naturaleza y que pueden ser percibidos por los sentidos, en la presente se pretende observar el comportamiento del seguimiento que se realizara al transporte público para identificar los denominados “trameajes”.
- **Formulación de la hipótesis.** - Formular una hipótesis consiste en elaborar una explicación provisional de los hechos observados y de sus posibles causas.
- **Elaboración del Marco Teórico.** - La falta de control de tramos y el desacatamiento de las mismas hace referencia a la necesidad de un Modelo de Realidad Aumentada para el control de tramos esto con el fin de coadyuvar a las autoridades afines del área.

- **Definición de la Investigación.** - Diseñar un Modelo de Realidad Aumentada para el control de tramos para coadyuvar a una correcta toma de decisiones a las autoridades asignadas del transporte público.
- **Establecimiento de la Hipótesis.** - El modelo de realidad aumentada para el control de tramos del transporte público basado en Android y códigos QR, coadyuvara al usuario en el problema denominado “trameaje” para el cumplimiento de ruta del vehículo
- **Selección de la Muestra.** - La muestra que se toma en cuenta para llevar a cabo la investigación hacen referencia a los datos que obtuvieron del registro de comportamiento que tiene el transporte público.
- **Recolección de Datos.** - Los datos a recolectar se enmarcan de acuerdo a la investigación realizada además de la información brindada por el sindicato “Simón Bolívar”.
- **Elaboración del reporte de Investigación.** - El reporte final es considerado como el perfil de tesis basándonos en los resultados de la investigación.

1.7.2. Método de Ingeniería

1.7.2.1. Metodología Mobile-D

Es un método para el desarrollo de Aplicaciones Móviles donde se tienen prácticamente los mismos problemas de desarrollos de software.

El objetivo de este método es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Mobile - D está basado en metodologías conocidas: XP², Crystal,³ y RUP⁴ se compone de distintas fases. (Koslela,2004)

- **Exploración.** Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases.
- **Inicialización.** Se preparan e identifican todos los recursos necesarios. Se establece el entorno técnico.

² Extreme Programming, Programación Extrema, metodología de desarrollo de software.

³ Crystal Methodologies, Metodología Crystal, un conjunto de metodologías para desarrollo de software.

⁴ Rational Unified Process, Proceso Unificado de Rational, un proceso de desarrollo de software.

- **Productización o fase de producto.** Se repiten iterativamente las subfases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del *test driven development*⁵ para conseguir la mayor calidad.
- **Fase de estabilización.** Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente.
- **Fase de pruebas y reparación.** Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente. (Robert Ramírez Vique,2012)

1.8. MÉTRICAS DE CALIDAD DE SOFTWARE

1.8.1. Modelo de QSOS

QSOS⁶ Qualification and Selection of Open Source Software Uno de los modelos que permite la cuantificación y calificación de software, está orientado exclusivamente al producto de software, el mismo se desarrolla en 4 pasos los cuales se describen a continuación:

- ❖ **Definir** y organizar lo que se evaluara (criterios y riesgos comunes de código abierto y funcionalidades específicas del dominio técnico), las cuales serán bases para su análisis.
- ❖ **Evalúe** el software de la competencia según los criterios definidos anteriormente y califique estos criterios individualmente.
- ❖ **Califique** su evaluación organizando los criterios en ejes de evaluación y definiendo el filtrado (ponderaciones, etc.) relacionado con su texto.
- ❖ **Seleccione** el OSS apropiado al calificar todo el software de la competencia utilizando el sistema de filtrado diseñado en el paso 3.

⁵ Test Driven Development, TDD o desarrollo dirigido por las pruebas, indica que antes de realizar una funcionalidad debe existir una prueba que verifique su funcionamiento.

⁶ Qualification and Selection of Open Source Software, Cuantificación y Calificación de Software.

1.8.1.1. Métricas usadas por QSOS

Básicamente la metodología busca dar indicadores sobre la funcionalidad que presenta un software determinado y los riesgos que podrá correr un usuario y/o un proveedor de servicios con dicho software.

1.8.1.2. Tipos de Métricas que usa QSOS

- **Métricas generales:** Que se aplican a todo tipo de Software Libre, además comprende aspectos como madurez, actividad en el desarrollo, portabilidad, entre otras.
- **Métricas Específicas:** Que se aplican a una familia determinada de software, comprenden aspectos esenciales a las características del tipo de software.

1.9. MÉTRICAS DE COSTOS

1.9.1. COCOMO

COCOMO⁷ Modelo Constructivo de Costes, se orienta en estimar los costes de un proyecto de software, y tiene tres submodelos: Básico, Intermedio y detallado.

- **El modelo básico.** - Estima el coste del proyecto –pequeño o mediano- en función de número de líneas de código estimadas. En este modelo, el algoritmo COCOMO establece varios criterios de desarrollo, dependiendo el nivel de dificultad no del nivel de experiencia de los desarrolladores.
- **El modelo intermedio.** - Se utiliza para estimaciones más complejas. Éste incluye 15 atributos –dentro de 4 categorías- del software para determinar el coste del proyecto.
- **Atributos del producto:** garantía de funcionamiento requerida para creación del software, tamaño de la BD, etc.
- **Atributos del ordenador usado:** capacidad de almacenamiento, rapidez del ordenador, etc.

⁷ (Constructive Cost Model) Modelo Constructivo de Costes (B. W. Boehm,)

- **Atributos del personal:** experiencia en el tipo de software a desarrollar, en el lenguaje usado, etc.
- **Atributos del proyecto:** software usado para el desarrollo, lenguaje necesario para crear el software, etc.
- **El modelo detallado.** - Incorpora las características del modelo intermedio y lleva a cabo una evaluación del impacto de los motivantes del coste en cada caso -análisis, diseño, etc.- del proceso de ingeniería del software.

1.10. HERRAMIENTAS

1.10.1. Sistema operativo

Windows un SO que utiliza un núcleo de Windows NT, disponibles para múltiples arquitecturas.

1.10.2. Base de Datos

PostgreSQL es un servidor de Bases de Datos relacionados Orientadas a Objetos, de Software libre bajo licencia BSD, siendo equivalente a otros gestores de Bases de Datos comerciales.

1.10.3. IDE

Eclipse IDE for Java Developers V Mars.2 Release(4.5.2) interfaz para la generación de códigos QR.

1.10.4. IDE Android

Android Studio (Entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android) es un potente editor de códigos, basado en el software IntelliJ IDEA de JETBrains.

1.10.5. Lenguaje de Programación

Java, siendo un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para no tener dependencias de implementación, además de ser el lenguaje base en la programación para dispositivos móviles.

1.10.6. Mapas

Leaflet es una moderna biblioteca JavaScript de código abierto para mapas interactivos adaptables para móviles, una popular opción para crear tu propio mapa deslizable.

1.10.7. Librerías de desarrollo

Java Barcode Generation, es una biblioteca generadora de código de barras Java, el uso es para la generación de código de barras lineales y 2D en J2SE, J2EE entre otras.

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

1.11. LÍMITES Y ALCANCES

1.11.1. Límites

- ✓ La presente investigación se limita a los datos obtenidos del Sindicato “Simón Bolívar”.
- ✓ Abarcará solamente minibuses del transporte público.
- ✓ El actual trabajo de investigación no es producto de análisis estadístico o consultas (queries).
- ✓ Se limitará en la presentación de dos sistemas de software: la aplicación móvil y el generador de códigos QR.

1.11.2. Alcances

- ✓ Se reconocerá una ruta establecida con más trameaje, establecido por el Sindicato “Simón Bolívar”.
- ✓ Diseñar el Modelo de Realidad Aumentada para el control de tramos
- ✓ Aplicar el Modelo de Realidad Aumentada para el control de tramos
- ✓ El modelo de Realidad Aumentada para el control de tramos se encargará del reconocimiento de los códigos QR.

- ✓ La aplicación móvil tendrá un módulo donde se pueda emitir la denuncia correspondiente, en caso de incumplir la ruta determinada.

1.12. APORTES

La presente investigación está orientada a la propuesta de un Modelo de Realidad Aumentada para control de tramos, con el fin de estudiar y analizar el comportamiento de los datos manera cotidiana, de esta manera se proveerá de información concreta y oportuna a las autoridades y usuarios, y tras la toma de decisiones realizada por las autoridades correspondientes bajar los índices del denominado “trameaje”, que son perjudiciales para la población en general.



CAPÍTULO II

MARCO TEORICO

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

El rápido desarrollo de la tecnología que se ha producido en las últimas décadas a nivel global, está generando modificaciones en las condiciones sociales con la incorporación de los nuevos avances en la ciencia y las técnicas, lo que está afectando a la vida cotidiana de las personas (Innerarity, 2016). En estos numerosos progresos tecnológicos se encuentra la incorporación de los dispositivos móviles, con una alta capacidad de procesamiento y almacenamiento de datos, sensores de datos específicos y vías de despliegue gráfico que permiten nuevas formas de representación de la realidad con la transformación de los datos en información con valor añadido. En este sentido, técnicas como la Realidad Aumentada han aportado, la configuración de una forma de entender y modelar la realidad sobre los lugares y los territorios (Shekhar, Feiner, & Aref, 2016). De esta manera, casi todos los fenómenos que se producen en un contexto de espacio físico, se ven influenciados por el marco que establecen las TIC⁸, abarcando desde la simple representación o modelación de los escenarios espaciales, hasta la posibilidad de generar y estructurar lugares desde nuevas y complejas estructuras de datos, con el desarrollo de técnicas de procesamiento y metodologías de despliegue gráfico.

2.2. MODELO

Un modelo es un objeto, concepto o conjunto de relaciones que se utiliza para representar y estudiar de forma simple y comprensible una porción de la realidad empírica (Ríos, 1995).

Un modelo es la representación del conjunto de relaciones que definen un fenómeno, con miras a su mejor comprensión. Aunque difieren con relación a su valor explicativo, todos los modelos comparten la característica de ser imágenes o representaciones construidas acerca de lo que podría ser la multiplicidad de

⁸ Tecnologías de la Información y la Comunicación

fenómenos o cosas observables reducidas a una raíz común que permita captarlas como similares en su estructura o al menos en su funcionamiento. (Flórez, 1999)

El término modelo puede ser definido como la representación de un hecho o fenómeno propuesta como ideal a seguir. Pretende mostrar las características generales de la estructura de dicho fenómeno, explicar sus elementos, mecanismos y procesos, cómo se interrelacionan y los aspectos teóricos que le dan sustento, para facilitar su comprensión. Una vez comprendido el concepto de modelo, conoceremos cuál es su función.

2.3. REALIDAD AUMENTADA

La incorporación de datos e información digital (textos, imágenes, audios, vídeos, modelos 3D) a un entorno real por medio del reconocimiento de patrones que se lleva a cabo gracias a un software instalado en un dispositivo preparado. Una tecnología que permite superponer elementos virtuales sobre nuestra visión de la realidad. (Thomas p. Caudell, 1990)

Se establece como la combinación de información real y virtualizada por un ordenador, efectuando una fusión en tres dimensiones con el fin de generar un modelo digital observable (Azuma et al., 1997)

La Realidad Aumentada nos permite añadir capas de información visual sobre el mundo real que nos rodea, utilizando la tecnología, dispositivos como pueden ser nuestros propios teléfonos móviles. Esto nos ayuda a generar experiencias que aportan un conocimiento relevante sobre nuestro entorno, y además recibimos esa información en tiempo real, básicamente podemos diferenciar dos tipos de Realidad Aumentada entre otros:

2.3.1. Realidad Aumentada con Marcadores

Los marcadores son símbolos impresos en papel o imágenes sobre las cuáles se superponen los elementos virtuales. Este contenido adicional aparece cuando la app de Realidad Aumentada asociada reconoce el marcador y activa la

experiencia. Para que funcione correctamente, es necesario que el marcador se encuentre en una superficie plana y que el dispositivo mantenga una distancia adecuada. (Cono, 2018)

El usuario trabaja con una o más imágenes llamadas “marcadores”, cuando son reconocidos por la cámara (por un software determinado) se superpone algún tipo de información. “Los marcadores son unas imágenes en blanco y negro, generalmente cuadradas, con dibujos sencillos y asimétricos” (Estebanell et al., 2012)

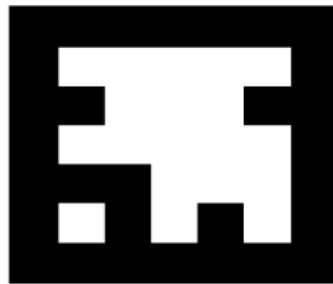


Figura 2. 1 Marcador (Estebanell,2012)

2.3.2. Realidad Aumentada con Geolocalización

Se trabaja con la cámara de un dispositivo que cuenta con brújula, GPS y acelerómetro. Estas tres herramientas nos permiten saber en qué punto del planeta estamos, a qué altura y en qué dirección está apuntando la cámara del dispositivo. Cuando se determina el lugar en el que nos encontramos, un software nos dará información de este punto del planeta, esta información la sacará de internet y nos aparecerá en la pantalla del dispositivo.



Figura 2. 2 Geolocalización (Estebanell, 2012)

2.4. GEOLOCALIZACIÓN

La geolocalización es la capacidad de asignar coordenadas geográficas a la información por medio de herramientas informáticas. La generalización de la tecnología GPS en dispositivos de uso personal como los teléfonos móviles y ordenadores personales ha permitido que esta capacidad esté al alcance de cualquier ciudadano, y como consecuencia, el desarrollo de aplicaciones a distintos campos. (Rodríguez E., 2014)

La geolocalización consiste en situar un objeto en el espacio mediante un sistema de coordenadas: latitud, longitud y altitud”. Este proceso es utilizado por los Sistemas de Información Geográfica también conocidos por siglas en inglés GIS, estos sistemas se encargan de integrar hardware, software y datos geográficos. (Rodríguez E., 2014)

Existen tres elementos clave que componen la geolocalización:

2.4.1. Hardware

Dispositivo que actúa como plataforma en la que se desarrolla el proceso de geolocalización como 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 incorpora los mecanismos necesarios para permitir dicha localización.

2.4.2. 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 localizaciones geográficas, y la asociación de ambos elementos.

2.4.3. 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 (Julio Alcón, 2007)

2.4.4. Geolocalización, localizar y GPS.

La semejanza entre estos tres términos puede dar lugar a confusiones, por lo que a continuación aclararemos en qué consiste cada uno.

- **Geolocalización:** consiste básicamente en determinar la posición de un objeto en el espacio, este concepto que hace referencia a la situación que ocupa un objeto en el espacio y que se mide en coordenadas de latitud (x), longitud (y) y altura (z). Este término es un nuevo concepto que se ha puesto de moda en Internet también conocido como georreferenciación.
- **Localización:** es un término genérico que hace referencia a la acción de determinar el emplazamiento de alguien o algo mediante coordenadas geográficas, pero no se trata de una definición en absoluto, sino más bien de un intento de acercamiento a comprender el término.
- **GPS:** El Sistema de Posicionamiento Global o GPS, es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto móvil, podemos alcanzar una precisión hasta de centímetros usando el GPS, pero lo habitual son unos pocos metros, midiendo la distancia desde un grupo de satélites a una posición cualquiera de la Tierra pueden calcularse las coordenadas exactas de dicha posición.
(Beltrán G., 2012)

2.5. CONTROL

Henry Fayol. Consiste en verificar si todo ocurre de conformidad con el plan adoptado, con las instrucciones emitidas y con los principios establecidos. Tiene como fin señalar las debilidades y errores a fin de rectificarlas e impedir que se produzcan nuevamente. (Henry Fayol, 2012)

El proceso de medir los actuales resultados en relación con los planes, diagnosticando la razón de las desviaciones y tomando las medidas correctivas necesarias. (Robert B Buchele,1950)

Control es el proceso por el cual los procesos se aseguran que la obtención y el empleo de los recursos se efectúen en forma efectiva y eficiente, en el logro de los objetivos de la organización.

2.6. SISTEMA DE CONTROL

Un sistema de control es un conjunto de mecanismos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema, con el fin de reducir las probabilidades de fallo y obtener los resultados deseados. (Juan Martín García,2017)

Se define como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado.

2.7. TRAMO

La etimología de la palabra tramo se encuentra en el latín “tramitis” que se aplica a un camino o vía transitable, que es una porción de un trayecto mayor.

Un tramo es una porción de superficie o terreno, que aparece de algún modo señalizada como subdivisión del mismo. (Espasa-Calpe, 2005)

Se comprende entre dos puntos (inicio y final) que forman parte de una línea que se desarrolla linealmente, especialmente un camino o una vía.

2.8. TRANSPORTE PÚBLICO

Transporte público o transporte en común es el término aplicado al transporte colectivo de pasajeros. A diferencia del transporte privado, los viajeros del transporte público tienen que adaptarse a los horarios y a las rutas que ofrezca el operador y dependen en mayor o menor medida de la intervención regulatoria del Gobierno. Un sistema de medios (vehículos) para llevar personas de un lugar a otro de la ciudad. (Illich, 1974)

En la búsqueda de transportes personales para cada estilo de vida diferente, la elección de uno o una combinación de varios modos de transporte permite a los individuos una cierta flexibilidad para desplazarse en la ciudad. De esta manera, la movilidad se ha convertido en una parte esencial de la vida moderna. Como se refiere a los flujos en el contexto urbano (Freudental Pedersen, 2009).

El transporte público es un sistema integral de medios de transporte de uso generalizado, capaz de dar solución a las necesidades de desplazamientos de las personas.

2.8.1. Características del Transporte Público

- **Operación del transporte:** Punto de vista del prestatario de transporte. Incluye el cumplimiento de horarios, frecuencias, asignación de roles y jornadas de trabajo, supervisión, operación y mantenimiento de las unidades de transporte.
- **Servicio de transporte:** Forma en que el usuario, eventual y potencial ve el transporte. Integra conceptos tales como calidad y cantidad del servicio, información que se le proporciona, costo, tiempos de viaje, etc.
- **Gobernanza:** En nuestro caso la provincia y/o zonas considerados área rural, el municipio o un ente creado a tal fin, es quien concede los servicios a terceros o lo presta por administración, garantiza el cumplimiento de los contratos, sanciona incumplimientos, planifica y regula los servicios de transporte.

2.8.2. Componentes Físicos del Transporte Público

Un sistema de transporte se compone principalmente de tres elementos físicos:

- **Vehículo:** Unidades de transporte, su conjunto se describe como parque vehicular en el caso de autobuses, unidad de transporte: un solo vehículo o un agrupamiento de vehículos que formen un tren y operen conjuntamente como uno solo.
- **Infraestructura:** Derecho de vías en que operan el transporte público, sus paradas y/o estaciones. Estaciones normales, terminales, puntos de trasbordo, garajes, depósitos, encierros o patios, talleres de mantenimiento y reparación.

- **Red de transporte:** está compuesta por las rutas de los minibuses y buses, las líneas de servicio con la que trabaja.

Gustavo Luis Pastor, P. (2017). Medios de transporte Urbano. Catedra Doctoral. Recuperado de <http://ingenieria.uncuyo.edu.ar/catedras/u1-medios-de-transporte-urbano>

2.9. 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. Android permite programar aplicaciones en una variación de Java llamada Dalvik⁹. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono como son GPS¹⁰, llamadas, sms, agenda, entre otras, de una forma muy fácil en un lenguaje de programación muy popular como es Java. (M. L.Tapia, 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 hardware, software y operadores de servicio. 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 descargada de sitios de terceros. (O. Pedrozo, 2012)



Figura 2. 3 Logo Android (M.Tapia,2013)

⁹ Es la máquina virtual que utiliza la plataforma para dispositivos móviles Android.

¹⁰ Global Positioning System, Sistema de Posicionamiento Global, un sistema que determina la posición de cualquier objeto y/o persona.

