



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniero de Sistemas**

**TEMA:
DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA
ADMINISTRACIÓN DE LA INFORMACIÓN DE DATOS DE PASAJEROS QUE
HACEN USO DE TRANSPORTE INTERPROVINCIAL, RUTA QUITO-
ATACAMES.**

**AUTOR:
HENRY LEONARDO BARRAGÁN PÉREZ**

**TUTOR:
FRANKLIN EDMUNDO HURTADO LARREA**

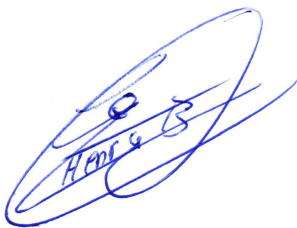
Quito, julio de 2019

CESIÓN DE DERECHOS DE AUTOR

Yo, HENRY LEONARDO BARRAGÁN PÉREZ, con documento de identificación N° 1721867891 manifesté mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación con el tema: “DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE LA INFORMACIÓN DE DATOS DE PASAJEROS QUE HACEN USO DE TRANSPORTE INTERPROVINCIAL, RUTA QUITO-ATACAMES”, mismo que ha sido desarrollado para optar por el título de INGENIERO DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada.

En concordancia, suscribo este documento en el momento que hago la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



HENRY LEONARDO BARRAGÁN PÉREZ

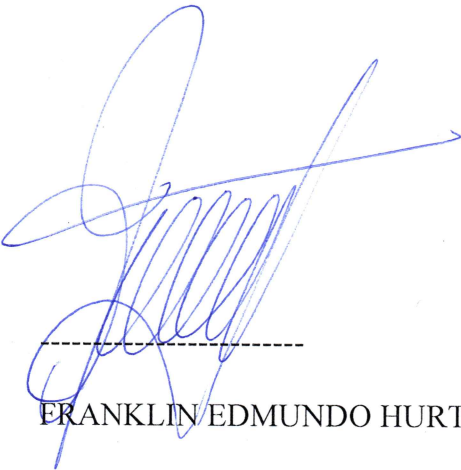
CI: 1721867891

Quito, julio de 2019

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE LA INFORMACIÓN DE DATOS DE PASAJEROS QUE HACEN USO DE TRANSPORTE INTERPROVINCIAL, RUTA QUITO-ATACAMES” realizado por HENRY LEONARDO BARRAGÁN PÉREZ, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, julio de 2019



FRANKLIN EDMUNDO HURTADO LARREA

CI: 1713382016

Dedicatoria

Para empezar, quiero dedicar este trabajo a cada persona que me brindó su apoyo en todo el camino arduo, fueron momentos difíciles, de angustia, desanimo, iras, impotencia y deseos de abandono, con una acción o palabra de las personas que estuvieron a mi lado, se transformaron en fuerza, alegría y determinación para que este día llegue.

A mi padre por ser la persona que admiro, por brindarme su apoyo, enseñarme con su experiencia lo que significa trabajo duro con determinación y esfuerzo, por guiarme y darme día a día una lección de superación, todas aquellas lecciones de vida dadas me ayudaron siempre a poder conllevar cada situación difícil que se presentó en mi carrera y superarlas para convertirme en una mejor persona.

A mi novia por ser la fuente de motivación e inspiración, por apoyarme en cada momento difícil, por darme su sonrisa y amor incondicional y tener la habilidad para poder transformar cada situación complicada en una situación favorable con el simple hecho de explicarme las cosas desde otro punto de vista y hacerme entrar en razón, por convertirse en un pilar fundamental para poder culminar este último peldaño y cumplir con mi objetivo.

Extiendo una dedicatoria especial a mi madre, no tengo ni encuentro las palabras para agradecerte madre mía y no sé si sean suficientes estas palabras, mi madre la única persona que creyó en mi desde el inicio que decidí estudiar Ingeniería en Sistemas, aunque algunas personas no creían y se opusieron, mi madre seguía creyendo, cuando renuncié mi madre siguió creyendo y luchó arduo para conseguirme un nuevo camino para estudiar, aun cuando mi madre no sabía de ingeniería ella me enseñaba porque sabía cómo se tienen que hacer las cosas, cada momento que tuve dudas y quise renunciar otra vez mi madre siempre estuvo ahí para darme el empujón necesario para continuar con mis estudios. Esta ingeniería es tuya madre mía.

Henry Leonardo Barragán Pérez

Agradecimientos

A todas las personas que estuvieron junto a mí los cuales supieron apoyarme, guiarme y brindarme su ayuda en todo el camino para poder obtener este reconocimiento.

A mi director de tesis el Ingeniero Franklin. Hurtado, por brindarme su apoyo y darme todas las guías necesarias, compartir conmigo sus conocimientos, su sabiduría, su experiencia, por estar siempre pendiente y exigirme un nivel alto de calidad, lo cual me motivo cada día para culminar con éxito y satisfacción la carrera de Ingeniería en Sistemas.

A la Universidad Politécnica Salesiana por brindarme la oportunidad para formar parte de su cuerpo de estudiantes que en un futuro se convertirían en ingenieros de sistemas con calidad humana, por asegurar de que en todo momento tenga ingenieros de calidad y que sepan transmitir los conocimientos adecuados para cumplir con el objetivo general.

Henry Leonardo Barragán Pérez

ÍNDICE

INTRODUCCIÓN	1
Antecedentes.....	1
Problema.....	1
Justificación.....	1
Objetivo General.....	2
Objetivos Específicos	2
Marco Metodológico	3
Aplicación de XP.....	3
Historias de usuario	4
Utilización de notaciones.....	7
Capítulo 1.....	8
1.1 Marco teórico.....	8
1.1.1 Automatización.....	8
1.1.2 Transporte público interprovincial de pasajeros.....	8
1.1.3 Manifiesto de pasajeros del servicio de transportes interprovinciales	9
1.1.4 Metodología XP.....	9
1.1.4.1 Modelo Incremental.....	9
1.1.4.2 Comunicación.....	10
1.1.4.3 Simplicidad.....	10
1.1.4.4 Proceso XP	10
1.1.4.5 Fase I exploración.....	11
1.1.4.5.1 Historias de usuario	11
1.1.4.6 Fase II planificación	11
1.1.4.7 Fase III Iteraciones	11
1.1.4.8 Fase III de puesta en producción	12
1.1.4.8.2 Pruebas de aceptación.....	12
1.1.4.8.3 Detección y correcciones de errores.....	12
1.1.4.9 Adaptación de metodología XP	13
1.1.5 Programación Android	13
1.1.5.1 Componentes de una aplicación móvil.....	13
1.1.5.2 Actividades	14
1.1.5.3 Proceso y diseño de una aplicación móvil.....	14
1.1.5.4 Conceptualización	15
1.1.5.5 Definición	15
1.1.5.6 Diseño.....	15

1.1.5.7 Desarrollo	16
1.1.5.8 Publicación	16
1.1.6 Programación Web	16
1.1.6.1 El cliente	17
1.1.6.2 PHP	17
1.1.6.3 CSS	18
1.1.6.4 El Servidor	19
1.1.6.5 Ejecución de páginas web.....	19
1.1.7 Arquitectura MVC	20
1.1.7.1 Modelo.....	20
1.1.7.2 Vista.....	20
1.1.7.3 Controlador	21
1.1.8 Laravel	21
1.1.8.1 Funcionamiento de framework laravel	22
1.1.8.2 Eloquent ORM.....	22
1.1.8.3 Apirest	22
1.1.9 MySQL	23
1.1.9.1 Base de datos	23
1.1.10 PSR-2.....	24
Capítulo 2.....	25
2.1 Análisis	25
2.1.1 Proceso de registro manual de pasajeros	25
2.1.2 Proceso registro con la aplicación móvil	26
2.2 Casos de uso	27
2.2.1 Caso de uso Administrador de Cooperativa	27
2.2.2 Caso de uso organismos de socorro 911	28
2.2.3 Caso de uso para usuario de aplicación móvil.....	28
2.3 Resumen de las tablas de la base de datos.....	29
2.3.1 Diagrama entidad relación de la base de datos.....	31
Capítulo 3.....	32
3.1 Estándares de programación.....	32
3.2 Arquitectura	33
3.2.1 Arquitectura aplicación web.....	33
3.2.2 Arquitectura aplicación móvil	35
3.3 Clases y Métodos.....	36
3.3.1 Clases aplicación web.....	36
3.3.2 Métodos aplicación web	37