2.9.1. Características

Android es una plataforma para dispositivos móviles que contiene una pila de software donde se incluye un sistema operativo, Middleware¹¹ y aplicaciones básicas para el usuario. Su diseño cuenta, entre otras, con las siguientes características: (Jaime Aranz, 2009)

- ✓ Busca el desarrollo rápido de aplicaciones, que sean reutilizables y verdaderamente portables entre diferentes dispositivos.
- ✓ Los componentes básicos de las aplicaciones se pueden sustituir fácilmente por otros.
- ✓ Cuenta con su propia máquina virtual, Dalvik, que interpreta y ejecuta código escrito en Java.
- ✓ Permite la representación de gráficos 2D y 3D.
- ✓ Posibilita el uso de bases de datos.
- ✓ Soporta un elevado número de formatos multimedia.
- ✓ Servicio de localización GSM¹².
- ✓ Controla los diferentes elementos hardware: Bluetooth, Wi-Fi, cámara fotográfica o de vídeo, GPS, acelerómetro, infrarrojos, etc...
- ✓ Cuenta con un entorno de desarrollo muy cuidado mediante un SDK¹³ 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.

2.9.2. Arquitectura

Los componentes principales del sistema operativo de Android se describen continuación a detalle: (O. Pedrozo, 2012)

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y

¹¹ Software que se sitúa entre un sistema operativo y las aplicaciones que se ejecutan en él.

¹² Global System for Mobile communications, Sistema Global para las Comunicaciones Móviles

¹³ Software Development Kit, Kit de Desarrollo de Software, reúne un grupo de herramientas que permiten la programación de aplicaciones móviles.

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 del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework).
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. este ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima.

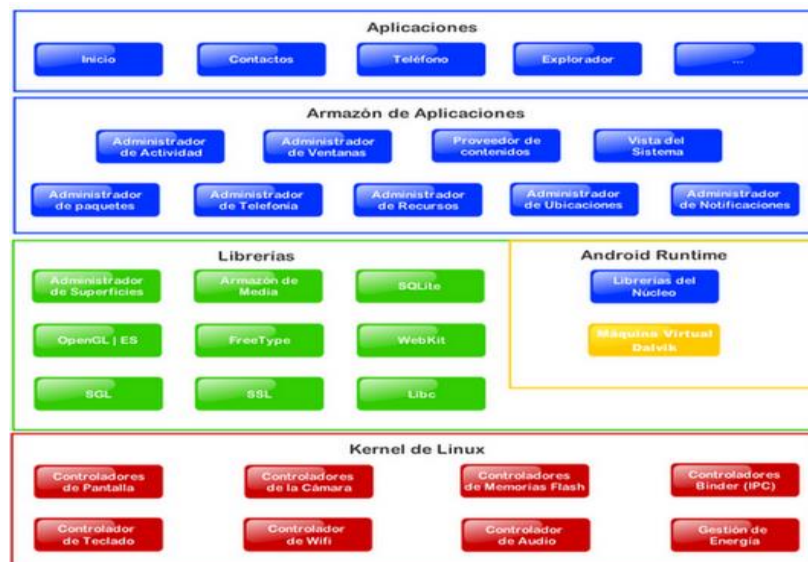


Figura 2. 4 Arquitectura de Android (Jaime Aranz, 2009)

- **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.

2.9.3. La máquina virtual Dalvik

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 ME¹⁴ (M. L.Tapia, 2013).

2.10. CÓDIGO QR “QUICK RESPONSE CODE”

Un código QR se define como una matriz en dos dimensiones formada por una serie de cuadrados negros sobre fondo blanco. Esta matriz es leída por un lector específico (Lector de QR) en nuestro dispositivo móvil y de forma inmediata nos lleva a una aplicación en internet ya sea un mapa de localización, un correo electrónico, una página web, entre otras. (Mario García, 2016)

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

¹⁴ Java Platform Micro Edition, Una plataforma que se adapta a entornos limitados y hace posible la creación de aplicaciones Java que se ejecuten en pequeños dispositivos con memoria, visualización y potencia limitadas.

de barras bidimensional. Fue creado en 1994 por la compañía japonesa Denso Wave, subsidiaria de Toyota. Presenta tres cuadrados en las esquinas que permiten detectar la posición del código al lector. El objetivo de los creadores un equipo de dos personas en Denso Wave, fue que el código permitiera que su contenido se leyera a alta velocidad. (Wikipedia, 2015)

Un método de representación y almacenamiento de información en una matriz de puntos bidimensional cuadrada que puede almacenar los datos codificados, la mayoría del tiempo los datos son enlaces.

2.10.1. Características Generales

Aunque inicialmente se usó para registrar repuestos en el área de la fabricación de vehículos, hoy los códigos QR se usan para administración de inventarios en una gran variedad de industrias. La inclusión de software que lee códigos QR en teléfonos móviles ha permitido nuevos usos orientados al consumidor, que se manifiestan en comodidades como el dejar de tener que introducir datos de forma manual en los teléfonos, además de leerse desde computadores personales, teléfonos inteligentes o tabletas mediante dispositivos de captura de imagen.

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. (Wikipedia, 2015)

2.10.2. 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 2.5 se presenta los módulos que están ubicados en una estructura cuadrada, que contiene 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 datos

codificados, sino información necesaria para su decodificación, son los denominados patrones de función, y existen de varios tipos:

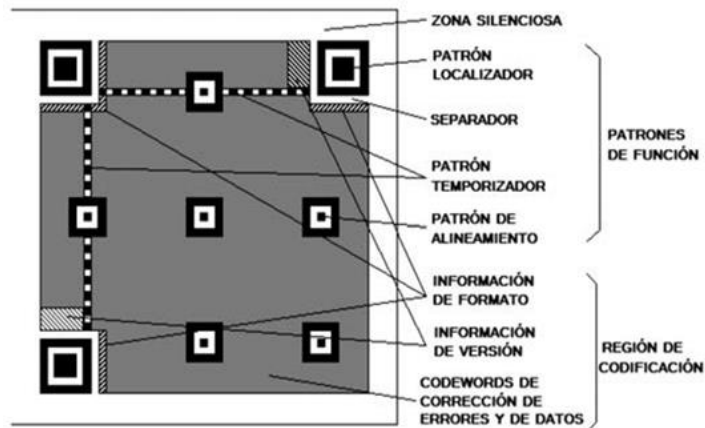


Figura 2. 5 Estructura Genérica de Símbolo QR (Luque,2015)

- **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 negros que ayuda a calcular las coordenadas de los módulos del símbolo.
- **Patrón temporizador:** patrón de función que permite re sincronizar 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 formato es un patrón codificado que contiene información sobre el grado de corrección de errores con el que se han codificado los datos de la región de codificación y el tipo de máscara que

se les ha aplicado. La información de versión, por su parte, es un patrón codificado que contiene información que indica la versión del símbolo. Asimismo, para poder delimitar correctamente los bordes de cada símbolo, se requiere de una banda de anchura 4 módulos para la zona silenciosa que debe estar en blanco y en negro si hay reflectancia inversa. El tamaño del símbolo se denomina versión véase figura 2.6, existen 40 versiones: la versión 1 tiene 21x21 módulos, la versión 2 tiene 25x25 módulos, y así sucesivamente existe incremento de 4 módulos por lado en cada versión hasta la versión 40, que contiene 177x177 módulos.

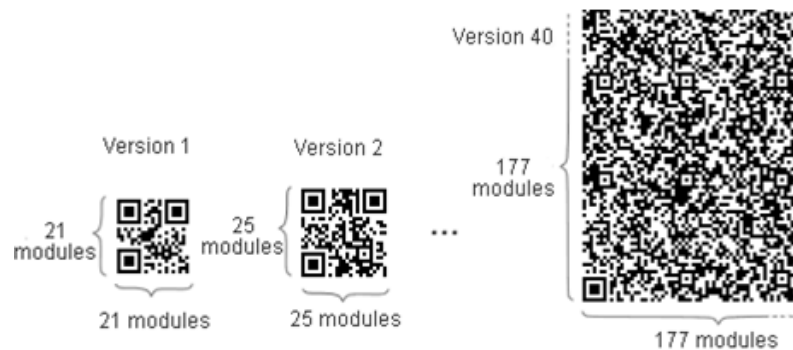


Figura 2. 6 Versiones de los Códigos QR (QR Code, 2011)

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)

2.10.3. Corrección de errores

Según los códigos QR emplean codificación de errores basada en algoritmos de Reed-Solomon, generando un conjunto de ECC¹⁵ que se añaden a los de datos aportando redundancia. Los algoritmos Reed-Solomon fueron desarrollados inicialmente para combatir el ruido de comunicaciones en los satélites artificiales y sondas espaciales, y hoy día se emplean por ejemplo también en la codificación

¹⁵ Error Correction Codewords, Corrección de Errores Codewords

de CDs de música. Permiten corrección a nivel de byte y son adecuados para errores de ráfaga. Existen 4 niveles de corrección de errores en los símbolos QR: (Luke, 2015)

L (Low). Puede corregir hasta el 7% de los codewords de datos del símbolo.

M (Medium). Puede corregir hasta el 15% de los codewords de datos del símbolo.

Q (Quality). Puede corregir hasta el 25% de los codewords de datos del símbolo.

H (High). Puede corregir hasta el 30% de los codewords de datos del símbolo.

2.10.4. Área de impresión reducida

Los códigos QR, requieren de un área de impresión mucho más reducida como se muestra en la figura 2.7, beneficiando costos de impresión y facilitando su lectura, son capaces de almacenar la misma cantidad de datos que un código de barras convencional en un tamaño 10 veces más chico aproximadamente. Incluso, otra versión de código QR es más pequeña aún conocida como Micro QR. (QRCode, 2011)



Figura 2. 7 Comparativa espacio de impresión código de barras vs QR (QR Code, 2011)

2.10.5. 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 varios cientos de veces más información.

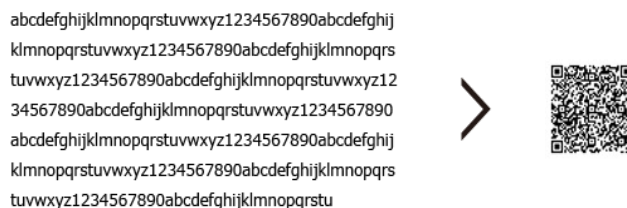


Figura 2. 8 Vista general Código QR (QR Code, 2011)

QR tiene capacidades máximas de datos, es capaz de manejar todos los tipos de datos, tales como caracteres alfabéticos, numéricos, códigos de Kanji, Kana (silabarios japoneses), símbolos, binarios y de control. Hasta 7.089 caracteres pueden ser codificados en un símbolo. (QRCode, 2011)

Tabla 2. 1
Descripción de capacidad Código QR

Capacidad de datos del código QR	
Numérico	Máx. 7.089 caracteres
Alfanumérico	Máx. 4.296 caracteres
Binario	Máx. 2.953 bytes
Kanji/Kana	Máx. 1.817 caracteres

Nota: Wikipedia (2015)

2.10.6. Resistentes a suciedad o daño parcial

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



Figura 2. 9 Daños en un Código QR impreso (QR, 2011)

2.10.7. Legibles desde cualquier dirección en 360°

Los códigos QR permiten una lectura omnidireccional¹⁶ de rápida respuesta. Esto es posible gracias a ciertos patrones de alineamiento presentes en 3 de las esquinas que conforman al código QR.

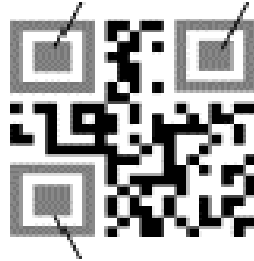


Figura 2. 10 Patrones de detección de posición (QR Code, 2011)

Estos patrones garantizan lectura rápida y estable del código disminuyendo los efectos negativos de la posible interferencia del fondo. (QRCode, 2011)

2.10.8. 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.

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



Figura 2. 11 Proceso de Etiquetado Movil (Ramonda, 2014)

¹⁶ Se puede utilizar en todas las direcciones o sentidos.

Una vez que tomamos una fotografía con nuestro celular, el software procesa la imagen y permite decodificarla en texto plano. Esto nos permite, por ejemplo, codificar una dirección URL, que justamente es la aplicación (fuera de la industrial) que mayor impacto tiene en el uso cotidiano de estos códigos (Ramonda, 2014)

2.11. INGENIERÍA DE SOFTWARE

La ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972)

La aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976)

En tal sentido se define a una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería. Brindando soporte operacional y de mantenimiento, el estudio de las aplicaciones.

2.11.1. Características

La ingeniería de software es una tecnología con varias capas. Como se aprecia en la figura 1.3, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad la administración total de la calidad, es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad. El fundamento para la ingeniería de software es la capa proceso. El proceso de ingeniería de software es el aglutinante que une las capas de la tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software. El proceso de software forma la base para

el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.



Figura 2. 12 Capas de la Ingeniería de Software (Pressman, 2002)

Los **métodos** de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas.

Las **herramientas** de la ingeniería de software proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software.

2.11.2. El Proceso del Software

Un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que

hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo.

La estructura del proceso establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de actividades estructurales que sean aplicables a todos los proyectos de software los cuales constan de cinco actividades.

- **Comunicación.** - Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.
- **Planeación.** - La actividad de planeación crea un “mapa” que guía al equipo, el mapa —llamado plan del proyecto de software— define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.
- **Modelado.** - Crea un “bosquejo” del objeto por hacer a fin de entender el panorama general, un ingeniero de software hace determina un bosquejo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará.
- **Construcción.** - Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
- **Despliegue.** - El software se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación. (Roger S. Pressman,2002)

2.12. METODOLOGÍA MOBILE-D

Se desarrolló junto con un proyecto finlandés en el 2004. Fue realizado, principalmente, por investigadores de la VTT¹⁷, el objetivo es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en menos de diez semanas. Se trata de un conjunto de métodos basado en soluciones conocidas y consolidadas. (Robert Ramirez Vique,2012)

El objetivo de este método es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Mobile-D está basado en metodologías conocidas y aplicadas de forma estricta, se compone de distintas fases las cuales se detallan a continuación:

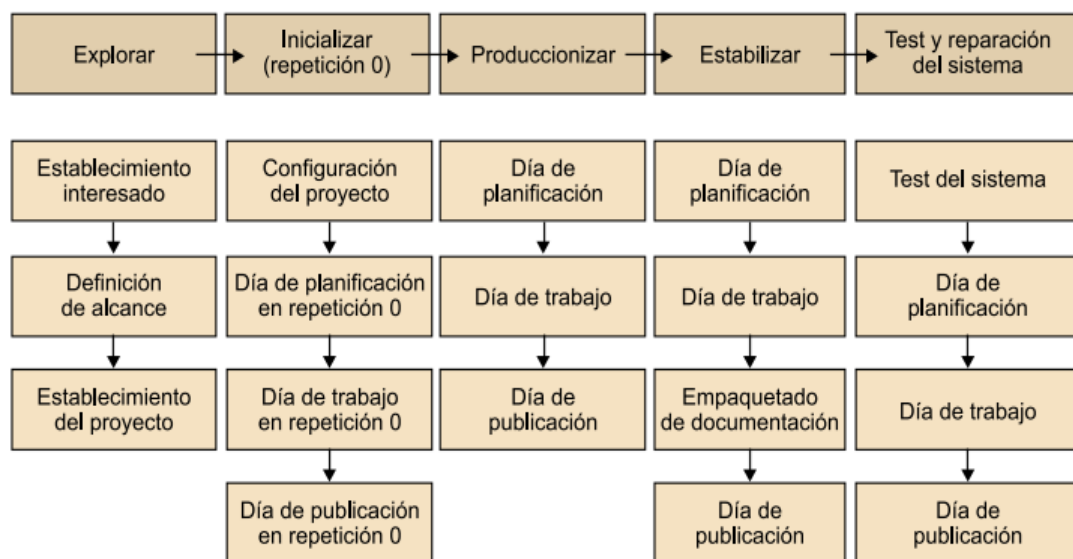


Figura 2. 13 Ciclo de desarrollo Mobil-D (Vique, 2012)

Cada fase (excepto la inicial) tiene siempre un día de planificación y otro de entrega. Las fases son:

¹⁷ Instituto de Investigación Finlandés

2.12.1. Exploración

El equipo de desarrollo debe generar un plan y establecer las características y los conceptos básicos que están alrededor del proyecto. Este proceso se realiza en tres etapas:

- **Establecimiento de actores.** - El propósito de esta tarea es reconocer que el grupo de clientes tiene la experiencia, conocimiento del dominio y de los requisitos para el desarrollo del producto software.
- **Definición del alcance.** - El propósito de esta etapa es definir los objetivos para el proyecto, así como la línea de tiempo, el ritmo.
- **Establecimiento de proyectos.** - El propósito de esta etapa es definir y asignar los recursos (tanto técnicos como humanos) que se necesita para que el proyecto de desarrollo de software inicie.

2.12.2. Inicialización

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. (Koslela, 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 todos ellos estén en plena preparación para la

implementación de los requisitos seleccionados por el cliente durante las próximas fases del proyecto.

2.12.3. Producto

Se repite la programación de tres días (planificación trabajo liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano de ahí el nombre de esta técnica de Test Driven Development, TDD. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

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 de las iteraciones, los criterios de ingreso:

- La fase de iteración precedente ha sido completada.
- Se identifican los requisitos más importantes.
- El equipo se ha reunido y 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.

2.12.4. Estabilización

Se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente. Esta será la fase más importante en los

proyectos multi-equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desarrollar en la fase de productización, aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

Los objetivos del modelo de fase estabilizar son:

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

Los criterios de ingreso:

- La funcionalidad esencial y valor de negocio del proyecto ha sido completado y se muestra al cliente.

2.12.5. 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. (Koslela, 2004)

Los objetivos del sistema de prueba y corrección con:

- Pruebe el sistema en base a la documentación presentada
- Proporcionar información de los defectos encontrados.
- Corregir los defectos.
- Producir como error de sistema libre como sea posible.

2.13. MODELO DE QSOS

QSOS es un modelo que describe el flujo de trabajo para evaluar proyectos y componentes de código abierto, así como un conjunto de herramientas para aplicar el método. Este modelo está sustentado en una comunidad que mantiene y desarrolla todo el conjunto. El modelo sistematiza todo este proceso de

evaluación en base a cuatro pasos independientes e iterativos: definición del ámbito a evaluar, evaluación, calificación y selección. (QSOS, 2013)

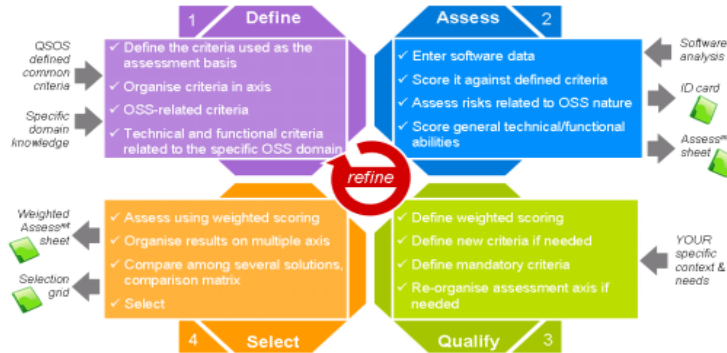


Figura 2. 14 Proceso QSOS (Pilot, 2006)

2.13.1. Definición

Construcción y enriquecimiento de los marcos de referencia que serán utilizados en los pasos siguientes. El objetivo de este paso es definir varios elementos de la tipología a ser utilizada por los 3 pasos que siguen. Los marcos de referencia son:

- **Familia de Software:** Este aspecto responde la pregunta “¿Qué tipo de software estamos analizando?”.
- **Tipos de Licencia:** Clasificación de las licencias más comunes de Software Libre y de código abierto.
- **Tipos de comunidades:** Clasificación de las comunidades que pueden desarrollar Software Libre.

2.13.2. Evaluación

Evaluación del software hecho de acuerdo a 3 ejes de criterios: cobertura funcional, riesgos del usuario y riesgos del proveedor de servicios.

Este paso tiene como objetivo la colección de información por parte de las comunidades de código abierto. Esta evaluación comprende la elaboración de la **tarjeta de identificación del software**, así como la elaboración de la **hoja de evaluación del software**.

- **Tarjeta de identificación del software.** - Contiene datos y hechos acerca del software, es utilizada como base para el proceso de evaluación. Contiene elementos como nombre, fechas de creación, tipo de software autores, descripción general, los servicios que presenta, aspectos técnicos y funcionales, entre otros, se identifica como:

Tabla 2. 2
Descripción de capacidad Código QR

Información general	Servicios existentes	Aspectos técnicos y funcionales
<ul style="list-style-type: none"> •Nombre del software •Referencia, fecha de creación, fecha de elaboración de esta tarjeta •Autor •Tipo de software 	<ul style="list-style-type: none"> •Documentación •Entre otros 	<ul style="list-style-type: none"> •Tecnologías de implementación •Funcionalidades detalladas

Nota: Oliver (2006)

- **Hoja de evaluación del software.** - Contempla la identificación, descripción y análisis en detalle de cada versión que se presenta del software. Con un puntaje de 0 al 2

Tabla 2. 3
Hoja de Evaluación de software

Cobertura funcional	Determinada por la definición establecida en el paso de Definición.
Riesgos perspectiva usuario	Cuando escoge una solución de Software Libre u Open Source.
Riesgos perspectiva proveedor.	que utilice dicha solución de software.

Nota: Oliver Pilot,(2006)

2.13.3. Calificación

Carga de los criterios divididos en ejes, modelando el contexto (requerimientos de usuario y/o estrategia escogida por el proveedor de servicios). El objetivo de este paso es definir los filtros que traduzcan las necesidades y restricciones relacionadas con la selección del software de código abierto en un contexto específico. Para ello se definen niveles de filtros sobre el software en base.

Tenemos cuatro tipos de filtros:

- ✓ Filtros sobre la tarjeta de identificación.
- ✓ Filtros sobre las funcionalidades.
- ✓ Filtros sobre los riesgos desde la perspectiva del usuario.
- ✓ Filtros sobre los riesgos desde la perspectiva del proveedor de servicios

2.13.4. Selección

Aplicación del filtro configurado en el paso anterior a los datos encontrados en los dos primeros pasos, de manera de realizar consultas, comparaciones y selección de productos.

Este paso tiene como objetivo identificar el software que contenga y satisfaga los requerimientos de usuario, puede ser de dos modos: un modo estricto (selección estricta), y otro un poco más holgado (selección holgada).

- ✓ **La selección estricta** se basa en la eliminación del software tan pronto como el software no cumpla con lo formulado en el paso de Calificación. Este método es muy restrictivo y puede no seleccionar software alguno.
- ✓ **La selección holgada** se basa en darle puntuación nuevamente al software dependiendo de lo obtenido en el paso de Evaluación. Al final se escoge el software con más (o menos) puntos. (Oliver Pilot,2006)

2.14. COCOMO

El Modelo Constructivo de Costes (Constructive Cost Model) fue desarrollado por B. W. Boehm a finales de los 70 y comienzos de los 80 (Prentice-Hall, 981).

Un modelo de formulación matemática con un fuerte componente de base empírica, principalmente utilizado para estimación de costos en los proyectos de software. Entre otras características, el modelo está orientado a la magnitud del producto final, está basado en estimaciones matemáticas, mide el “tamaño” del proyecto y utiliza las líneas de código como unidad de medida. Dos de los aspectos fundamentales del modelo COCOMO son los submodelos y los modos de desarrollo. (Boehm, 1981)

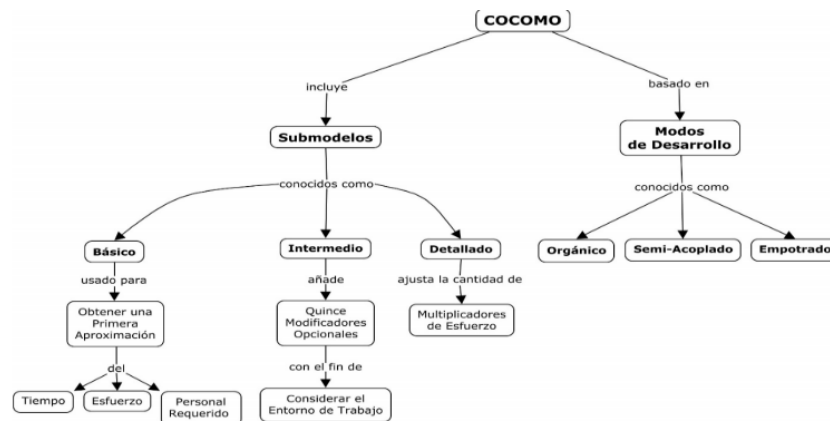


Figura 2. 15 Conceptualización básica COCOMO (Garita, 2016)

2.14.1. Modos de Desarrollo

- ✓ **Orgánico.** - Proyectos relativamente sencillos, menores de 50 KDLC¹⁸, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- ✓ **Semi-Acoplado.** - Proyectos intermedios en complejidad y tamaño, menores de 300 KDLC, donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- ✓ **Empotrado.** - Proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica.

¹⁸ Es la cantidad de líneas de código.

<i>Modo de desarrollo</i>	<i>Requisitos</i>	<i>Tamaño</i>	<i>Complejidad</i>	<i>Personas</i>	<i>Experiencia</i>
Orgánico	Poco rígidos	Pequeño (<50KLDC)	Pequeña	Pocas	Mucha
Semiacoplado	Poco/medio	Medio (50 a 300KLDC)	Medio	Medio	Medio
Empotrado	Alto	Grande (>300KLDC)	Alta	Alta	Poca

Figura 2. 16 Esquema de modos de desarrollo de software (Boehm, 1981)

Estos modos de desarrollo permiten utilizar cuatro valores constantes. En la Tabla 5. se muestran los modos de desarrollo y los valores constantes respectivos. Estos valores constantes, codificados aquí como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo.

<i>Modo de desarrollo</i>	COCOMO Básico <i>a</i>	COCOMO Intermedio <i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Figura 2. 17 Valores constantes por modo de desarrollo (Boehm,1981)

2.14.2. Modelos de Estimación (Básico, Intermedio y Detallado)

- ✓ **Modelo Básico.** – Este modelo trata de estimar, de una manera rápida el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (Cost drivers): el tamaño del software y el modo de desarrollo, las ecuaciones básicas son:

Esfuerzo:

$$PM = A \times (KSLOC)^B$$

Donde:

- **PM** es el esfuerzo estimado. Representa los meses-persona³ necesarios para ejecutar el proyecto
- **KSLOC** es el tamaño del software a desarrollar en miles de líneas de código
- **A y B** son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semiacoplado, Empotrado)

Cronograma:

$$TDEV = C \times (PM)^D$$

Donde:

- **TDEV** representa los meses de trabajo que se necesitan para ejecutar el proyecto
- **C y D** son coeficientes que varían según el Modo de Desarrollo (Orgánico, Semiacoplado, Empotrado)

La Tabla 2.4 muestra la variación de la fórmula de estimación de esfuerzo y cronograma para los tres Modos de Desarrollo:

Tabla 2. 4
Ecuaciones del Modelo Básico de COCOMO.

Modo de Desarrollo	Esfuerzo	Cronograma
Orgánico	$PM=2.4 \times (KSLOC)^{1.05}$	$TDEV=2.5 \times (PM)^{0.38}$
Semiacoplado	$PM=3.0 \times (KSLOC)^{1.12}$	$TDEV=2.5 \times (PM)^{0.35}$
Empotrado	$PM=3.6 \times (KSLOC)^{1.20}$	$TDEV=2.5 \times (PM)^{0.32}$

Nota: Boehm (1981)

- ✓ **Modelo Intermedio.** – En este modelo se introducen 15 atributos de coste para tener en cuenta el entorno de trabajo. Estos atributos se utilizan para ajustar el coste nominal del proyecto al entorno real, incrementando la previsión de la estimación. Además, que provee un detalle y precisión

superior, por lo cual es más apropiado para la estimación de costos en etapas de mayor especificación.

Atributos del producto de software

- RELY Confiabilidad Requerida
- DATA Tamaño de la Base de Datos
- CPLX Complejidad del Producto

Atributos del hardware

- TIME Restricción del Tiempo de Ejecución
- STOR Restricción del Almacenamiento Principal
- VIRT Volatilidad de la Máquina Virtual*
- TURN Tiempo de Respuesta de la computadora expresado en horas

Atributos del personal involucrado en el proyecto

- ACAP Capacidad del Analista
- AEXP Experiencia en Aplicaciones Similares
- PCAP Capacidad del Programador
- VEXP Experiencia en la máquina virtual
- LEXP Experiencia en el Lenguaje de Programación

Atributos propios del proyecto

- MODP Prácticas Modernas de Programación
- TOOL Uso de Herramientas de Software
- SCED Cronograma de Desarrollo Requerido

El proceso de estimación del esfuerzo puede sintetizarse en los siguientes pasos:

- Se calcula el esfuerzo nominal $PM_{Nominal}$, al igual que en el modelo Básico, donde los únicos factores de costo son el tamaño y el modo de desarrollo.
- Se determina el Factor de Ajuste del Esfuerzo (EAF, Effort Adjustment Factor) según la fórmula:

$$EAF = \prod_{i=1}^{15} EM_i$$

Donde cada EM, llamado factor multiplicador de esfuerzo, es el valor que corresponde a cada atributo de acuerdo al grado de influencia (Muy Bajo, Bajo, Nominal, Alto, Muy Alto, Extra Alto) en el esfuerzo del desarrollo del software. Finalmente, se ajusta el esfuerzo nominal aplicando el EAF.

$$PM = A \times EAF \times (KSLOC)^B$$

La Tabla 2.5 muestra la variación de la ecuación de estimación de esfuerzo y cronograma según los tres modos de desarrollo.

Tabla 2. 5
Ecuaciones del Modelo Intermedio de COCOMO

Modo de Desarrollo	Esfuerzo Nominal	Esfuerzo Ajustado	Cronograma
Orgánico	$PM_{nominal} = 3.2 \times (KSLOC)^{1.05}$	$PM = 3.2 \times EAF \times (KSLOC)^{1.05}$	$TDEV = 2.5 \times (PM)^{0.38}$
Semiacoplado	$PM_{nominal} = 3.0 \times (KSLOC)^{1.12}$	$PM = 3.0 \times EAF \times (KSLOC)^{1.12}$	$TDEV = 2.5 \times (PM)^{0.35}$
Empotrado	$PM_{nominal} = 2.8 \times (KSLOC)^{1.20}$	$PM = 2.8 \times EAF \times (KSLOC)^{1.20}$	$TDEV = 2.5 \times (PM)^{0.32}$

Nota: Boehm(1981)

- ✓ **Modelo Detallado.** - Este modelo puede procesar todas las características del proyecto para construir una estimación. Introduce dos características principales.

Las ecuaciones incluidas en este artículo, son las utilizadas para los submodelos básico e intermedio. Estas ecuaciones se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P). En la Tabla 4, se muestran las ecuaciones para esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P) así como los multiplicadores de esfuerzo (ME), utilizados solo en la ecuación de esfuerzo del submodelo intermedio.

Ecuación	Submodelo básico	Submodelo intermedio
Esfuerzo (E)	$(E) = a * (KLDC)b$	$(E) = a * (KLDC)b * ME$
Tiempo (T)	$(T) = c * (E)d$	$(T) = c * (E)d$
Personal (P)	$(P) = E/T$	$(P) = E/T$

Figura 2. 18 Ecuaciones por tipo de modelo (Boehm, 1981)

Los multiplicadores de esfuerzo, utilizados en la ecuación de esfuerzo del submodelo intermedio, son quince agrupados en cuatro grandes categorías: atributos de producto, atributos de computador, atributos personales y atributos del proyecto. En la Tabla 7 se muestran los multiplicadores de esfuerzo. Cada uno de estos multiplicadores de esfuerzo, tiene una valoración que se clasifica en una escala de 6 valores desde “muy bajo”, “bajo”, “nominal”, “alto”, “muy alto” y “extraordinariamente alto”. Estos multiplicadores de esfuerzo ajustan el valor real del esfuerzo. Dos ejemplos nos ayudan a entender el objeto de los multiplicadores de esfuerzo. Primero, en caso de que no se desee ajustar el esfuerzo nominal, se debe utilizar el valor nominal (es decir, el valor “1”) para cualquier multiplicador de esfuerzo. Segundo; si, por el contrario, se desea hacer algún ajuste en particular al esfuerzo nominal por razones de, en este caso, poca experiencia en la aplicación, se podría utilizar el multiplicador AEXP en valor nominal “muy bajo” (1,29) (Boehm, 1981).

2.15. HERRAMIENTAS A UTILIZAR

2.15.1. Sistema operativo

Windows Un sistema operativo desarrollado por la empresa de software Microsoft Corporación, el cual se encuentra dotado de una interfaz gráfica de usuario basada en el prototipo de ventanas (su nombre en inglés). Una ventana representa una tarea ejecutada o en ejecución, cada una puede contener su propio menú u otros controles, y el usuario puede ampliarla o reducirla

mediante un dispositivo señalador como el ratón o esta base de ventanas fue éxito histórico porque permitió dejar en el pasado las secuencias de comando de control como las usadas en el sistema operativo DOS. **Windows es el sistema operativo más propagado a nivel mundial**, utilizado por millones de usuarios. (Jose Palacios, 2017)

Características:

- ✓ Interfaz de usuario gráfica (mayor información y más asequible).
- ✓ Multitarea (permite ejecutar varias aplicaciones al mismo tiempo).
- ✓ Posibilidad de integrar recursos multimedia (textos, imagen y sonido).
- ✓ Herramienta para el trabajo en red, transmisión de información y comunicación entre usuarios.
- ✓ Incorporación de importantes programas (accesorios o utilitarios) para diversos usos: Un Bloc de notas, un procesador de textos (Wordpad), etc.

2.15.2. Base de Datos

PostgreSQL un sistema de código abierto de administración de bases de datos del tipo relacional, aunque también es posible ejecutar consultas que sean no relaciones. En este sistema, las consultas relacionales se basan en SQL, dos detalles a destacar de PostgreSQL es que posee data types (tipos de datos) avanzados y permite ejecutar optimizaciones de rendimiento avanzadas, que son características que por lo general solo se ven en sistemas de bases de datos comerciales. (Santiago Borges, 2019)

Características:

- ✓ Es de código abierto
- ✓ Es gratuito
- ✓ Es multiplataforma
- ✓ Es fácil de usar
- ✓ Puede manejar un gran volumen de datos
- ✓ Soporte total de ACID¹⁹ (sin pérdida de datos en una transacción)

¹⁹ Atomicity, Consistency, Isolation Y Durability, Atomicidad, Consistencia, Aislamiento Y Durabilidad de las transacciones que se realizan en una base de datos.

2.15.3. IDE

Eclipse IDE es un IDE²⁰ de código abierto, es una plataforma potente con un buen editor, depurador y compilador. El JDT (Java development toolkit) es de los mejores que existen en el mercado y tiene detrás una gran comunidad de usuarios que van añadiendo mejoras al software, fue desarrollado por IBM como evolución de su VisualAge, pero ahora lo mantiene la fundación Eclipse, que es independiente y sin ánimo de lucro. (Carlos Moreno, 2015)

Características:

Existen muchas versiones de Eclipse para Java y además se pueden extender, mediante plugins. Es muy difícil enumerar sus componentes ya que crecen cada día, pero a continuación daremos un detalle de los más usados:

- ✓ **Editor inteligente:** nos permite ver las clases disponibles, los métodos y campos que soporta un objeto o clase.
- ✓ **Corrección y compilado automático:** Eclipse, compila el código sin necesidad de solicitarlo y nos muestra los errores automáticamente.
- ✓ **Soporte para depuración (debugging):** Podemos detener la ejecución sobre la línea de código que nos interesa. Podemos avanzar paso a paso, depurar diferentes hilos, etc.
- ✓ **Web:** tiene un editor para jsp, permite iniciar, parar y desplegar en diferentes servidores web y mucho más.
- ✓ **Manejador de versiones:** instalando plugins, soporta Subversion, Github y más.
- ✓ **Android:** Google ha desarrollado plugins para la mayoría de sus tecnologías Java. (Carlos Moreno, 2015)

2.15.4. IDE Android

Android Studio Android Studio es un entorno desarrollo integrado para la plataforma de **ANDROID**, con el objetivo de crear un entorno dedicado en

²⁰ Acrónimo del inglés integrated development environment (entorno de desarrollo integrado). Entorno donde el programador tiene todas las herramientas de trabajo a su disposición.

exclusiva a la programación de aplicaciones para dispositivos Android, fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O. Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0, está programado en Java y es multiplataforma. (Honig Zach, 2013)

Características:

- ✓ **Soporte** para programar aplicaciones para **Android Wear** (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- ✓ Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- ✓ Plantillas para crear diseños comunes de Android y otros componentes.
- ✓ Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario
- ✓ Soporte para programar aplicaciones para Android Wear.
- ✓ Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.
- ✓ Renderizado en tiempo real.
- ✓ Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso. (Honig Zach, 2013)

2.15.5. Lenguaje de Programación

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. (James Gosling, 1995)

Características:

- ✓ Orientado a objetos
- ✓ Distribuido y dinámico

- ✓ Robusto
- ✓ Seguro
- ✓ Multitarea
- ✓ Portable

Java tiene la característica de ser al mismo tiempo compilado e interpretado. El compilador es el encargado de convertir el código fuente de un programa en un código intermedio llamado bytecode que es independiente de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java. (James Gosling, 1995)

2.15.6. Mapas

Leaflet es una librería de código abierto JavaScript que permite crear mapas interactivos para la web y para entorno móvil de manera fácil. Pesa apenas unos 33KB, con esta biblioteca podemos usar mapas base de Google, ESRI, OpenStreetMap, entre otros, agregar marcadores, polígonos, líneas, realizar zoom y extender su funcionalidad con una buena cantidad de plugins, entre ellos, muchos desarrollados por ESRI.

(Aurelio Morales, 2012)

Características:

- ✓ Sencillo y rápido de aprender
- ✓ Facilidad de uso
- ✓ Características básicas pero que funcionan a la perfección
- ✓ Soporte móvil
- ✓ HTML 5 y CSS3
- ✓ Funciona tanto en los modernos como en los viejos navegadores web
- ✓ Ampliable con plugins
- ✓ API bien documentado

2.15.7. Librerías de desarrollo

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 (QR) en

J2SE, J2EE entre otras. Barcode Maker permite crear una gran cantidad de códigos de barras entre rangos especificados. Básicamente lee el conteo inicial, crea el código de barras y lo guarda en la ubicación especificada, además de ser una biblioteca de código abierto. (Catalin Burcea, 2020)

ZXing es una biblioteca de procesamiento de imágenes de código de barras 1D / 2D multiformato de código abierto implementada en Java, con puertos a otros idiomas. Esta es la biblioteca principal que admite códigos QR en Java.

La biblioteca QRGen ofrece una API simple de generación de códigos QR construida sobre ZXing. Proporciona módulos separados para Java y Android, es un proyecto open-source libre de derechos de autor para uso y modificación, ofrece soporte para lectura y decodificación para códigos BIDI o QR en múltiples plataformas. (Catalin Burcea, 2020)



CAPÍTULO III

MARCO APLICATIVO

3. MARCO APLICATIVO

3.1. INTRODUCCIÓN

El modelo de realidad aumentada para el 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 “trameaje”. En este capítulo se detalla los elementos y metodologías que se plantearon de forma teórica definiendo los aspectos esenciales del modelo de realidad aumentada para el control de tramos, seguidamente implementar el modelo mencionado con los requisitos funcionales conseguidos, realizando al final las pruebas correspondientes.

Para cumplir a cabalidad el desarrollo del modelo de realidad aumentada para el control de tramos, es necesario estudiar la estructura del modelo el cual se propone, ya que se tendrá un mejor entendimiento y definición de la misma, esto con el fin de cumplir con el objetivo que se menciona en el capítulo uno.