3.2.3 Clases aplicación móvil	38
3.2.4 Métodos aplicación móvil	38
3.3 Diagrama de clases	39
3.3.1 Diagrama de clase aplicación web.....	39
3.3.2 Diagrama de clase aplicación móvil	40
3.4 Código relevante.....	41
3.4.1 Manejo de la información y escaneada del código de barras	41
3.4.2 Consultar datos del chofer, conexión entre la aplicación móvil y web.	43
3.4.3 Generar la lista de pasajeros en la aplicación móvil.....	45
3.4.4 Validación para activar y desactivar los viajes.....	47
3.4.5 Generar la lista de pasajeros	48
3.4.6 Llenar la cédula manualmente y al poner el último número automáticamente sale los datos de los pasajeros.	49
3.5 Pruebas.....	50
3.5.1 Plan de pruebas.....	50
3.5.1.1 Pruebas funcionales	50
3.5.1.2 Pruebas de usabilidad	56
3.5.1.3 Pruebas de rendimiento	66
3.5.1.3.1 Pruebas de carga	66
3.5.1.3.2 Pruebas de rendimiento aplicación móvil.....	66
3.6 Perspectiva visual del software.....	70
3.6.1 Perspectiva visual de la aplicación web	70
3.6.2 Perspectiva visual de la aplicación móvil.....	73
CONCLUSIONES.....	75
RECOMENDACIONES	76
LISTA DE REFERENCIAS.....	77

ÍNDICE DE TABLAS

Tabla 1: Resumen historias de usuario.	4
Tabla 2: Componentes de la base de datos.	29
Tabla 3: Clases de la aplicación web.	36
Tabla 4: Métodos de la aplicación web.....	37
Tabla 5: Clases de la aplicación móvil.	38
Tabla 6: Métodos de la aplicación móvil.	38
Tabla 7: Resumen de las pruebas funcionales aplicación web.	51
Tabla 8: Resumen de las pruebas funcionales aplicación móvil.....	55
Tabla 9: Resultados la pregunta 1.	56
Tabla 10: Resultados la pregunta 2.	57
Tabla 11: Resultados la pregunta 3.	58
Tabla 12: Resultados la pregunta 4.	59
Tabla 13: Resultados la pregunta 5.	60
Tabla 14: Resultados la pregunta 6.	61
Tabla 15: Resultados la pregunta 7.	62
Tabla 16: Resultados la pregunta 8.	63
Tabla 17: Resultados la pregunta 9.	64
Tabla 18: Resultados la pregunta 10.	65
Tabla 19: Teléfonos inteligentes usados para las pruebas.	67

ÍNDICE DE FIGURAS

Figura 1: Proceso del registro manual de pasajeros.....	25
Figura 2: Proceso registro con la aplicación móvil.....	26
Figura 3: Caso de uso del administrador de la cooperativa.....	27
Figura 4: Caso de uso del administrador de los organismos de socorro 911.....	28
Figura 5: Caso de uso del usuario de la aplicación móvil.....	28
Figura 6: Diagrama entidad relación de la base de datos.....	31
Figura 7: Arquitectura de la aplicación web.....	34
Figura 8: Arquitectura de la aplicación móvil.....	35
Figura 9: Diagrama de clases de la aplicación web.....	40
Figura 10: Diagrama de clases de la aplicación móvil.....	40
Figura 11: Código para extraer en número de cédula.....	42
Figura 12: Conexión entre la aplicación web y la aplicación móvil.....	45
Figura 13: Código para crear la lista de pasajeros.....	46
Figura 14: Validación para activar y desactivar los viajes.....	47
Figura 15: Generar de la lista de pasajeros.....	49
Figura 16: Llenado automático para ingresar del número de la cédula manualmente.....	49
Figura 17: Resultado porcentual de la pregunta 1.....	56
Figura 18: Resultado porcentual de la pregunta 2.....	57
Figura 19: Resultado porcentual de la pregunta 3.....	58
Figura 20: Resultado porcentual de la pregunta 4.....	59
Figura 21: Resultado porcentual de la pregunta 5.....	60
Figura 22: Resultado porcentual de la pregunta 6.....	61
Figura 23: Resultado porcentual de la pregunta 7.....	62
Figura 24: Resultado porcentual de la pregunta 8.....	63
Figura 25: Resultado porcentual de la pregunta 9.....	64
Figura 26: Resultado porcentual de la pregunta 10.....	65
Figura 27: Resultados de las pruebas con teléfonos inteligentes Xiaomi 4A.....	67
Figura 28: Resultados de las pruebas con teléfonos inteligentes Xiaomi 4A ^a	68
Figura 29: Resultados de las pruebas con teléfonos inteligentes Samsung Galaxy S5.....	69
Figura 30: Resultados de las pruebas con teléfonos inteligentes Samsung Galaxy S5.....	69
Figura 31: Pantalla principal de la aplicación web.....	70
Figura 32: Pantalla de inicio de la aplicación web.....	71
Figura 33: Formulario para crear choferes.....	72
Figura 34: Consular y descargar lista de pasajeros.....	72
Figura 35: Pantallas 1 a la 4 de la aplicación móvil.....	73