3.2. ESTRUCTURA DEL MODELO

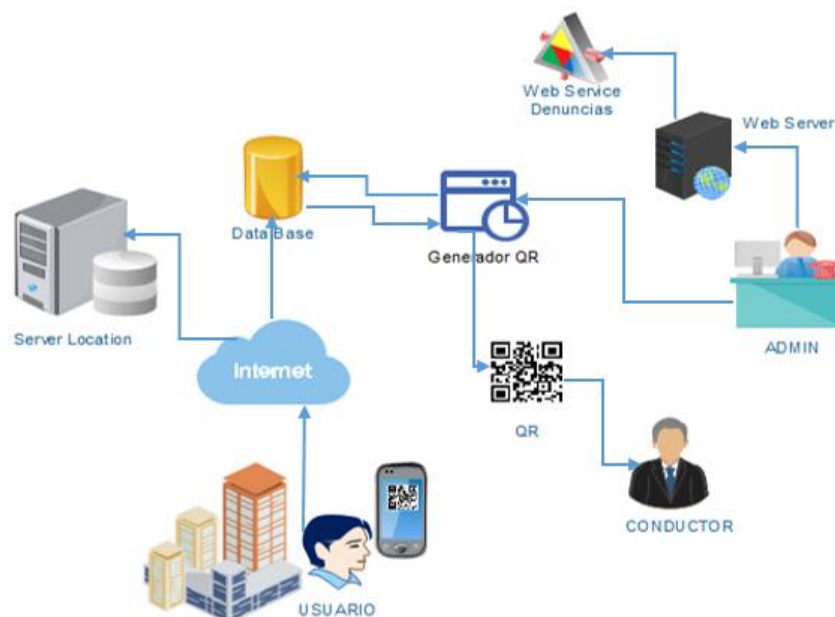


Figura 3. 1 Estructura del Modelo General (Elaboración propia)

3.3. METODOLOGÍA MOBILE-D

Para el desarrollo de la aplicación móvil se recurre a la presente metodología, que básicamente se enfoca a efectuar aplicaciones en equipos pequeños con ciclos de desarrollo muy rápidos.

3.3.1. Fase 1: Exploración

En esta actividad se definen a los involucrados del proyecto y se identificó sus tareas, roles y responsabilidades:

Líder de Proyecto: 1 Analista programador

Usuarios de la aplicación: Ciudadano de la Ciudad del El Alto

Sponsor: Sindicato “Simón Bolívar”

En reunión con todos ellos se definió la propuesta de producto, el cual es el desarrollo de la App en plataforma Android, RUApp

3.3.1.1. Establecimiento de actores. – Se identifica a los actores que intervienen:

Tabla 3. 1
Identificación de Actores

Usuario	Descripción
Usuario	Persona que tendrá instalado la aplicación móvil en su dispositivo móvil.
Administrador	Persona que administrará la base de datos, genera códigos QR.

Nota: Elaboración Propia

3.3.1.2. Definición del alcance. - En esta actividad se determina los requisitos previos, así como los objetivos y el alcance del producto en base al tiempo de duración del proyecto.

a) Requisitos previos:

- ✓ Información del registro de vehículos del sindicato.
- ✓ 1 Smartphone con Sistema operativo Android en versión 3.0 o superior.

b) Objetivos:

- ✓ Tomar foto al código QR previamente registrado y recabado mediante el sindicato.
- ✓ Con la obtención de la información mediante el QR, comprobar la ruta que tiene asignada la movilidad.
- ✓ Consultar en la base de datos intermedia información relevante de la movilidad (situación adicional y/o opcional).
- ✓ Reportar el incumplimiento del tramo asignado si fuese el caso, además de alguna otra irregularidad.

c) Alcance:

- ✓ Prototipo funcional de una App Android en el cual se pueda controlar el cumplimiento de la ruta asignada por el sindicato “Simón Bolívar”

3.3.1.3. Establecimiento de proyectos. – En esta etapa se define el entorno técnico y físico del proyecto.

- **Documento de requerimientos iniciales:** Documento de Análisis Inicial y Diseño de Arquitectura base.
- **Tecnología:** Android
- **Lenguaje de Programación:** Java
- **Librerías Java :** jdk 1.8.0, Java Barcode Generation
- **IDE :** Android Studio - Eclipse for Java Developers
- **Sistema Operativo:** Android versión 3.0 superior
- **Equipos:** 1 Gestor de Base de Datos PostgreSQL, 1 laptop, espacio mínimo disponible en Disco de 10 GB.
- **Metodología de Desarrollo:** Mobile-D

3.3.2. Fase 2: Inicialización

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.

3.3.2.1. Configuración del proyecto

a. Preparación del ambiente:

Instalación del jdk 1.8.0, instalación de Android Studio 4, instalación del Eclipse, instalación del Gestor de Base de Datos PostgreSQL.

b. Capacitaciones:

Capacitación técnica sobre la tecnología de desarrollo móvil con Android Studio.

c. Plan de comunicación:

Se solicito la lista de correos y teléfonos como medio de comunicación con los administradores del sindicato “Simón Bolívar”

3.3.2.2. Planeamiento Inicial (Dia de planificación repetición 0)

Exposición del plan del proyecto y la arquitectura de la aplicación.

La arquitectura del proyecto consta de 3 partes: Aplicación Móvil, Servicio Web Geolocalización

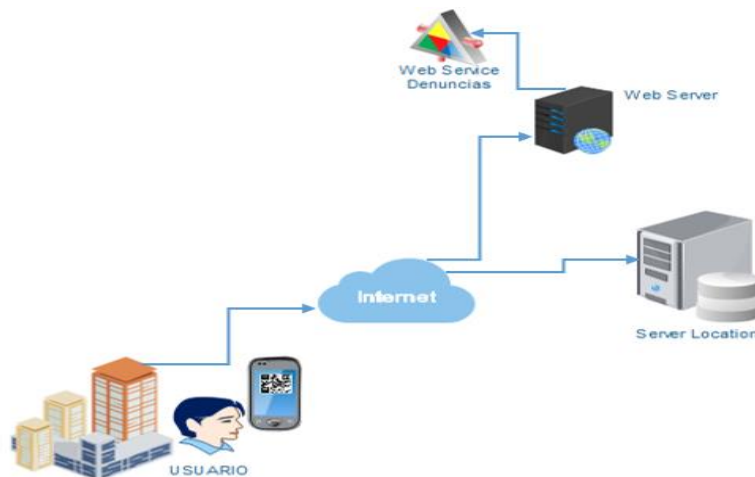


Tabla 3. 2 Estructura de la aplicación (Elaboración propia)

En la figura 3.2, se presenta la arquitectura de la aplicación RUApp (apk), la cual se instalará en un smartphone con sistema operativo Android versión 3.0 o superior, el teléfono debe contar con el acceso a internet puesto que la aplicación una vez escaneada el código QR redireccionara a un mapa en la nube donde

estará apuntando la ubicación actual del usuario y marcará la ruta de la movilidad, de disponibilidad 24x7.

3.3.2.3. Explicación del producto (Día de trabajo en repetición 0)

Requerimientos de usuario:

Tabla 3. 3

Requerimientos definidos –Tomar foto al código QR

Identificador	F01	Nombre	Tomar una foto al código QR
Tipo	Funcional	Prioridad	Alta
Necesidad	Si	Verificable	Si
Descripción	El usuario desde la aplicación debe poder enfocar al código QR para georreferenciar la ruta del autotransporte, el mismo debe visualizarse en el contexto de la aplicación RUApp.		

Nota. Elaboración Propia

Tabla 3. 4

Requerimientos definidos – Georreferenciar Ruta

Identificador	F02	Nombre	Georreferenciar la ruta desde la posición 0
Tipo	Funcional	Prioridad	Alta
Necesidad	Si	Verificable	Si
Descripción	El usuario luego de capturar el código QR debe poder visualizar el mapa en tiempo real donde estará trazado la ruta del autotransporte esto con el punto de partida del usuario (posición 0).		

Nota. Elaboración Propia

Tabla 3. 5*Requerimientos definidos – Consultar información reservado*

Identificador	F03	Nombre	Consultar en la BD la información necesaria de la movilidad
Tipo	Funcional	Prioridad	Media
Necesidad	Si	Verificable	Si
Descripción	El usuario además de verificar en tiempo real de la ruta, también debe poder visualizar datos concretos del autotransporte.		

Nota. Elaboración Propia**Tabla 3. 6***Requerimientos definidos – Denunciar el incumplimiento de la ruta*

Identificador	F04	Nombre	Reportar el incumplimiento de la ruta asignada
Tipo	Funcional	Prioridad	Alta
Necesidad	Si	Verificable	si
Descripción	La aplicación RUApp debe mostrar la opción “Denunciar vehículo”, cada vez que este no cumpla con una ruta		

Nota. Elaboración propia

3.3.2.4. Caso de Uso

En este punto se presentan los casos de uso de la aplicación de escritorio representada en eventos, además se realiza una descripción de cada uno de los casos de usos, este proceso denominado descripción de alto nivel, las mismas contienen el nombre del caso de uso, actor, tipo, descripción con el fin de tener una mejor percepción del caso de estudio.

- **Caso de uso: Usuario**

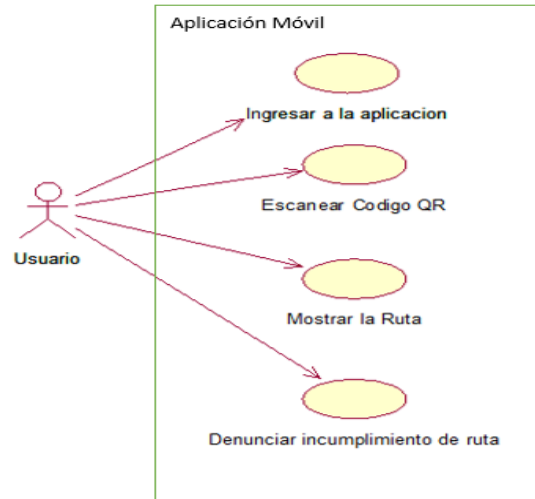


Figura 3. 2 Caso de uso Usuario (Elaboración propia)

Tabla 3. 7

Descripción de alto nivel – ingresar a la aplicación

Caso de uso: Ingresar a la aplicación	
Actor	Usuario
Tipo	Primario
Descripción	El usuario debe instalar el apk de la aplicación en el dispositivo móvil, luego buscar el icono de RUApp en el móvil e iniciar con la aplicación.

Nota Elaboración Propia

Tabla 3. 8

Descripción de alto nivel – Escanear Código QR

Caso de uso: Escanear código QR	
Actor	Usuario
Tipo	Primario
Descripción	El usuario selecciona la opción <u>Escanear Código QR</u> y accede mediante la cámara del dispositivo móvil, captura el código QR mostrando el mapa donde estará trazado la ruta del transporte público.

Nota. Elaboración Propia

Tabla 3. 9

Descripción de alto nivel – Mostrar Ruta

Caso de uso: Mostrar la ruta	
Actor	Usuario
Tipo	Primario
Descripción	El usuario observa el mapa verificando que no se presente alguna irregularidad con la ruta en seguimiento, el mismo se verifica en tiempo real.

Nota. Elaboración Propia

Tabla 3. 10

Descripción de alto nivel – Denunciar el Incumplimiento de la Ruta

Caso de uso: Denunciar incumplimiento de la ruta	
Actor	Usuario
Tipo	primario
Descripción	El usuario puede emitir la denuncia oportuna en caso del incumplimiento de la ruta asignada al autotransporte, donde además se captura una copia del mapa, con el fin de tener el respaldo correspondiente, debe llenar dos datos (nombre y carnet) y enviar la denuncia.

Nota. Elaboración Propia

- **Caso de uso: Administrador**

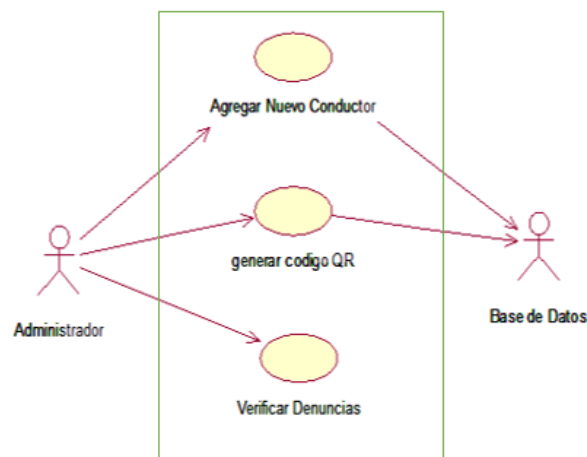


Figura 3. 3 Caso de uso - Administrador (Elaboración propia)

En el caso de uso del administrador se realiza una descripción de cada uno de los casos de usos, este proceso denominado descripción de alto nivel, descrito en las siguientes tablas:

Tabla 3. 11

Descripción de alto nivel - Agregar Nuevo Conductor

Caso de uso: Agregar Nuevo Conductor	
Actor	Administrador
Tipo	Primario
Descripción	El administrador debe registrar a nuevos conductores que no figuren en la base de datos, con el fin de proveer del código QR.

*Nota.*Elaboración propia

Tabla 3. 12

Descripción de alto nivel-Generar QR

Caso de uso: Generar Código QR	
Actor	Administrador
Tipo	Primario
Descripción	El administrador debe generar e imprimir el código QR para que cada conductor situé la imagen dentro de la movilidad.

Nota. Elaboración propia

Tabla 3. 13

Descripción de alto nivel - verificar denuncias

Caso de uso: Verificar Denuncias	
Actor	Administrador
Tipo	Primario
Descripción	El administrador tendrá el acceso a un portal WEB donde se verificará las denuncias por incumplimiento de la ruta asignada, para su posterior sanción.

Nota. Elaboración propia

3.3.2.5. Elaboración de Prototipos de mediana fidelidad (Dia de publicación en repetición 0)

Las pantallas planteadas buscan cumplir con los requerimientos de la aplicación móvil.

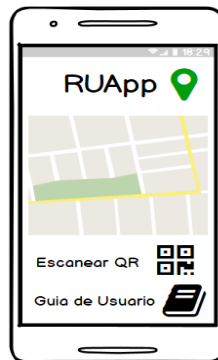


Figura 3. 4 Prototipo Inicial - Pantalla de Inicio (Elaboración propia)

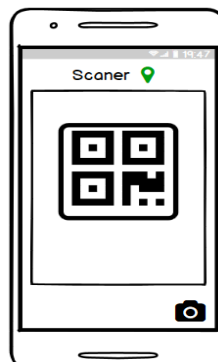


Figura 3. 5 Prototipo Inicial - Escaneado de Código QR (Elaboración propia)

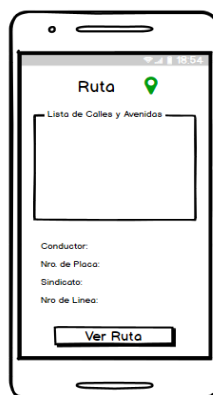


Figura 3. 6 Prototipo Inicial – Lista de Información (Elaboración propia)

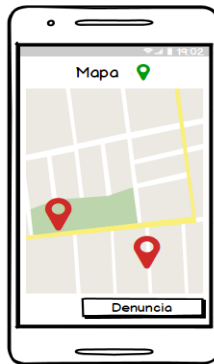


Figura 3. 7 Prototipo Inicial – Visualización del mapa y el trazado de la Ruta (Elaboración Propia)

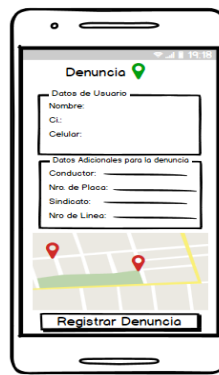


Figura 3. 8 Prototipo Inicial – Registro de la Denuncia (Elaboracion propia)

3.3.3. Fase 3: Producción

3.3.3.1. Modelo de Entidad Relación

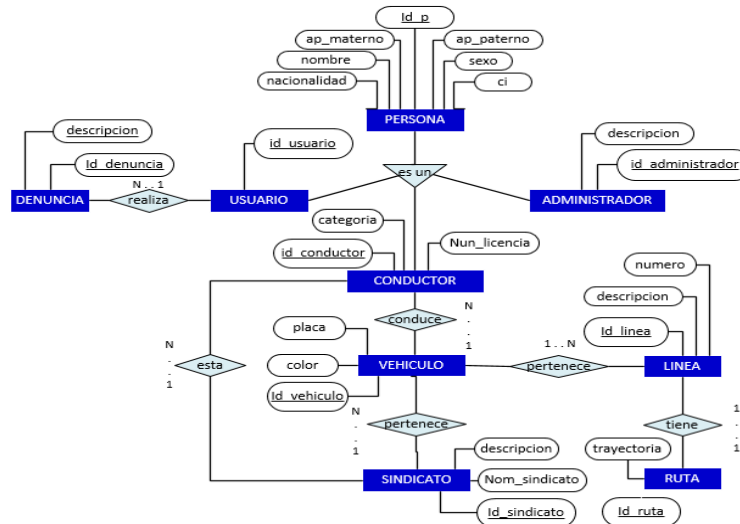


Figura 3. 9 Modelo Entidad Relación (Elaboración propia)

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

3.3.3.2. Diagrama de Clase

En este punto se muestra el diagrama modelo de base de datos.

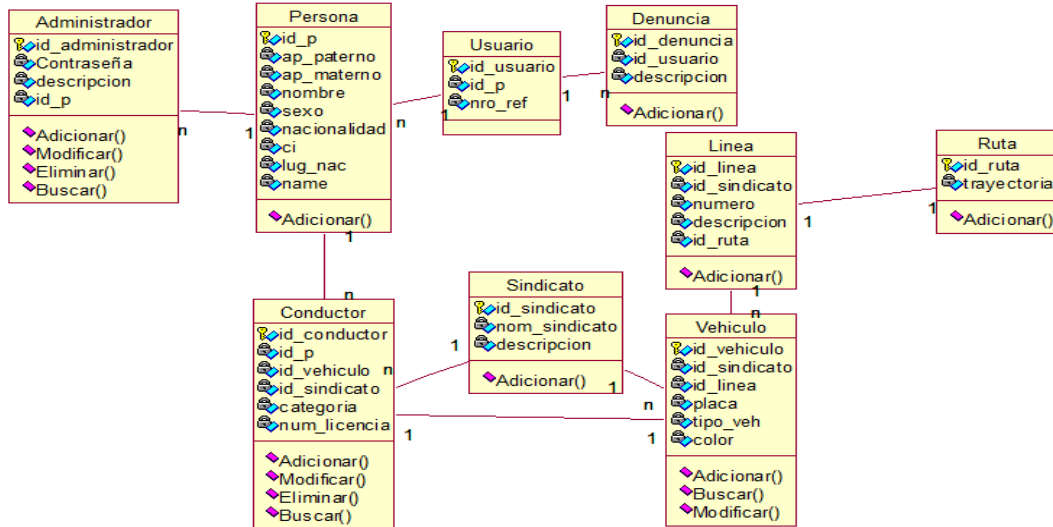


Figura 3. 10 Diagrama de Clase SGQR (Elaboración propia)

3.3.3.3. 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, estas tareas se muestran a continuación en la tabla:


Tabla 3. 14
Plan de Iteraciones del Desarrollo

Iteraciones	Cod.	Tareas
Primera iteración	P1	Diseño de Interfaz – Sistema Generador QR
	P2	Diseño e implementación de la base de datos
	P3	Creación de código QR – Integrar Barcode.jar
Segunda iteración	S1	Diseño de Interfaz de la App “RUApp”
	S2	Reconocimiento de QR - Integrar ZXing.
	S3	Pantalla información obtenida del código QR

Tercera iteración	T1	Diseño de interfaz, integrar y obtener el Mapa en tiempo real con Leaflet.
Cuarta iteración	C1	Unificación de GPS
Quinta iteración	Q1	Diseño de interfaz pantalla Denuncia

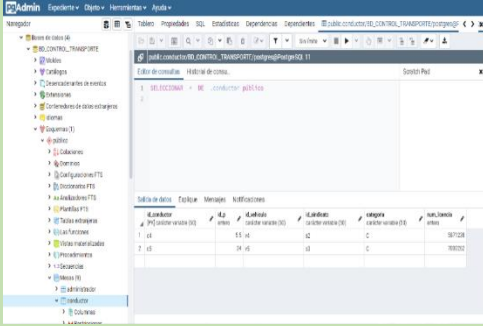
Nota. Elaboración Propia

Tabla 3. 15
Diseño de Interfaz – primera iteración

Diseño de Interfaz – Sistema		Cod: P1
Nombre:	Generador QR	
Planificación:	En esta iteración se desarrolló la interfaz para la generación de código QR.	
Trabajo	En esta interfaz desarrollada para el administrador, el cual contiene las opciones de altas, bajas, modificaciones, búsqueda y registro del administrador.	
Liberación		

Nota. Elaboración Propia

Tabla 3. 16
Implementación de la base de datos

Diseño e implementación de la base de datos		Cod: P2
Nombre:		
Planificación:	En la iteración seguida se procede a la implementación de la base de datos, el cual fue desarrollada en el gestor de BD PostgreSQL de código abierto.	
Trabajo	Se crea la BD, las tablas que interactúan con la misma y se adicionan los atributos de las respectivas entidades referentes al transporte público.	
Liberación		

Nota. Elaboración Propia

Tabla 3. 17
Creación de código QR

Creación de código QR - Integrar Barcode.jar		Cod: P3
Nombre:		
Planificación:	<p>En el código QR se almacenará la información de los atributos más importantes de las entidades como ser:</p> <p>Conductor: nombre, apellidos, sexo, nacionalidad, categoría de licencia.</p> <p>Vehículo: placa y color.</p> <p>Sindicato: nombre de sindicato, línea</p>	

Trabajo	En esta iteración también será necesario la librería Barcode.jar de Java para poder crear el código QR, el cual se llegó a adicionar en la codificación que se realizó en el IDE Eclipse.
Liberación	<p>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.</p> <p>Condiciones de ejecución: Para la operación la condición inicial es que se cuente con la base de datos debidamente adicionada los datos del conductor para la búsqueda y generación del código QR.</p> <p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Ingresar 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> <p>El sistema guardará un símbolo QR en formato jpg.</p>
Código	<p>✓ Agregar variables con el fin de establecer los parámetros con los cuales estaremos creando las imágenes que contendrán los códigos QR.</p> <p><i>int udm=0, resol=72, rot=0; float mi=0.000f, md=0.000f, ms=0.000f, min=0.000f, tam=5.00f;</i></p> <p>La variable udm contendrá la unidad de medida que utilizaremos, 0 para pixeles, 1 para centímetros y 2 para pulgadas.</p> <p>resol contiene la resolución de la imagen en pixeles.</p> <p>rot indica la cantidad en grados a rotar la imagen.</p> <p>Las variables mi, md, ms y min contienen el tamaño del</p>

margen blanco que rodeará al código QR, en este caso 0. **tam** indica el tamaño de la imagen.