Figura 36: Escaneo de la cédula.....74

Resumen

El presente proyecto tiene como propósito la automatización del registro de los pasajeros en buses interprovinciales, caso Quito-Atacames, para lo cual se desarrolló una aplicación web y una aplicación móvil, dichas aplicaciones trabajan de forma conjunta para satisfacer la necesidad de automatizar el registro de pasajeros, haciendo uso del código de barras de la cédula de identidad.

El documento está organizado de la siguiente manera: Marco metodológico, Marco teórico, Análisis y diseño, Construcción y pruebas, Conclusiones y recomendaciones.

En el Marco metodológico se especifica la manera como se llevó a cabo el proyecto, haciendo uso de la metodología de gestión de desarrollo de proyectos de software XP, la forma como fue aplicada y las adaptaciones que fueron realizadas para la implementación del proyecto.

En el Marco teórico se detallan los principales conceptos que sustentan la aplicación de la metodología, para sustentar su aplicación se presentan los productos entregables que recomienda la misma.

Palabras clave: aplicación web, aplicación móvil, administrador de la cooperativa, lista de pasajeros.

Abstract

The purpose of this project is to automate the registration of passengers on interprovincial buses, in the case of Quito-Atacames, for which a web application and a mobile application were developed, these applications work together to meet the need to automate the registration of passengers, making use of the barcode of the identity card.

The document is organized as follows: Methodological framework, Theoretical framework, Analysis and design, Construction and testing, Conclusions and recommendations.

In the theoretical framework, the main concepts that support the application of the methodology are detailed, in order to support its application, the deliverable products recommended by it are presented.

Keywords: web application, mobile application, the cooperative administrator, passenger list.

INTRODUCCIÓN

Antecedentes

Actualmente los usuarios cuando hacen uso de los transportes interprovinciales para despasarse de una ciudad a otra, la forma con la cual se lleva un registro de los pasajeros que hacen uso de las diferentes unidades, se lo lleva acabo de forma manual, solicitando a los pasajeros que se registren llenando una hoja con sus datos.

Problema

Debido que actualmente el registro de los datos de los usuarios que hacen uso de los servicios interprovinciales se realiza de una forma manual, por parte del transportista, el controlador entrega una hoja donde cada pasajero tiene que llenar sus datos personales, en el mejor de los casos, y peor aun cuando los pasajeros se suben en el transcurso del viaje y no hay un registro.

Algunos transportistas no realizan ningún tipo de registro o se pierden la hoja y no existe un respaldo legal en un accidente de tránsito. En el momento de suscitarse un accidente de tránsito los cuerpos de socorro no tienen oportunamente un listado para identificar a los usuarios.

Justificación

El registro de los usuarios se realiza de forma manual, por parte del transportista, haciendo uso de una hoja y esfero, usando la tecnología existente se ha desarrollado una aplicación web y otra móvil, las aplicaciones ahorrarán tiempo y recursos a los beneficiarios y clientes al momento de registrarse.

En algunos casos los transportistas no registran ningún pasajero o también se les pierde la hoja de registro, mediante la aplicación web la lista de pasajeros estará disponible 24/7 y los beneficiarios tendrán la lista actualizada, y se evitará que al momento del viaje los pasajeros

se incomoden al tener que llenar el registro manualmente, también se podrá evitar que dicha lista sea manipulada de una manera irresponsable por terceros y que se pierdan.

En el transcurso del viaje suben pasajeros y son registrados con la aplicación móvil, se podrá realizar el registro de los pasajeros haciendo uso del código de barras de la cédula o su número de cédula en caso de no tenerla, al momento de la consulta la aplicación móvil devolverá los datos de los pasajeros y se registrará en la lista.