- ✓ Una vez que hemos creado y dado un valor a los parámetros agregaremos dentro del método `geneQR` una nueva instancia de la clase `QRCode` e la siguiente manera:

```
QRCode c = new QRCode();
```

Al realizar lo anterior debemos importar la librería **`com.barcode.lib.barcode.QRCode`**

- ✓ Posteriormente agregaremos en forma de cadena la información que deseamos que se incluya dentro del código QR por medio del método `setData()`

```
c.setData(dato);
```

- ✓ Ahora indicamos el tipo de información que vamos a manejar con el método `setDataMode()`

```
c.setDataMode(QRCode.MODE_BYTE);
```

MODE_BYTE (para bytes)

- ✓ Posteriormente indicamos al objeto los parámetros con los cuales vamos a trabajar.

```
c.setUOM(udm);
```

```
c.setLeftMargin(mi);
```

```
c.setRightMargin(md);
```

```
c.setTopMargin(ms);
```

```
c.setBottomMargin(min);
```

```
c.setResolution(resol);
```

```
c.setRotate(rot);
```

```
c.setModuleSize(tam);
```

- ✓ Después generamos el archivo de imagen en el disco duro con la siguiente instrucción:

```
String archivo=
```


```
System.getProperty("user.home")+"/qrdibujo.jpg";
```

	<p><i>c.renderBarcode(archivo);</i></p> <p>El método <i>renderBarcode()</i> nos permite generar el código en una imagen en este caso dentro del directorio del usuario actual y bajo el nombre de <i>qrdibujo.jpg</i>.</p> <p>✓ Por último, utilizamos la clase Desktop para ver la imagen con el visor de imágenes predeterminado del sistema.</p> <p><i>Desktop d=Desktop.getDesktop();</i> <i>d.open(new File(archivo));</i></p>
--	---

Nota. Elaboración Propia

Tabla 3. 18

Diseño de Interfaz de la App “RUApp” - Segunda iteración

Diseño de Interfaz de la App		Cod: S1
Nombre:	“RUApp”	
Planificación:	En la iteración presente se diseñó y desarrolló la interfaz de inicio de la aplicación móvil.	
Trabajo	El desarrollo de la interfaz tiene dos botones Escanear QR y Guía de Usuario , El primer botón inicia el escaneo de código QR y el segundo describe brevemente el uso de la aplicación.	
Liberación		

Nota. Elaboración Propia

Tabla 3. 19
Reconocimiento de QR - Integrar ZXing

Nombre:	Reconocimiento de QR - Integrar ZXing Cod: S2
Planificación:	La aplicación escaneará un código QR, que podrá ser accedido por medio de la cámara del dispositivo móvil en uso, con el fin de enfocar un código de naturaleza QR y extraer la información que este tiene almacenado.
Trabajo	Para la codificación se integró a nuestro prototipo para el dispositivo móvil la librería <i>ZXing</i> mediante el cual se realiza el procesamiento y reconocimiento del código QR.
Liberación	<p>Descripción: El sistema espera el reconocimiento de un código QR para mostrar el mensaje almacenado en el símbolo.</p> <p>Condiciones de Ejecución: La condición inicial es que el usuario cuente con un dispositivo móvil que tenga cámara, el cual podrá reconocer el código QR.</p> <p>Además, se deberá tener al alcance de la vista el código QR impreso y pegado en el vehículo para su escaneo.</p> <p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Ingresa al sistema (usuario) 2. Presiona la opción “Escanear QR” 3. Se activa la cámara del Dispositivo Móvil. 4. Escanea el código QR <p>Resultado: El sistema muestra el mensaje contenido en el Código QR, siendo exitoso el reconocimiento del mismo.</p>
Código	<p>✓ Instalación de la dependencia: <i>implementation 'com.journeyapps:zxing-android-embedded:3.4.0'</i></p> <p>Librería <i>import com.google.zxing.integration.android.IntentIntegrator;</i> <i>import com.google.zxing.integration.android.IntentResult;</i></p>

```


Iniciar captura y Personalización
public void scanCode(){
IntentIntegrator integrator = new IntentIntegrator(this);
//inicia la captura
integrator.setCaptureActivity(CaptureAct.class);
integrator.setOrientationLocked(false);
integrator.setDesiredBarcodeFormats(IntentIntegrator
.ALL_CODE_TYPES);
integrator.setPrompt("ESCANEANDO...!");
integrator.initiateScan();}

```

Nota. Elaboración Propia

Tabla 3. 20


Pantalla información obtenida del código QR

Nombre:	Pantalla información obtenida del código QR	Cod: S3
Planificación:	Continuando en la segunda iteración se diseñó y desarrollo la interfaz donde se mostrará la información obtenida del Código QR, así como los nombres de calles y avenidas por donde circula el vehículo de transporte público.	
Trabajo	Esta interfaz es un <i>Actividad</i> en programación Android, la cual contiene un botón con la opción que permitirá observar un mapa con ubicación y la ruta de la línea vehicular.	
Liberación		

Nota. Elaboración Propia

Tabla 3. 21

Diseño de interfaz y Mapa con Leaflet - Tercera iteración

Diseño de interfaz, integrar y obtener el Mapa en tiempo real con Leaflet.		Cod: T1
Nombre:		
Planificación:	Se desarrolla el diseño de la interfaz donde se mostrará el mapa con la ruta trazada acorde al número de línea de transporte público, siendo necesario la biblioteca JavaScript Leaflet de código abierto para mapas interactivos adaptables para móviles.	
Trabajo	Para añadir el mapa se utilizaron las librerías de Leaflet, el código llega a tener una cantidad menor que usualmente tendría con otra herramienta de pago, además se realizó la codificación de la ruta por los calles y avenidas exactas por donde recorre el vehículo del número de línea de transporte público y se procedió a integrar en el mapa sin necesidad de recurrir a distintas opciones (ver anexo).	
Liberación		

Nota. Elaboración Propia

Tabla 3. 22

Unificación de GPS - Cuarta iteración

Unificación de GPS		Cod: C1
Nombre:		
Planificación:	Para poder controlar la ruta que sigue el vehículo 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 la línea en estudio.	

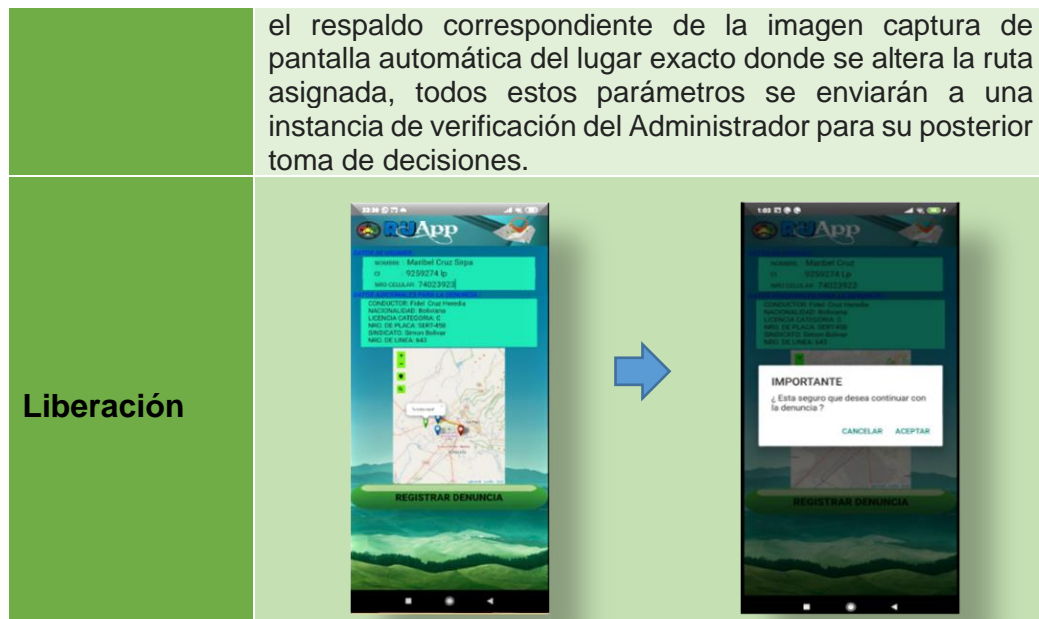
Trabajo	Se realiza el código Leaflet mediante una serie de métodos que llegaron a completar adecuadamente los parámetros necesarios para la aplicación móvil. (ver anexo)
Liberación	<p>Descripción: El sistema requiere que se active el GPS para que se muestre el punto exacto en la ruta del transporte público en estudio.</p> <p>Condiciones de Ejecución: las condiciones para esta prueba es que el usuario cuente con acceso a internet y cuente con un receptor GPS que en este caso será un dispositivo móvil.</p> <p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Ingreso al sistema RUApp (usuario) 2. Se activa GPS 3. Muestra la ubicación exacta del usuario <p>Resultado Esperado: El sistema muestra la posición inicial del usuario en la ruta que este trazado en el mapa, la geolocalización se realiza con éxito.</p>
Código	<p>Mediante el método <i>map.on</i> capturamos la localización por latitud y longitud, esto en tiempo real con el GPS integrado en mismo dispositivo.</p> <pre>map.on('locationfound', e => { const coords = [e.latlng.lat, e.latlng.lng]; const marker = L.marker(coords); marker.bindPopup('Tu estas aqui!'); map.addLayer(marker); })</pre>

Nota. Elaboración Propia

Tabla 3. 23

Diseño de interfaz pantalla Denuncia - quinta iteración

Diseño de interfaz pantalla		Cod: Q1
Nombre:	Denuncia	
Planificación:	En la quinta iteración se diseñó la ventana para realizar la denuncia en caso que el vehículo no transite por la ruta que le fue asignada	
Trabajo	En la Actividad programada en Android Studio se tiene 3 campos, donde el usuario debe adicionar los datos personales Nombre, CI y Núm. Celular , además de tener	



Nota. Elaboración Propia

3.3.4. Fase 4: 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 del modelo de control.

3.3.4.1. Workshop de post Iteración

Mejoras: Se mejoró los estilos de interfaz gráfica de la aplicación móvil RUApp dándole una mejor apariencia y presentación al usuario final y al administrador respectivamente.

Fortalezas: la aplicación móvil RUApp funciona de manera offline parcialmente, para la funcionalidad total y adecuada es necesario contar con conexión a internet.

Debilidades: la aplicación móvil RUApp funciona solamente si se cuenta con una conexión a internet, el buen funcionamiento de la aplicación también depende de la buena estabilidad de internet.

3.3.5. Fase 5: Pruebas. -

Esta fase tiene como objetivo verificar el funcionamiento de los prototipos, además de apreciar la disponibilidad de una versión estable de los dos mismos que sean plenamente funcionales, en base a las iteraciones realizadas de manera respectiva se hallaron algunos aspectos que fueron mejorados gradualmente mediante las iteraciones realizadas posteriormente, las siguientes tablas muestran algunos detalles de la prueba.

3.3.5.1. Prueba de aceptación para el sistema Gestor de Códigos QR

Tabla 3. 24

Prueba de Aceptación Sistema Gestor de QR

PRUEBA DE ACEPTACIÓN
Caso de prueba: Funcionalidad Sistema portable
Descripción: El sistema requiere mejora en uno de los campos cuando se ingresa el password debe de tener reserva y poder visualizar por un hecho del usuario, se necesita verificar los datos actualizados de la opción modificar afiliado, esto con la finalidad de fácil usabilidad del administrador.
Condiciones de Ejecución: las condiciones para la prueba es que se cuente con acceso a un equipo computador para la instalación del portable, no requiere de recursos de última generación.
Procedimiento: <ul style="list-style-type: none">✓ Se Ingresó al sistema✓ Se ingresa a la ventana principal✓ Se ingresa a la los datos de Loguin✓ Se ingresa al panel de control para el ingreso de nuevos usuarios tanto como afiliados como administradores.

Resultado Esperado:

El sistema muestra los cambios efectuados en la visualización de interfaz de usuario u administrador y además se llegaron a aplicar cambios de manera que beneficie a su fácil usabilidad para el administrador que forma parte del Sindicato “Simón Bolívar”

Nota. Elaboración Propia

3.3.5.2. Prueba de aceptación para la aplicación móvil RUApp

Tabla 3. 25

Prueba de Aceptación - funcionalidad de la aplicación móvil

PRUEBA DE ACEPTACIÓN
Caso de 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 e inductivos para el usuario.
Condiciones de Ejecución: las condiciones para la prueba es que se cuente con acceso a internet en el móvil en el cual se realizara la prueba.
Procedimiento: <ul style="list-style-type: none"> ✓ Se Ingresó al sistema ✓ Se ingresa a la ventana principal ✓ Se ingresa a la ventana donde ve se la información de línea vehicular ✓ Se ingresa a la ventana donde se muestra la ruta ✓ Se ingresa a la ventana denuncia
Resultado Esperado: <p>El sistema muestra los cambios efectuados en la visualización de interfaz de usuario los cuales se priorizaron el fácil entendimiento del mismo para su mejor uso por el usuario.</p>

Nota. Elaboración Propia

3.4. MÉTRICA DE CALIDAD QSOS

Se describe el flujo de trabajo así para evaluar los prototipos además de sus componentes, la valoración se llevará con dos tipos de usuario: Administrador (afiliado del área administrativa) y Usuario (pasajero del minibús) cada uno con el respectivo sistema a interactuar.

3.4.1. Tarjeta de identificación del software “Sistema Generador QR”

Tabla 3. 26

Tarjeta de identificación del software “Sistema Generador QR”

INFORMACIÓN GENERAL			
Nombre del Software:	Sistema Generador QR	Fecha de creación:	05/05/2020
Tipo de Software:	Aplicación de escritorio	Autor:	Maribel Cruz
Fecha de elaboración de la tarjeta:	23/06/2020		
SERVICIOS EXISTENTES			
Documentación:	SI		
Manual nivel Admin.	SI		
ASPECTOS TÉCNICOS Y FUNCIONALES			
Funcionalidades	<ul style="list-style-type: none">• Registro del afiliado• Generación de códigos QR• Búsqueda de datos de afiliado• Modificación de datos• Eliminación de información		
Tecnologías de implementación	<ul style="list-style-type: none">• Desarrollado en IDE Eclipse• Librerías Barcode.jar• Ejecución portátil		

Fuente. Elaboración Propia

3.4.2. Tarjeta de identificación del software “RUApp”

Tabla 3. 27

Tarjeta de identificación del software “RUApp”

INFORMACIÓN GENERAL			
Nombre del Software:	RUApp	Fecha de creación:	03/06/2020
Tipo de Software:	Aplicación Android	Autor:	Maribel Cruz
Fecha de elaboración de la tarjeta:	23/06/2020		
SERVICIOS EXISTENTES			
Documentación:	SI		
Manual nivel Usuario	SI		
ASPECTOS TÉCNICOS Y FUNCIONALES			
Funcionalidades	<ul style="list-style-type: none"> • Escaneo de Código QR • Visualización del Mapa • Visualización de la ruta de la línea • Denuncia en caso de alguna irregularidad 		
Tecnologías de implementación	<ul style="list-style-type: none"> • Desarrollado en Android Studio 4.0 • Librerías Leaflet código libre • Funciones de escaneo 		

Nota. Elaboración Propia

3.4.3. Evaluación de los criterios de valoración (Sistema Generador QR)

Las métricas capturadas se clasificaron según la gravedad de la métrica y luego se agrupan en tres dimensiones principales (calidad del sistema, calidad de la información y calidad del servicio).

Tabla 3. 28
Evaluación Criterios Sistema Generador QR

Dimensión	Criterios	Características	Descripciones (pregunta relacionada)
Calidad del sistema	Fiabilidad	Madurez	¿El software es nuevo en el sindicato Simón Bolívar?
	Usabilidad	Aprendizaje	¿Qué tan fácil de aprender o entender el software sin usar el manual del usuario?
		Operabilidad	¿El software es fácil de operar?
		Accesibilidad	¿Es fácil acceder a este software sin otro software o complemento de terceros?
		Interfaz de usuario estética	¿La interfaz de usuario es adecuada con su funcionalidad de software?
	Eficiencia en el desempeño	Comportamiento de tiempo	¿Es este software fácil de instalar / configurar y operar en poco tiempo?
	Funcionalidad	Integridad funcional	¿El software cumple con las expectativas y requisitos del usuario?
Adecuación funcional		¿Funciona el software adecuadamente?	
Calidad de servicio	Tangible	Documentación	¿Manual técnico y de usuario?
	Empatía	Comunicación	¿La comunidad reconoce sus problemas y ayuda a resolverlos?

Nota. Elaboración Propia

- **Valoración:** En esta encuesta, se pidió a los encuestados que evaluaran la importancia de cada característica basándose en la siguiente escala: 0 = no importante, 1 = importante, 2 = muy importante.

Tabla 3. 29
Valoración Criterios Sistema Generador QR

Dimensión	Criterios	Características	Escala 0,1 y 2			Promedio
Calidad del sistema	Fiabilidad	Madurez	2	2	2	2
	Usabilidad	Aprendizaje	2	2	2	2
		Operabilidad	2	2	2	2
		Accesibilidad	2	1	2	1.66
		Interfaz de usuario estética	2	2	2	2
	Eficiencia en el desempeño	Comportamiento de tiempo	2	2	2	2
	Funcionalidad	Integridad funcional	1	2	2	1.66
Adecuación funcional		2	2	2	2	
Calidad de servicio	Tangible	Documentación	2	2	2	2
	Empatía	Comunicación	2	2	2	2

Nota. Elaboración Propia

3.4.3.1. Análisis de datos y resultados

Los datos obtenidos se analizaron para investigar la puntuación de importancia de cada característica identificada entre las personas encuestadas en el sindicato “Simón Bolívar”.

- **Calidad del sistema**

8 elementos de características se han incluido en esta encuesta. El resultado muestra que el 20% tiene un promedio medio, por problemas particulares de cada caso en encuesta y el 80% de la funcionalidad del sistema es sustancial para su selección, ya que puntúa los valores medio más alto, lo que indica que todas las características son importantes para la selección del sistema.

- **Calidad de servicio**

Se incluyen dos características para los criterios de dimensión de calidad del servicio, en base al resultado se demuestra que el 100% de la valoración es aceptada por los afiliados (administradores) del sindicato.

3.4.4. Evaluación de los criterios de valoración (RUApp)

Tabla 3. 30

Evaluación Criterios RUApp

Dimensión	Criterios	Características	Descripciones (pregunta relacionada)
Calidad del sistema	Fiabilidad	Madurez	¿El software es nuevo en este ámbito?
	Usabilidad	Aprendizaje	¿Qué tan fácil de aprender o entender el software sin usar el manual del usuario?
		Accesibilidad	¿Es fácil acceder a este software sin otro software o complemento de terceros?
		Interfaz de usuario estética	¿La interfaz de usuario es adecuada con su funcionalidad?
	Eficiencia en el desempeño	Comportamiento de tiempo	¿Es este software fácil de instalar / configurar y operar en poco tiempo?
	Funcionalidad	Integridad funcional	¿El software cumple con las expectativas y requisitos del usuario?
Adecuación funcional		¿Funciona el software adecuadamente?	
Calidad de servicio	Tangible	Documentación	¿Manual técnico y de usuario?

Nota. Elaboración Propia

- **Valoración:** En esta encuesta, se pidió a los encuestados que evaluaran la importancia de cada característica basándose en la siguiente escala: 0 = no importante, 1 = importante, 2 = muy importante.

Tabla 3. 31
Valoración Criterios RUApp

Dimensión	Criterios	Características	Escala 0,1 y 2			Promedio
Calidad del sistema	Fiabilidad	Madurez	2	2	2	2
	Usabilidad	Aprendizaje	2	2	2	2
		Accesibilidad	2	1	2	1.66
	Eficiencia en el desempeño	Interfaz de usuario estética	2	2	2	2
			Comportamiento de tiempo	2	2	2
		Funcionalidad	Integridad funcional	2	2	2
	Adecuación funcional		2	2	2	2
Calidad de servicio	Tangible	Documentación	2	2	2	2

Nota. Elaboración Propia

3.4.4.1. Análisis de datos y resultados.

Los datos obtenidos se analizaron para investigar la puntuación de importancia de cada característica identificada entre las personas encuestadas en esta ocasión fueron los usuarios principales de la aplicación “pasajeros”.

- **Calidad del sistema**

7 elementos de características se han incluido en esta encuesta. El resultado muestra que el 5% tiene un promedio medio, por problemas accesibilidad (mínimo conocimiento en instalar una app) y el 95% de la funcionalidad del sistema es sustancial para su selección, ya que puntúa los valores medio más alto, lo que indica que todas las características fueron aprobadas por los usuarios en la revisión.

- **Calidad de servicio y otra dimensión.**

Se incluyen una característica para el criterio de dimensión de calidad del servicio, en base al resultado se demuestra que el 100% de la valorización es aceptada por los usuarios.

3.5. ANÁLISIS DE COSTO COCOMO

Para la estimación de los costos que conlleva realizar un proyecto de esta dimensión nos apoyamos en la metodología COCOMO I, el cual se desarrolla de la siguiente manera:

- **Estimación de LDC (Líneas de Código)**

$$LDC = 5367$$

- **Estimación de Miles de líneas de Código (KLDC)**

$$KLDC = LDC/1000 = 5367/1000 = 5.36$$

Las fórmulas básicas de esfuerzo, tiempo y personal que se requiere, son las siguientes:

$$E = a_b(KLDC)^{b_b} \left[\frac{\text{Personas}}{\text{mes}} \right]$$

$$D = c_b E^{d_b} [\text{meses}]$$

Donde:

E: Esfuerzo aplicado en personas por mes.

D: tiempo de desarrollo en meses cronológico.

KLDC: Número estimado de líneas de código distribuidas (en miles).

Tabla 3. 32

Coeficiente a_b y c_b y los exponentes b_b y d_b

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi – acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.30

Nota. Pressman(2005)

Los proyectos de software son semi-acoplados ya que son proyectos intermedios en tamaño y complejidad.

- **Cálculo de esfuerzo:**

$$E = 3.0 * (5.36)^{1.12} = 19.66 \text{ [personas/mes]} = 20 \text{ [personas/mes]}$$

- **Cálculo de tiempo de desarrollo:**

$$D = 2.5 * (19.66)^{0.35} = 7.09 \text{ [mes]} = 8 \text{ [meses]}$$

- **Cálculo de personal necesario:**

$$P = E/D = 19.66/7.09 \text{ [personas]} = 2.77 \text{ [personas]} = 3 \text{ [personas]}$$

- **Cálculo de Productividad:**

$$PR = LDC/E = 5367/19.66 \text{ [LDC/persona mes]} = 272.99 \text{ [LDC/persona mes]}$$

- **Costo total del Proyecto:** Pago por mes por persona = 2792 Bs

$$\text{Costo Mes} = 3 * 2792 \text{ Bs}$$

$$\text{Costo Mes} = 8.376 \text{ Bs}$$

Entonces:

$$\underline{\text{Costo total} = 8.376 * 8 = 67.008 \text{ Bs}}$$

Por tanto, se concluye que se requieren **3 desarrolladores**, en un tiempo **de 8 meses** y el costo total del proyecto será de **67.008 bolivianos**.



CAPÍTULO IV

PRUEBAS Y RESULTADOS

4. PRUEBAS Y RESULTADOS

4.1. PRUEBAS

Con el objetivo de garantizar y evaluar la implementación de los prototipos se iniciará una fase adicional de pruebas evaluando los resultados relacionadas con la ejecución y carga de los prototipos

Primero será necesario generar el código QR, para esto se tendrá acceso al sistema Generador de Códigos QR

4.1.1. Generación de Código QR

Se debe dar de alta al nuevo afiliado para poder guardar los datos requeridos para el código QR así como se muestra en la figura 4.1

SISTEMA GENERADOR QR

Alta de Nuevo Afiliado

Tipo de Registro: Usuario Administrador Usuario Afiliado

Validación: Guardar Borrar Imprime QR

Datos Usuario:

NOMBRE: Carlos CI no registrado

AP. PATERNO: Quispe CI valido

AP. MATERNO: Coronei

CI: 7030262 EXP: LA PAZ

SEXO: MASCULINO

NACIONALIDAD: Boliviana

LIC. CATEGORIA: C

CONTRASEÑA:

DESCRIPCION:

Datos Sindicato:

NOMBRE SIND: Simon Bolivar

NRO. DE LINEA: 804

Datos Vehiculo:

TIPO DE VEHICULO: MINIBUS

PLACA: 4023-SDF

COLOR: AZUL

Figura 4. 1 Prueba adición de afiliado (Elaboración propia)

SISTEMA GENERADOR QR

Alta de Nuevo Afiliado

Tipo de Registro: Usuario Administrador Usuario Afiliado

Validación: Guardar Borrar Imprime QR

Datos Usuario:

NOMBRE: Carlos CI no registrado

AP. PATERNO: Quispe CI valido

AP. MATERNO: Coronei

CI: 7030262 EXP: LA PAZ

SEXO: MASCULINO

NACIONALIDAD: Boliviana

LIC. CATEGORIA: C

CONTRASEÑA:

DESCRIPCION:

Datos Sindicato:

NOMBRE SIND: Simon Bolivar

NRO. DE LINEA: 804

Datos Vehiculo:

TIPO DE VEHICULO: MINIBUS

PLACA: 4023-SDF

COLOR: AZUL

Figura 4. 2 Prueba Guardado de afiliado (Elaboración propia)

Una vez guardado los datos del afiliado realizamos la prueba de generación de código QR con la opción *imprimir QR*.



Figura 4. 3 Prueba Generación de QR (Elaboración propia)

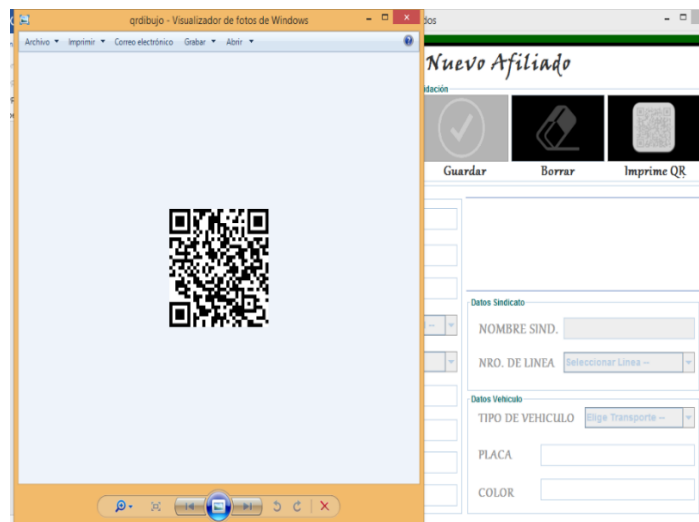


Figura 4. 4 Prueba Impresión de QR (Elaboración propia)

Esta opción nos permitió codificar la información obteniéndose como resultado una imagen en formato .jpg el cual se imprimió en papel adhesivo de 120 gr, el símbolo QR es impreso en un tamaño de 9x9 cm el cual tiene una distancia de escaneo máxima de 110 cm y mínima de 25 cm.

4.1.2. Implementación de Código QR en el minibús de la línea 643

Estos códigos QR fueron pegados en las ventanas laterales del minibús del Sindicato “Simón Bolívar” línea 643 como se refleja en la figura 4.5 la ubicación de los códigos QR en las ventanas tiene la finalidad que los usuarios puedan tener un acceso fácil y rápido al mismo.



Figura 4. 5 Minibús Línea 643 (Fotografía propia)

Posteriormente teniendo ya instalada la aplicación RUApp en el equipo móvil, se procede con iniciarlo y nos muestra dos botones “ESCANEAR QR” y “GUIA DE USUARIO”, con la primera opción se procedió a escanear el código QR como se muestra en la figura 4.6

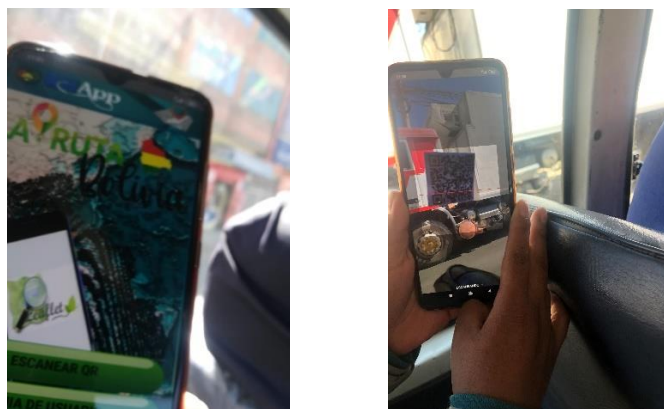


Figura 4. 6 Escaneado Código QR (Elaboración propia)

Este procedimiento muestra la siguiente el resultado de una ventana que muestra información que se tuvo del Código QR estos datos son acerca del conductor, vehículo y sindicato. Adicionalmente existe la información de nombres de calles y avenidas principales entre otras, la misma se obtiene automáticamente.

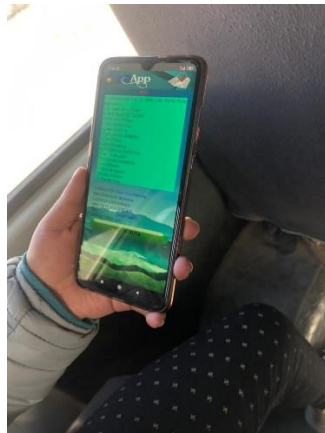


Figura 4. 7 Información de la línea Vehicular (Elaboración propia)

También se tiene el botón de “VER MI RUTA” el cual nos lleva a una ventana que tiene el mapa, la localización actual y la ruta trazada de la línea de transporte publico 643

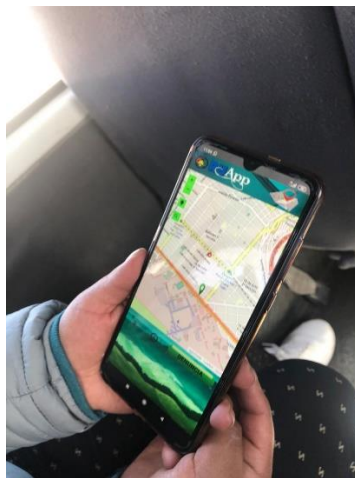


Figura 4. 8 Ruta de la línea Vehicular (Elaboración propia)

Como se puede ver el resultado de la anterior acción nos lleva a la ventana donde verificaremos el cumplimiento de la ruta, pero además tenemos un botón “DENUNCIA” si en caso fuera el conductor desvía la ruta y/o realizara el “trameaje” este botón tiene por finalidad enviar la denuncia a un portal web donde se recepcionará las denuncias, el mismo estará dispuesto por el encargado de la administración de turno, cabe aclarar que para la denuncia debemos llenar ciertos datos del usuario para hacer propia la misma con el respaldo correspondiente.



Figura 4. 9 Ventana de Denuncia (Elaboración propia)

Una vez enviado la información para la denuncia correspondiente se mostrarán en un portal web en el cual solo tendrá acceso el administrador para posteriormente dar parte a las autoridades correspondientes dentro del Sindicato “Simón Bolívar”.

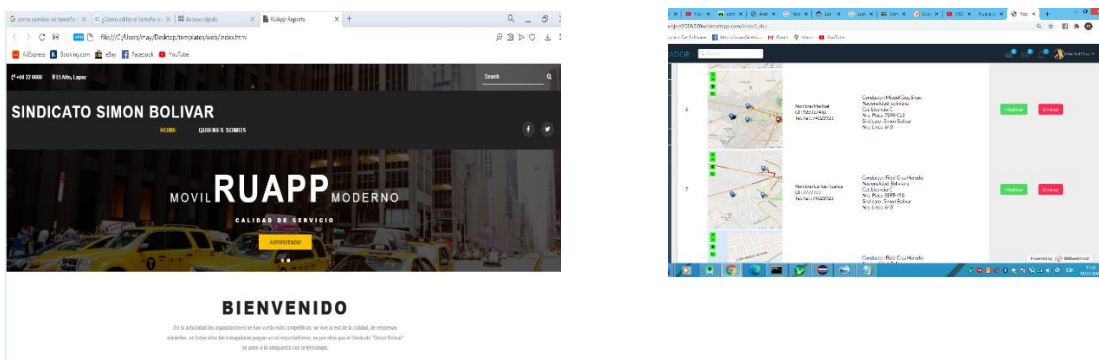


Figura 4. 10 Portal Web Denuncias (Elaboración propia)

4.2. RESULTADOS

Después de realizar las diferentes pruebas de aceptación en las distintas iteraciones y fases, se determina que la parte más importante en el Modelo de Realidad Aumentada para el Control de tramos es la aplicación móvil RUApp mediante el cual se podrá realizar el control del vehículo de forma eficiente es la razón que los resultados obtenidos se enfocaron en el aspecto mencionado.

4.2.1. Resultados Prueba de escaneo código QR

Para los resultados de la primera prueba nos basamos en el reconocimiento de los Códigos QR y la respuesta que tiene la aplicación móvil, activando la cámara de mismo dispositivo para el escaneo, así comprobando la funcionalidad de la interacción entre la aplicación y dispositivo. En esta prueba se tomó en cuenta el tamaño y la distancia de escaneo del código QR el cual se detalla a continuación:

Tabla 4. 1

Dimensión y distancia de escaneo QR

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

Nota. Elaboración Propia

Tabla 4. 2

Resultados de la primera prueba

PRUEBAS		
PRUEBAS	CANTIDAD	PORCENTAJE
Aceptadas	9	90%
Rechazadas	1	10%
Total pruebas	10	100%
Resultado	Con una valoración del 90% de aceptación, se denota efectivo	

Nota. Elaboración Propia

4.2.2. Resultados Prueba visualización de del mapa, geolocalización y trasado de ruta.

Se realizaron las pruebas en distintos equipos móviles con el fin de verificar el cargado del mapa, el trasado de la ruta y la ubicación en tiempo real esto recurriendo a la geolocalización GPS, así autenticando el funcionamiento del módulo en prueba.

Tabla 4. 3
Resultados de la segunda prueba

PRUEBAS		
PRUEBAS	CANTIDAD	PORCENTAJE
Aceptadas	15	93.75%
Rechazadas	1	6.25%
Total pruebas	16	100%
Resultado	Con una valoración del 93.75% de aceptación, se denota efectivo	

Nota.. Elaboración Propia

4.2.3. Resultados prueba funcionamiento general de la aplicación móvil RUApp

En esta última iteración de pruebas nos enfocamos en el funcionamiento total de la aplicación RUApp, basándonos en todos los módulos cargados en las cuales tendrá el acceso el usuario, verificando que el resultado sea optimo y eficaz para su fácil entendimiento y manipulación, las pruebas fueron numerosas, de las cuales se obtuvieron buenos resultados, aquí en la presente tabla de pruebas.

Tabla 4. 4
Resultados de la tercera prueba

PRUEBAS		
PRUEBAS	CANTIDAD	PORCENTAJE
Aceptadas	15	83.33%
Rechazadas	3	16.66%
Total pruebas	18	100%
Resultado	Con una valoración del 83.33% de aceptación, se denota efectivo	

Nota Elaboración Propia

4.2.4. Resultado Total

La valoración de todas las pruebas realizadas en los anteriores puntos, obteniendo los resultados de cada uno de ellos será la base para el análisis de del resultado total

Tabla 4. 5
Resultados de la Prueba Total

PRUEBAS		
PRUEBAS	CANTIDAD	PORCENTAJE
Aceptadas	39	88.63%
Rechazadas	5	11.37%
Total pruebas	44	100%
Resultado	Con una valoración del 90.90% de aceptación, se denota efectivo	

Nota. Elaboración Propia

El resultado es aprobado ya que sobrepasa el 94% de la aceptación requerida, por tanto, denotamos que el proyecto es efectivo para su uso y aplicación, pasando la prueba con un resultado optimo en las evaluaciones.

4.3. PRUEBA DE HIPÓTESIS

En este punto el objetivo central es el de demostrar que la hipótesis planteada en el primer capítulo se cumplió, para esto usaremos herramientas de tipo estadístico, al usar el método estadístico se podrán realizar pruebas unitarias al sistema como también pruebas globales de todo su funcionamiento.

4.4. EVALUACIÓN DE LOS CASOS DE PRUEBA

En este punto recordaremos el resumen de los resultados obtenidos mediante las pruebas de aceptación por iteraciones desarrolladas con anterioridad, siendo estos los siguientes.

Pruebas Totales	44
Pruebas Aceptadas	39
Pruebas Reprobadas	5
Porcentaje de éxitos	88.63%

Porcentaje de Fracayos 11.37%

Como se observa en los resultados, el porcentaje de éxitos es del 91% esperado al momento de plantearse la hipótesis, pero para comprobar de manera cuantificable que este valor se asemeja al valor esperado, se realiza una prueba de hipótesis estadística.

4.5. PROCEDIMIENTO DE LA PRUEBA DE HIPÓTESIS

Una hipótesis es una suposición que se establece como base de una investigación que puede confirmar o negar su validez, su función principal es de marcar el problema que se va a investigar considerando componentes tales como lugar, características de los sujetos tiempo y otros.