Para solucionar los problemas que se evidencian anteriormente se desarrollaron una aplicación web y una aplicación móvil que permite automatizar el proceso, haciendo uso de servidores en la nube, que ofrece una disponibilidad 24/7.

Objetivo General

Desarrollar una solución informática conformada por una aplicación web y una Android, que automatice el registro de pasajeros en buses interprovinciales, caso Quito-Atacames

Objetivos Específicos

Desarrollar una aplicación web que permita almacenar y administrar los datos de los pasajeros que hacen uso de los transportes interprovinciales permitiendo que las partes interesadas puedan acceder a los datos de forma más dinámica.

Desarrollar una App que permita verificar los datos del pasajero mediante el código de barras de la cédula y un módulo que permita ingresar el número de cédula en caso de no tenerla, al momento de la consulta la App devolverá los datos de los pasajeros y se registrará en la lista.

Generar reportes relacionados con el proceso de registro de los pasajeros en la ruta Quito-Atacames que estarán disponibles para los beneficiarios y para el público interesado.

Marco Metodológico

El proyecto fue llevado a cabo utilizando la metodología XP, esta decisión se tomó luego de un análisis en el que se encontró que ésta es la metodología que se adapta a las características de este proyecto. Esta metodología permitió realizar un desarrollo incremental e iterativo, el cual facilitó la construcción de la aplicación informática, de modo que se puedan cumplir, de forma adecuada, los objetivos del proyecto.

Aplicación de XP

Para la utilización de la metodología y facilitar la transferencia de conocimiento de cómo llevar a efecto su proceso de negocio, en las reuniones iniciales, se planteó la creación de un diagrama de procesos que especifica a detalle el registro de pasajeros, dicho diagrama evidencia cómo se llevan a efecto las tareas que permiten el registro de pasajeros con sus respectivos datos. Con el levantamiento del proceso se hace evidente la aplicación de la normativa relacionada con la RESOLUCIÓN N° 161-DIR-2013-ANT, que ayuda a entender de mejor manera dicho proceso.

Este proceso facilita la aplicación del modelo disminuyendo la frecuencia de reuniones con el cliente, debido a que los desarrollos de las iteraciones están basadas en el proceso, más las historias de usuario provistas por el cliente. Evidentemente, la utilización del diagrama de procesos ha significado una adaptación al marco de trabajo propuesto por XP, sin que esto ocasione inconsistencias en la utilización de esta metodología.

Historias de usuario

Se presenta las historias de usuario y los ajustes que se realizaron en el desarrollo.

Tabla 1: Resumen historias de usuario.

# Semanas	Duración (días)	Historia de usuario	# Ajustes	observaciones
Semana 1	7	Registro datos chofer (aplicación web)	0	Se crea los formularios para los choferes
Semana 2	4		1	Ajustes de la información relacionada con los choferes
Semana 3	7	Registro datos buses (aplicación web)	0	Se crea los formularios para los choferes
Semana 4	3		1	Ajustes de la información relacionada con los buses
Semana 5	4	Asignación chofer (aplicación web)	0	Se crea botón para asignar chofer a un bus
Semana 5	3		1	Se válida que se pueda asignar un chofer a un bus
Semana 6	2	Información datos del chofer, cooperativa y bus (aplicación móvil)	0	Se carga la información del chofer, cooperativa y del bus
Semana 6	2		1	Se aumenta información de la cooperativa (Nombre de chofer, nombre cooperativo, ruc, marca, número unidad)

Semana 6	1		3	Se crean los botones, cargar pasajero y cerrar sesión
Semana 7	2	Asignación destino (aplicación móvil)	0	Se carga la información de la cooperativa, chofer y del bus
Semana 7	1		1	Se aumenta información de la cooperativa (Nombre cooperativa, nombre chofer, número unidad)
Semana 7	1		2	Se carga la ruta, la selección de horario
Semana 7	2		3	Se cambia la forma de representar los horarios
Semana 8	3		0	Se crea el formulario del pasajero para el registro
Semana 8	1	Formulario pasajero (aplicación móvil)	1	Se modifica la información que tenga solo (CI, nombres, apellidos)
Semana 8	1		2	se crean los botones (escanear cedula, confirmar carga del pasajero, pausar, cierre definitivo)
Semana 9	1	Escaneo código de barras de la cédula de identidad (aplicación móvil)	0	se crea la pantalla para escanear el código de barras de la cédula