4.5.1. Definición de la Hipótesis en Investigación

H₀: “Con la ayuda de la ingeniería de software, herramientas y metodologías se tiene un modelo de realidad aumentada aplicado al control de tramos de transporte público basado en Android y código QR el cual tendrá una eficiencia del 90%”.

H₁: Se Rechaza H₀

En este caso se espera que el porcentaje de éxitos sea igual o mayor a 95%, tomando un nivel de significancia del 5%.

$$H_0: p_0 \geq 0.9$$

$$H_1: p_0 < 0.9$$

4.5.2. Evaluación de Resultados

Para determinar si el porcentaje de éxitos obtenido en las pruebas puedan ser considerado cercano al 90% 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.

$p_0 = 0.9$
 $X = 39$
 $n = 44$
 $\alpha = 0.5$

4.5.3. Determinación de la Región Crítica

La región crítica para la hipótesis planteada es la siguiente:

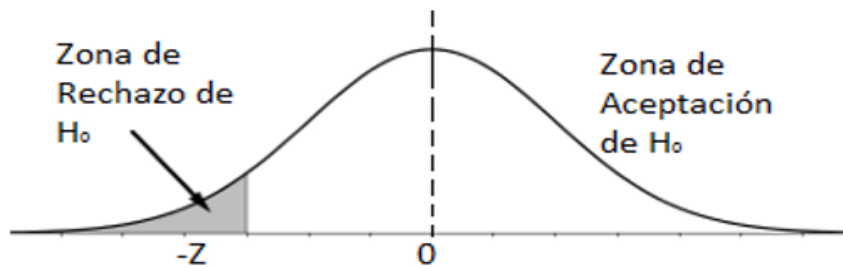


Figura 4. 11 Región crítica para la hipótesis (Elaboración propia)

Como n se refiere en este caso al número de pruebas, en este caso 44, el punto crítico a usar es $-Z_0$ y se determina mediante:

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

Este valor se halla de la tabla de la función de distribución normal, para obtener este 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

Tabla 4. 6

Resultado tabla de la función de distribución normal

z	...	0.04
...		↓
1.6	→	0.94950

Nota. - Elaboración propia

El valor de Z se obtiene sumando ambos valores:

$$-Z_{0.95} = -1.6 + 0.04 = -1.64$$

4.5.4. Cálculo estadístico de la prueba

Como se conoce el número total de individuos en el espacio muestral, el valor estadístico de la prueba se obtiene mediante la fórmula.

$$Z = \frac{X - n * p_0}{\sqrt{n * p_0(1 - p_0)}}$$

Remplazando valores y haciendo los cálculos correspondientes obtenemos:

$$Z = \frac{40 - 44 * 0.9}{\sqrt{44 * 0.9(1 - 0.9)}}$$

$$Z = \frac{40 - 44 * 0.9}{\sqrt{44 * 0.9(1 - 0.9)}} = -0.30$$

4.5.5. Toma de decisión

En base a los resultados obtenidos

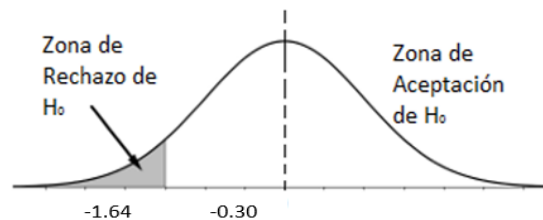


Figura 4. 12 Distribución de Z en el grafico para la toma de decisiones (Elaboración propia)

El promedio de éxito del prototipo al momento de recorrer las muestras se acerca al 90%, por tanto, como se acepta H_0 , se concluye la hipótesis:

H_0 : "Con la ayuda de la ingeniería de software, herramientas y metodologías se tiene un modelo de realidad aumentada aplicado al control de tramos de transporte publico basado en Android y código QR, que tiene una eficiencia del 90% para colaborar en el control de rutas.



CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5. CONCLUSIÓN Y RECOMENDACIÓN

5.1. CONCLUSIONES

La presente investigación se ha dedicado al estudio de una problemática en la población de usuarios del transporte Público en la ciudad de El Alto, la misma tiene bases de las infracciones que realizan los conductores al llevar a cabo el llamado “trameaje” en horarios pico de su jornada laboral.

Es por esta razón que se llevó a cabo la investigación del caso, aplicando el Modelo de Realidad Aumentada para el Control de tramos del transporte público dimos soluciones alternativas y modernas para un mejor servicio de autotransporte en el caso particular del Sindicato “Simón Bolívar”.

En el desarrollo del trabajo de investigación que ha dado lugar a la presente tesis se han alcanzado los objetivos inicialmente planteados en cuando a:

5.1.1. CUMPLIMIENTO DE LOS OBJETIVOS.

- ✓ Respecto al objetivo general, se desarrolló un Modelo de Realidad Aumentada que pueda realizar el control de tramos del transporte público de una línea de autotransporte ya determinada, en este caso la línea 643, del Sindicato “Simón Bolívar” este modelo se aplica con un dispositivo móvil y el escaneado del código QR que tiene cada movilidad (previa dotación), le permite al usuario conocer el recorrido que tendrá el vehículo de servicio activando la geolocalización del propio dispositivo móvil.
- ✓ Para los objetivos específicos, previa implementación de los prototipos realizados se llevó a cabo un análisis en cuanto al cumplimiento de las rutas asignadas de la línea de autotransporte.
- ✓ Posteriormente se plantea el Modelo de Realidad Aumentada para el control de tramos como posible solución al denominado “trameaje” en la determinada línea.
- ✓ Se diseñó la aplicación para dispositivos móviles Android con un nivel de Realidad Aumentada, para controlar las rutas de la línea en cuestión.

- ✓ Para la generación de los códigos QR se implementa nuevas tecnologías que además nos permiten tener una información única y propia de cada conductor perteneciente al sindicato de autotransporte.
- ✓ En la aplicación móvil RUApp se desarrolló un lector de código QR para el acceso rápido a la información de la ruta a recorrer además de información adicional de cada conductor de la movilidad.
- ✓ Cuando el usuario verifique mediante la aplicación que el conductor no cumplió con la ruta asignada, podrá realizar la denuncia correspondiente con la captura de pantalla y los datos del usuario como respaldo de dicha denuncia.
- ✓ Se realizó las pruebas de calidad en cada iteración citada por la metodología Móvil-D, adicionalmente se tienen un capítulo donde se recurre a las métricas de calidad para los prototipos, la misma se denomina QSOS.

De acuerdo a las pruebas realizadas se demostró que el modelo es útil para los usuarios que requieren los servicios de transporte público, brindando así un servicio más eficiente y oportuno a la población.

5.2. RECOMENDACIONES

Dentro de una investigación tan ambiciosa como lo fue este, siempre se desea que haya una mejora continua del mismo, analizar las posibilidades de la adición de nuevos módulos donde se pueda tener más información acerca de la movilidad en servicio, así orientándonos hacia la seguridad de cada persona en altas horas de noche.

Se considera interesante investigar sobre otros aspectos relacionados con el control de tramos para el sector del autotransporte donde se pueda aplicar la tecnología que emerge y que cada vez está mucho más al alcance de cada persona en la sociedad.

BIBLIOGRAFÍA

[Andrés Castellanos Ramírez, 2016] Manual de gestión logística del transporte público.

[Miguel Alejandro López Olivera, 2015] El transporte de pasajeros y el sistema vial

[Paul Gustavo Peralta Fajardo, 2018] Realidad Aumentada (QR codes) como herramienta interactiva y potenciadora de información.

[Paul Gustavo Peralta Fajardo, 2012] Realidad Aumentada como nuevo concepto de la publicidad online a través de los Smartphones

[Pressman, R., 2002] Realidad Aumentada como nuevo concepto de la publicidad online a través de los Smartphones

[Andrea Paz, 2011] Códigos de barras de la vida: introducción y perspectiva de Barcode of life

[Ricardo Moya, 2018] Manual básico Android Studio

[Tomas, Jesús, 2018] El gran libro de Android 2ª edición

[Jaime Aranaz Tudela., 2009] Desarrollo De Aplicaciones Para Dispositivos Móviles Sobre La plataforma Android

[Rodríguez, E, 2014] La Geolocalización Coordenadas hacia el Éxito

[Alcón Ayuso, 2007] Aplicación Web Para La Geolocalización Y Monitorización En Tiempo Real De Los Recursos Integrantes de una Red Grid Facultad de Informática. Universidad Complutense de Madrid Desarrollo De Aplicaciones Para Dispositivos Móviles Sobre La plataforma Android

[Nieto Masot, A. (Ed.), 2016] Tecnologías de la información geográfica en el análisis espacial. Aplicaciones en los Sectores Público, Empresarial y Universitario.

[José,2020] Cómo calcular distancias con Leaflet y otros métodos de conversión

[José L. García G., 2019] Creaciones de visores de mapas web de Leaflet

[Franz P.C., 2018] Manual de prácticas de QGIS

[Sánchez Parra, 2016] QGIS User Guide publicación 2.8

[Alberto Orpez M., Francesc Uribe Porta, 2015] Georreferenciación masiva y visualización de datos de biodiversidad.

[Jusoh., Yahaya.,Chamili, Che Pa, 2012] Los criterios de selección para la adopción de software de código abierto

[Adriana Gomez,Maria del C. Lopez,Migani, 2010] Un modelo de estimación de proyectos de software COCOMO

[Ana Maria Garcia Sanchez, 2014] Evaluación de métricas de calidad del software sobre un programa Java.

REFERENCIAS WEB

Puntos de trameaje en la Ciudad de El Alto

(www.001):<https://www.elaltodigital.com/sociedad/cuales-son-los-puntos-de-trameaje-en-el-alto/>

Ley para la prestación de servicio de autotransporte público en Bolivia

(www.002); https://sionet.oopp.gob.bo/Public/doc/RA_018760.pdf

QR Code

(www.003): <http://www.denso-wave.com/qrcode/aboutqr-e.html>

QR Code Generator

(www.004): <https://es.qr-code-generator.com/qr-codes-on/books/>

Lector de códigos QR en Android Studio

(www.005):<https://www.desarrollolibre.net/blog/android/como-crear-un-lector-de-codigos-qr-en-android-con-android-studio#.Xv7ID21KjIU>

Librerías ZXing

(www.006):<https://code.tutsplus.com/es/tutorials/android-sdk-create-a-barcode-reader--mobile-17162>

Creación de WebView con Android Studio

(www.007):<https://www.gestionatuweb.net/crear-webview-con-android-studio-para-cargar-una-web-en-una-app-android/>

Como crear WebView

(www.008):<https://developer.android.com/guide/webapps/webview?hl=es-419>

Tutoriales Leaflet

(www.009): <https://leafletjs.com/examples.html>

Marcadores animados con Leaflet

(www.010):<https://mappinggis.com/2019/02/creando-marcadores-animados-con-leaflet-svg-gifs-y-font-awesome/>

Guia Manual de QGIS

(www.011):

https://docs.qgis.org/2.14/es/docs/user_manual/preamble/foreword.html

OpenMapTiles

(www.012): <https://openmaptiles.org/docs/website/leaflet/>

Tiles

(www.013): <https://wiki.openstreetmap.org/wiki/Tiles>

Cálculo de rutas en un mapa web de leaflet

(www.014): <https://mappinggis.com/2016/08/calculo-de-rutas-en-un-mapa-web-de-leaflet/>

Calcular rutas con OpenStreetMap partiendo de la geolocalización del usuario

(www.015):<https://www.gestionatuweb.net/calcular-rutas-con-openstreetmap-partiendo-de-la-geolocalizacion-del-usuario/>

MappingGIS

(www.016): <https://mappinggis.com/2014/06/mejores-plugins-para-leaflet/>

Mapas interactivos con leaflet.js

(www.017): <https://www.adictosaltrabajo.com/2016/06/22/mapas-interactivos-con-leaflet-js/>

Métricas de calidad de software

(www.018): <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/metricas-de-calidad-de-software-una-solucion-excelente>

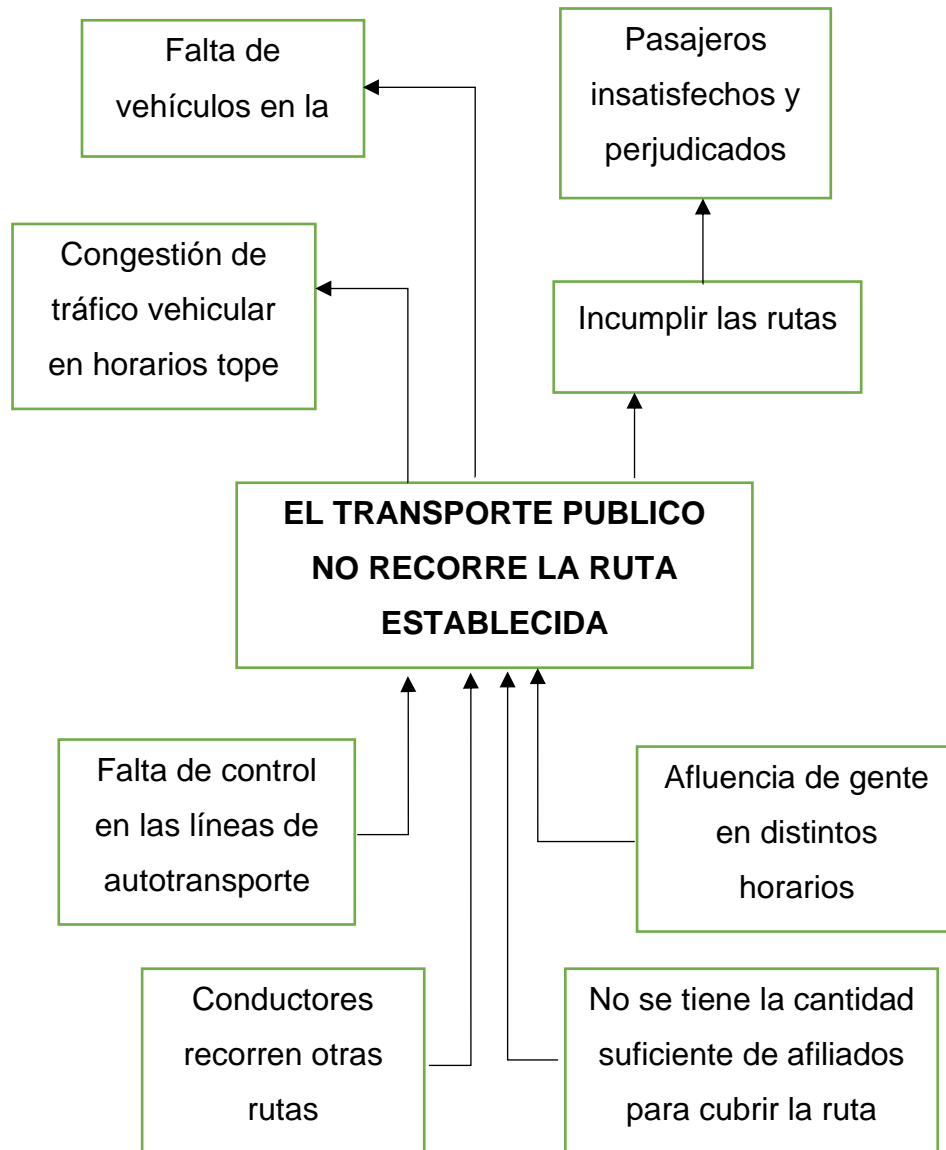
QSOS

(WWW.019):http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992017000200007

ANEXOS

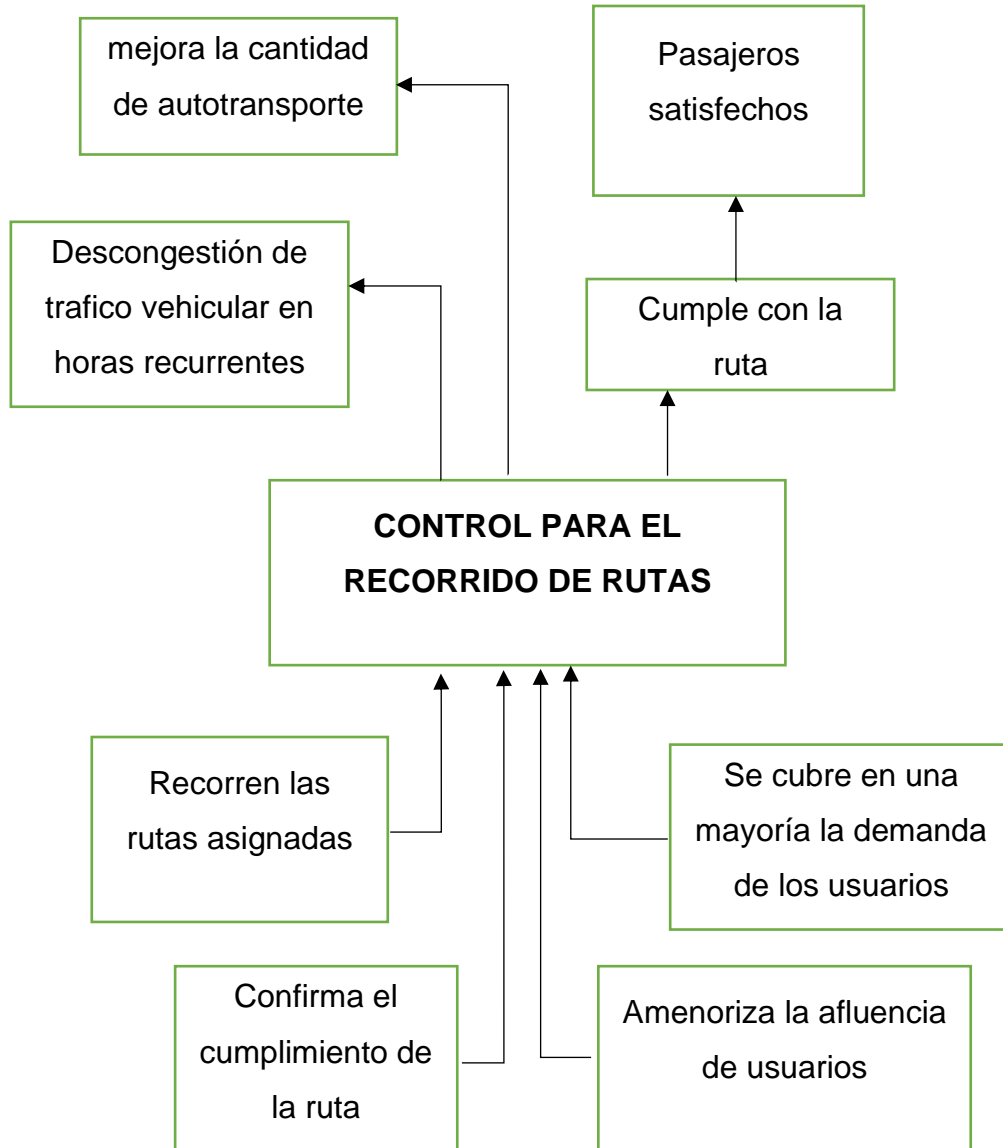
Anexo A

ARBOL DE PROBLEMAS



Anexo B

ARBOL DE OBJETIVOS



Anexo C

MANUAL PARA EL SISTEMA GESTOR DE CODIGO QR:

Conocimientos Fundamentales del Sistema

Edición 1.0

Julio 2020

Contenido

Introducción i-3

Objetivos del Manual I-3

1 Modo de ingreso a la aplicación

Objetivo 1-4

Visualización de apertura 1-4

Visualización de panel de control 1-5

2 Funciones del Panel de Control

Objetivo 2-6

Funciones de panel de control 2-6

3 Manipulación de datos mediante funciones

Objetivo 3-7

Ingreso de nuevos registros 3-7

Registro de Administrador 3-8

Registro de Afiliado 3-8

Generación de Código QR 3-8-9-10

Modificación de datos 3-10

Eliminación de registro 3-11

Búsqueda de registro 3-12

Perfil

Antes de Empezar este Manual

Antes de comenzar este curso, debe tener conocimiento de términos básicos de informática además de información adicional (parte administrativa) sobre el Sindicato de transporte “Simón Bolívar”. El requisito es que esté familiarizado con algunos casos particulares dentro de la organización esto a petición estricta de la misma.