Semana 9	6			se modifica que salga el autoenfoco automático
Semana 10	7		1	
Semana 11	2	Ingreso manual de cédula (aplicación móvil)	0	se ingresa la cédula y se consulta la información
Semana 11	3		1	se elimina el boto para consultar la cédula, y al ingresar el ultimo dígito se consulta automáticamente la cédula
Semana 12	2	lista de pasajeros (aplicación móvil)	0	crear los botones para confirmar pasajero y cierre de definitivo
Semana 12	4		1	Mensajes que indique que está por cerrar la lista
Semana 13	1	Diseño de la lista de pasajeros (aplicación web)	0	se crea la interfaz para generar la lista de pasajeros
Semana 13	6			se genera una pre visualización de la lista de pasajeros
Semana 14	5		1	
Semana 14	1		2	validación, mensaje de que no se tiene generada lista para ese día
Semana 15	5	Activar Viajes (aplicación web)	0	se genera la pantalla para activar viajes

Semana 16	3		1	que se seleccione la unidad y se carga automáticamente el chofer que está asignado
-----------	---	--	---	--

Nota: Esta tabla contiene todas las historias de usuario con sus respectivos ajustes

Utilización de notaciones

Se utiliza UML

- Los diagramas de caso de uso se usan para analizar y modelar partes funcionales del software, se utilizó los diagramas de caso de uso para dar a conocer la funcionalidad de la aplicación web y la aplicación móvil con sus respectivos roles.

Se utiliza BPMN

- BPMN se utiliza para modelar procesos en un flujo de trabajo entendible, se usó para diagramar el proceso de negocio actual, y el proceso que dará como resultado al ser automatizarlo

Se utilizó diagramas entidad relación

- El diagrama de entidad relación se usa para modelar base de datos y evidenciar sus relaciones, se utilizó el diagrama de entidad relación para diagramar la estructura de la base de datos.

Capítulo 1

1.1 Marco teórico

1.1.1 Automatización

La automatización en el ámbito profesional es conocida como la transformación del trabajo convencional a uno más práctico, hace 45 años atrás los trabajos se realizaban de forma manual, esta forma de realizar los distintos trabajos se remonta a los años 1760 en la revolución industrial, fue uno de los avances más significativos para la humanidad, su aporte ayudó mucho a masificar la producción, los trabajadores se realizaban de forma manual y su tiempo de acabado era elevado, al automatizar el trabajo que era realizado de forma manual se logró aumentar la producción y disminuir el tiempo de fabricación, esto se hace más visible a partir siglo XX y en adelante que se realizan descubrimientos claves como el internet poco a poco se ha ido evolucionado de manera satisfactoria para el beneficio de la humanidad hasta la actualidad que es la época del conocimiento donde la automatización ha tomado tanta fuerza prácticamente el trabajo de forma manual ha pasado a segundo plano, con la ayuda de novedosas aplicaciones creadas con software y hardware la automatización está en su mejor momento para tener una evolución aún más significativa. («ANTECEDENTES DE LA AUTOMATIZACION», s. f.; Juan, 1993)

1.1.2 Transporte público interprovincial de pasajeros

El objetivo de la normativa es regular la prestación del servicio de transporte público interprovincial de pasajeros; así como también puntualizar el lugar de operación y circulación de las unidades que pertenezcan a las distintas operadoras autorizadas. (BONE VELÀQUEZ, CASTILLO AMPUDIA, & SANDOVAL EGAS, 2015, p. 4)

1.1.3 Manifiesto de pasajeros del servicio de transportes interprovinciales

La normativa vigente para regular la seguridad de los pasajeros se encuentra en la RESOLUCIÓN No 161-DIR-2013-ANT que reposan en la Agencia Nacional de Tránsito en el capítulo IV artículo 37, se estipula de la siguiente manera.

El manifiesto de pasajeros es el documento elaborado por el transportista que contiene la relación de pasajeros transportados por viaje, debiendo consignar, como mínimo el nombre de la operadora, placa del vehículo, nombre del conductor o conductores, origen y destino del viaje, hora de salida, nombres completos y cédula de identidad de cada pasajero, se elabora en original y dos copias, debiendo quedar el original en las oficinas de la operadora donde inicia el viaje, una copia para ser remitida a la Agencia Nacional de Tránsito y la otra copia para ser portada en el vehículo.(Agencia Nacional de Tránsito, 2013)

1.1.4 Metodología XP

Las metodologías ágiles en la actualidad son la forma más eficiente de desarrollar software, ya que tienen establecido un proceso muy eficaz para el desarrollo dentro de todas las metodologías ágiles para el desarrollo del proyecto técnico se escogió XP (programación extrema). XP usa el modelo incremental e iterativo que permite desarrollar el software mediante las iteraciones que sean necesarias, en cada iteración se repite el proceso (análisis, diseño, implementación, pruebas), al tener como resultado una parte funcional, después de ser presentado al cliente se recibe retroalimentación para poder ir afinando las aplicaciones, al mismo tiempo poder ir guiando al cliente para llegar al requerimiento real que satisfaga las necesidades del cliente.(«Extreme Programming: A Gentle Introduction.», s. f.) (BORJA LÓPEZ, s. f.)

1.1.4.1 Modelo Incremental

El modelo incremental está enfocado en desarrollar pequeñas partes funcionales que cumplan con los requerimientos haciendo uso del modelo (análisis, diseño, implementación,

pruebas), posteriormente, en las siguientes versiones se realizan incrementos funcionales, para cada incremento está definido su ciclo de vida y está basado en el anterior, sin afectar ni modificar su funcionalidad, al ser aprobado el incremento ya no se realizan cambios, solo se realizan las correcciones de errores si lo amerita.(José Joskowicz, 2008)

1.1.4.2 Comunicación

Uno de los valores más significativos de XP es la comunicación cara a cara, donde el cliente se comunica directamente con el jefe de proyecto en reuniones cortas, se realiza el levantamiento de requerimientos con la herramienta definida (historia de usuario), esta es la mejor manera para poder comunicarse y tener un mejor entendimiento de los requerimientos, al manejar de esta forma la comunicación es más fácil que el equipo de desarrollo pueda hacer los cambios necesarios.(José Joskowicz, 2008)

1.1.4.3 Simplicidad

El valor de la simplicidad que ofrece XP, ayuda que el equipo de desarrollado descubra soluciones simples y acertadas, según las necesidades del cliente, se implementa el diseño más simplemente que funcione, no se tiene que adelantar las funcionalidades que no estén dentro iteración que se está trabajando actualmente.(José Joskowicz, 2008)

1.1.4.4 Proceso XP

Los proyectos XP son exitosos cuando el cliente escoge el trabajo que genera el mayor valor para el negocio y decide automatizar e implementar, teniendo en cuenta las habilidades del equipo para medir la funcionalidad que genera en un tiempo dado. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos.(BORJA LOPEZ, s. f, p. 4)

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.

- El programador construye ese valor de negocio. Vuelve al paso 1.

1.1.4.5 Fase I exploración

La fase de exploración permite establecer el alcance del proyecto, dentro de esta fase el cliente hace conocer los requerimientos iniciales aunque no los tenga muy claro mediante las historias de usuario, este tipo de requerimientos permiten realizar una estimación del tiempo que tomará desarrollar las aplicaciones, las estimaciones podrían variar según avance el desarrollo y se realicen análisis posteriores, esto no se puede demorar más de dos semanas.(BORJA LÓPEZ, s. f.)

1.1.4.5.1 Historias de usuario

Las historias de usuario son los requerimientos descritos por el cliente, son definidas en un lenguaje no técnico, son las partes funcionales que el cliente desea que la aplicación tenga, las historias de usuario deben comunicar al equipo de desarrollo una parte funcional para que el mismo pueda dar una estimación del tiempo que tomara la construcción del proyecto.(José Joskowicz, 2008, p. 10)

1.1.4.6 Fase II planificación

La fase de planificación eventualmente dura poco, en dicha fase se ponen de acuerdo el cliente, el jefe de proyectos, los gerentes se ponen de acuerdo en el orden en que se comenzará a implementar las historias de usuario, y se define los entregables, esta fase contempla algunas reuniones grupales para realizar la planificación, como resultado se obtiene un plan de entregas.(José Joskowicz, 2008)

1.1.4.7 Fase III Iteraciones

Esta fase es la primordial para el periodo de desarrollo de XP, se empiezan a implementar las partes funcionales de los requerimientos según está establecido en el plan de entregas, al

final se obtiene un entregable funcional que cumple con la historia de usuario.(BORJA LÓPEZ, s. f.)