Organización del manual

El Manual para el Sistema Gestor de Código QR es un curso dirigido por los diseñadores del mismo, que ofrecen clases de capacitación. Las demostraciones en línea y las sesiones prácticas que sirven para reforzar los conceptos y las habilidades que se representan en él.

Introducción



Objetivos del Manual

Al finalizar este Manual, debería estar capacitado para:

- **Ejecutar la aplicación sin ningún problema**
- **Identificar los componentes estructurales principales de Sistema Gestor de Código QR**
- **Mostrar las funciones de la aplicación**
- **Describir las funciones de la aplicación**
- **Comprender el manejo de la aplicación**
- **Comprender los objetivos del Manual**

Objetivo

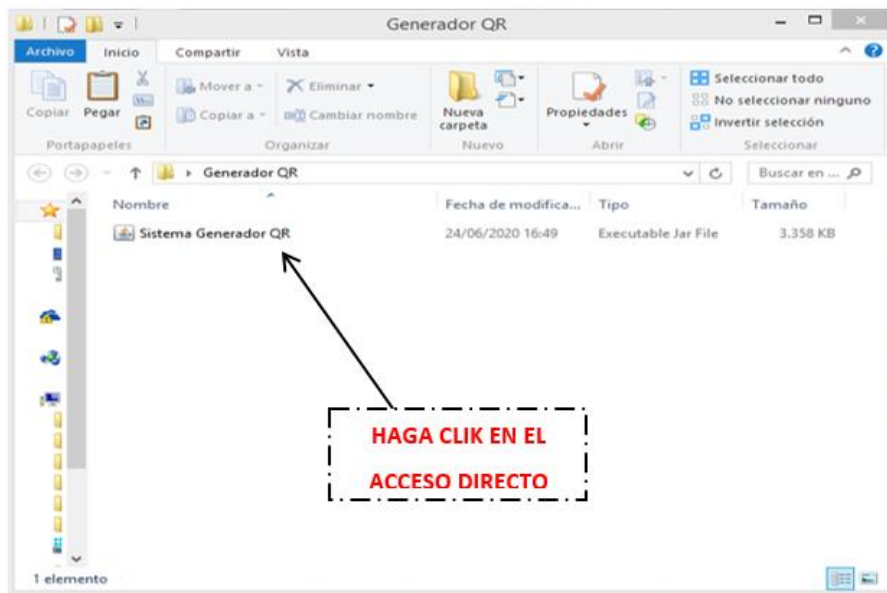
Este Manual tiene como finalidad dar en conocimiento al usuario el funcionamiento del sistema, conocerá la manera en el cual se debe ejecutar la aplicación, también se presentará imágenes con las cuales se facilitará la comprensión del manual, reconociendo algunas limitaciones de la aplicación.

1 Modo de ingreso a la aplicación

Objetivo

Dar a conocer la manera en el que se ingresa a la aplicación, de tal manera que se facilite el conocimiento para el manejo del sistema.

Visualización de Apertura



Este trascurso le llevara directamente con la ejecucion de la aplicación, mostrando una ventana de admisión, el cual nos lleva a la respectiva validacion de los datos del administrador, **Ci: número de carnet** , **Contraseña: *******, en aquel momento procuramos un clik en el Boton **Aceptar**.

Visualización de Panel de Inicio



2

Funciones del “Panel de Control”

Objetivo

Describir funcionamiento del *panel de control*, para una mejor comprensión del manejo del sistema, desglosaremos el panel de control, ya que está comprendido de botones que contienen distintas funciones.

Funciones del Panel de Control



1. Registro de Conductor: Dicho botón nos lleva al registro del administrador y conductor.

2. Modificar Datos: El botón nos lleva a la modificación de algún dato erróneo ingresado al momento del registro ya sea del conductor o del administrador.

3. Eliminar Conductor: Si fuera el caso de la baja de algún afiliado del Sindicato esta opción nos facilita la eliminación del mismo.

4. Buscar Conductor. - Con el fin de filtrar y verificar los datos del conductor se procederá con el botón buscar.

3 Manipulación de datos mediante funciones

Objetivo

Mostrar el manejo de los botones ajustados para la Adición, Modificaciones, Eliminación y Búsqueda, dejando muy en claro la actividad que realiza cada apartado de las operaciones.

Adición de un nuevo registro (Conductor y Administrador)

SISTEMA GENERADOR QR

Alta de Nuevo Afiliado

Tipo de Registro

- Usuario Administrador
- Usuario Afiliado

Validación

Guardar Borrar Imprime QR

Datos Usuario

NOMBRE: Miguel

AP. PATERNO: Cruz

AP. MATERNO: Sirpa

CI: 6780638 EXP. LA PAZ

SEXO: MASCULINO

NACIONALIDAD: Boliviana

LIC. CATEGORIA:

CONTRASENA: princesa1

DESCRIPCION: Administracion de rote

Datos Sindicato

CI no registrado

CI valido

NOMBRE SIND.:

NRO. DE LINEA: Seleccionar Linea

Datos Vehiculo

TIPO DE VEHICULO: Elige Transporte

PLACA:

COLOR:

1.1 Registro de Administrador. – Se debe seleccionar la opción **Usuario Administrador** para adicionar un nuevo usuario con perfil de administrador, además de llenar los datos requeridos y presionar el botón **Guardar**, el mismo podrá generar los códigos QR una vez generado registrado el conductor.

1.2 Registro de Afiliado. – Se debe seleccionar la opción de **Usuario Afiliado** para adicionar a un nuevo conductor, además de llenar los datos que se requieren para generar el código QR, una vez culminada el registro se debe guardar el cambio con el botón **Guardar**.

1.2.1 Generación del código QR. – Una vez terminado el **llenado y guardado** de datos se habilitará el botón **Imprime QR** el cual se debe presionar para generar el código QR, el mismo se abrirá en el visor de imágenes del sistema para luego proceder a imprimirlo.

Registro de Afiliado:

SISTEMA GENERADOR QR

Alta de Nuevo Afiliado

Tipo de Registro

Usuario Administrador

Usuario Afiliado

Validación

Guardar

Borrar

Imprime QR

Datos Usuario

NOMBRE: Carlos

AP. PATERNO: Quispe

AP. MATERNO: Coronel

CI: 7030262 EXP. LA PAZ

SEXO: MASCULINO

NACIONALIDAD: Boliviana

LIC. CATEGORIA: C

CONTRASENA: []

DESCRIPCION: []

Datos Sindicato

NOMBRE SIND.: Simon Bolivar

NRO. DE LINEA: 604

Datos Vehículo

TIPO DE VEHICULO: MINIBUS

PLACA: 4023-SDF

COLOR: AZUL

Ayuda Salir

Guardado de afiliado:

SISTEMA GENERADOR QR

Alta de Nuevo Afiliado

Tipo de Registro

- Usuario Administrador
- Usuario Afiliado

Validación

- Guardar
- Borrar
- Imprime QR

Datos Usuario

NOMBRE: Carlos

AP. PATERNO: []

AP. MATERNO: []

CI: 7030262 EXP. LA PAZ

SEXO: MASCULINO

NACIONALIDAD: Boliviana

LIC. CATEGORIA: C

CONTRASEÑA: []

DESCRIPCION: []

Datos Sindicato

NOMBRE SIND.: Simon Bolivar

NRO. DE LINEA: 604

Datos Vehiculo

TIPO DE VEHICULO: MINIBUS

PLACA: 4023-SDF

COLOR: AZUL

Mensaje: Datos Guardados

Ayuda | Salir

Habilitación del botón *Imprime QR*:

SISTEMA GENERADOR QR

Alta de Nuevo Afiliado

Tipo de Registro

- Usuario Administrador
- Usuario Afiliado

Validación

- Guardar
- Borrar
- Imprime QR

Datos Usuario

NOMBRE: []

AP. PATERNO: []

AP. MATERNO: []

CI: [] EXP. Elige ciudad --

SEXO: Seleccionar --

NACIONALIDAD: []

LIC. CATEGORIA: []

CONTRASEÑA: []

DESCRIPCION: []

Datos Sindicato

NOMBRE SIND.: []

NRO. DE LINEA: Seleccionar Linea --

Datos Vehiculo

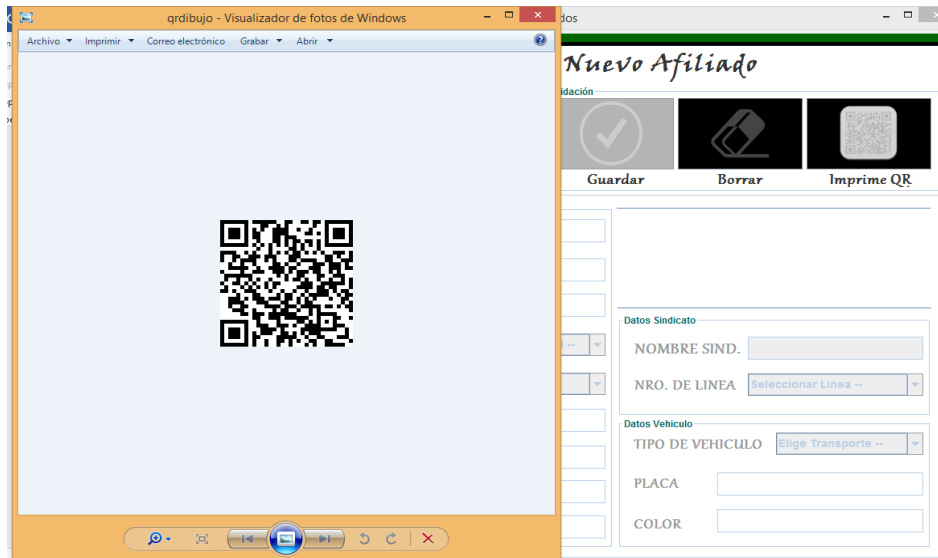
TIPO DE VEHICULO: Elige Transporte --

PLACA: []

COLOR: []

Ayuda | Salir

Impresión del código QR:



Modificado de datos de un registro

Se pueden realizar modificaciones tomando muy en cuenta que los debemos guardar una vez hecha la modificación, con el objetivo de no perder ningún dato modificado, **El parámetro de búsqueda es el número de CI.**



Eliminar un registro

El parámetro de búsqueda para la eliminar un registro es el CI carnet de identidad, solo necesitamos puntualizarnos en el icono mostrado, aparecerá una ventana donde nos aclaran si queremos eliminar dicho registro, a consideración realizamos la acción.

Nota. - una vez eliminado el registro no se puede recuperar ya que se lo elimina desde la memoria del sistema.



Búsqueda de un registro

Para realizar la búsqueda de registros nos dirigimos al icono de la imagen, se mostrará el parámetro de búsqueda que es el número de carnet de identidad, se realiza el filtrado para fines correspondientes.







¿QUE ES RUApp?

Una aplicación móvil donde puedes controlar el cumplimiento de la ruta de la línea de autotransporte.

¿Ventajas de RUApp?

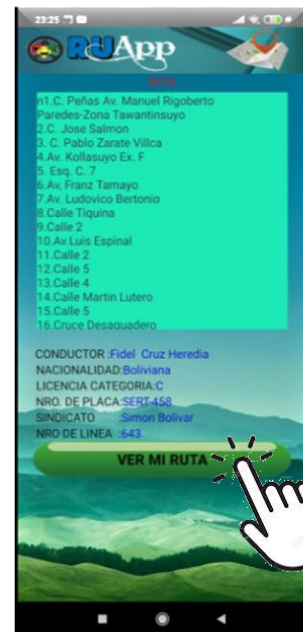
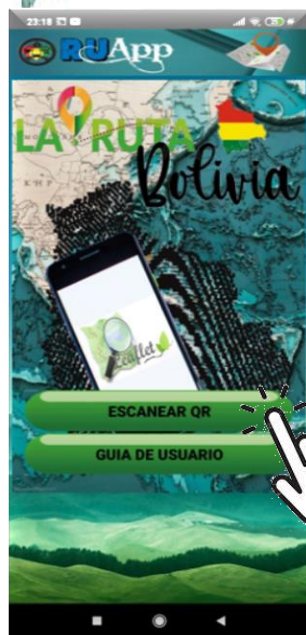
- Permite Denunciar el "trameaje" o desvió de la ruta en tiempo real.
- Herramienta pensada ti, que usas el servicio de transporte público.
 - Solo 3 pasos para controlar tu ruta.

¿Cómo empezar con RUApp?

iiVamos a verlo en este Manual de Usuario en 3 Pasos!!

PASO 1

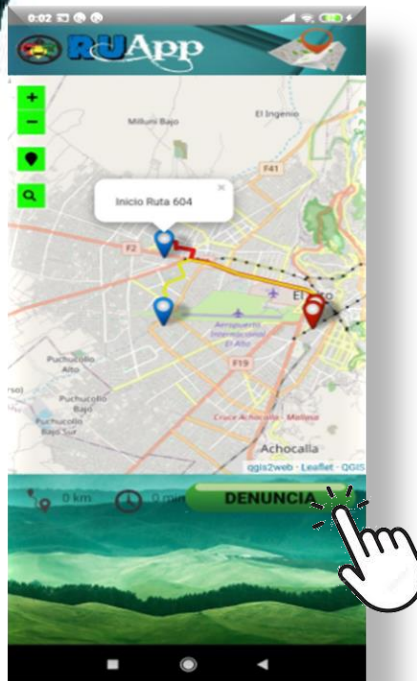
Ingresa a la aplicación y escanea el código QR que está pegado en la ventana lateral del minibús.



En la 2da pantalla tendrás una información de las calles y avenidas del recorrido de la ruta.
Ahora presiona **“VER MI RUTA”**

PASO 2

De inmediato tendrás el mapa con la ruta trazada y empieza en recorrido!!



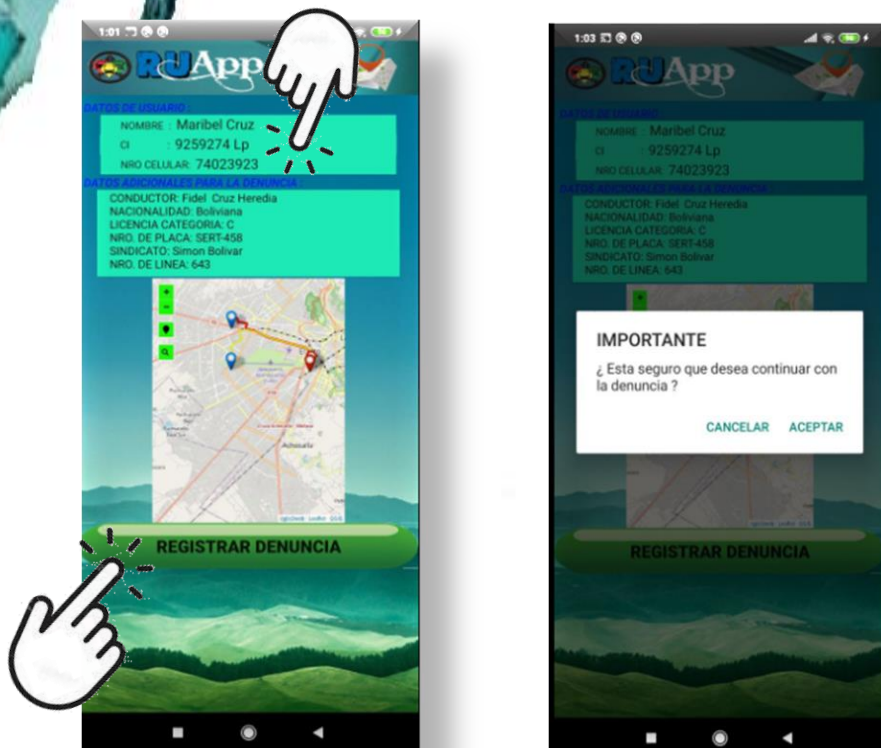
Ojo: No olvides darle permiso a la App de usar el GPS

iiAlerta!! Si el conductor no cumple con la ruta asignada o tramea

Presiona el botón
"DENUNCIA"

PASO 3

Y como último paso debes respaldar la denuncia llenando tu nombre, CI. y número de celular,



Presionar el botón **"REGISTRAR DENUNCIA"** y confírmalo.

¡¡Listo!! Se registró tu denuncia.

AVAL TUTOR ESPECIALISTA

La Paz, 09 de julio de 2020

Señor
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II
Presente.

Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero,

Mediante la presente tengo a bien comunicarle mi conformidad con el perfil de tesis de grado **“MODELO DE REALIDAD AUMENTADA PARA EL CONTROL DE TRAMOS DEL TRANSPORTE PÚBLICO BASADO EN ANDROID Y CODIGO QR CASO: SINDICATO “SIMON BOLIVAR”** que propone la postulante **Univ. Maribel Ana Cruz Sirpa**, con cedula de identidad 9259274 LP. Para su defensa pública, la evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente a la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,



LIC. Norma Mamani Quispe
TUTOR ESPECIALISTA

cc. Arch.

AVAL TUTOR REVISOR

La Paz, 10 de julio de 2020

Señor
Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II
Presente.

Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero,

Mediante la presente tengo a bien comunicarle mi conformidad con el perfil de tesis de grado **"MODELO DE REALIDAD AUMENTADA PARA EL CONTROL DE TRAMOS DEL TRANSPORTE PÚBLICO BASADO EN ANDROID Y CODIGO QR CASO: SINDICATO "SIMON BOLIVAR"** que propone la postulante **Univ. Maribel Ana Cruz Sirpa**, con cedula de identidad 9259274 LP. Para su defensa publica, la evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente a la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente,


LIC. María Magdalena Aguilar Guanto
TUTOR REVISOR

cc. Arch.

AVAL TUTOR METODOLÓGICO

La Paz, 10 de julio de 2020

Señor
Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA DE INGENIERIA DE SISTEMAS
Presente.

Ref.: AVAL DE CONFORMIDAD

Distinguido Ingeniero,

Mediante la presente tengo a bien comunicarle mi conformidad con el perfil de tesis de grado **“MODELO DE REALIDAD AUMENTADA PARA EL CONTROL DE TRAMOS DEL TRANSPORTE PÚBLICO BASADO EN ANDROID Y CODIGO QR CASO: SINDICATO “SIMON BOLIVAR”** que propone la postulante **Univ. Maribel Ana Cruz Sirpa**, con cedula de identidad 9259274 LP. Para su defensa pública, la evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente a la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.


Atentamente,

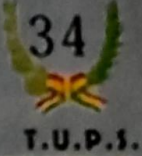


Ing. Enrique Flores Baltazar
TUTOR METODOLÓGICO TALLER II

cc. Arch.


AVAL INSTITUCIÓN

**AFILIADO CONFEDERACION DE CHOFERES DE BOLIVIA**
CENTRAL UNICA DE TRANSPORTE URBANO DE PASAJEROS
PERSONERIA JURIDICA NO. 168023

**34**
T.U.P.S.

La Paz, 10 de julio de 2020

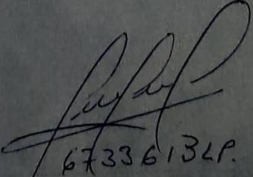
Señor
Ing. Enrique Flores Baltazar
Universidad Pública de El Alto
Carrera Ingeniería de Sistemas
TUTOR METODOLÓGICO TALLER II



Presente.

Ref.: AVAL DE CONFORMIDAD

El Sí ndicato Simon Bolivar, a través de la Secretarí a General de la organizaci3n, en uso de las atribuciones, felicita la iniciativa de la Univ. Maribel A. Cruz Sirpa el proponer el proyecto “Modelo de Realidad Aumentada para el Control de Tramos del Transporte P3blico Basado en Android y C3digo QR” Caso : Sindicato “Simon Bolivar”, el cual es una nueva alternativa que esfuerza las polí ticas de calidad de servicio que se viene implementando en la organizaci3n: en tal virtud concede el aval y la conformidad para la implementaci3n del proyecto.


67336134P.
Sr. Carlos Quispe Coronel
Srio. General Sindicato Mixto de Transporte
“Simon Bolivar”