1.1.4.8 Fase III de puesta en producción

En esta fase no se realizan más desarrollos, se lleva a cabo revisiones de rendimiento antes de que las aplicaciones sean presentadas al cliente, dentro de esta fase también se toma decisiones si se fueran a implementar nuevas características, esto se estima acorde a nuevos cambios potenciales. (José Joskowicz, 2008)

1.1.4.8.1 Pruebas unitarias.

Las pruebas unitarias es una de las partes indispensables de XP, todos los módulos tienen que tener partes funcionales deben ser sometidos a las pruebas antes de ser publicadas, dichas pruebas que se usan deben estar definidas acorde al requerimiento antes de realizar el código.(José Joskowicz, 2008)

1.1.4.8.2 Pruebas de aceptación.

Las pruebas de aceptación están basadas en las historias de usuarios, dichas pruebas deben ser implementadas en cada iteración enfocados en las partes funcionales. Los escenarios deben ser descritos por el cliente para tener la certeza que las historias de usuario fueron implementadas acorde a las especificaciones del usuario.(José Joskowicz, 2008,p.15)

1.1.4.8.3 Detección y correcciones de errores.

En la instancia que se detecta un error, dicho error o errores deben ser corregidos de inmediato, adicionalmente se tiene que documentar el tipo de error para que se puedan mitigar futuros errores similares, después de corregir el error o los errores se someten a nuevas pruebas. (José Joskowicz, 2008)

1.1.4.9 Adaptación de metodología XP

La metodología XP es una de la más usadas dentro de todas las metodologías ágiles, cuando el cliente no tiene claro que es lo que realmente quiere, esto está relacionado al tener una gran capacidad adaptación cuando surge algún imprevisto en la forma de trabajo que se está llevando al momento, se debe a que los requerimientos puedan cambiar radicalmente con el proyecto, esto puede suceder en cualquier momento durante el desarrollo del mismo, puede ir variando indistintamente, la adaptación permite que la forma del trabajo evolucione continuamente y no sean evidentes las complicaciones. Los creadores de XP ponen a disposición las herramientas para el desarrollo y a su vez nos indican que es una buena práctica adaptarse a los requerimientos que sufran cambios.(«Metodologías del Desarrollo de Software», 2016).

1.1.5 Programación Android

Para desarrollar la aplicación móvil se hace uso de la herramienta Android Studio, la cual permite el desarrollo en un entorno java que es bastante familiar y amigables.

Android studio es el entorno de desarrollo específico de Android, basado en IntelliJ IDEA (entorno de desarrollo conocido en la comunidad de desarrolladores Java). Aporta muchas novedades respecto a Eclipse.(BENBOURAHILA, 2015)

1.1.5.1 Componentes de una aplicación móvil

Cuando se desarrolla una aplicación en Android, se tiene que entender los elemento que componen la funcionalidad, uno de las partes más importantes que siempre se tiene que estar pendientes es su funcionamiento, cada aplicación usa su proceso propio y se lo puede distinguir por ID, este ID funciona como un identificador para que solo dicha ampliación tenga acceso a sus recursos (archivos),cada aplicación usa la pila de tareas, la cual ejecuta en secuencia los procesos en Andriod, están compuestas por actividades que se encolumnan cada vez que son llamadas, y de la misma forma terminas en secuencia una atrás de la otra, la

aplicación administra el tiempo de ejecución y si encuentra una activada que se está demorando mucho la termina, esto sucede cuando la aplicación necesita más memoria para ejecutar otras actividades.(BENBOURAHILA, 2015)

1.1.5.2 Actividades

La funcionalidad de la actividad es encargarse de presentar al usuario la interfaz gráfica, la actividad está representada por una ventana, está encargada de ser el canal de comunicación entre usuario y la ventana, cada interfaz debe tener una actividad por cada proyecto, los ficheros XML (*./res/layout*) tiene que contener todos los elementos que se presentan en cada actividad, de esta manera se los puede llamar en cada clase.

NameActivity.class se encuentra dentro de los ficheros XML que son heredados de la clase Activity, todos los elementos de la pantalla están definidos dentro de NameActivity , como por ejemplo los botones, texto, layouts, checklist, etc.(*android.pdf*, s. f.)

Servicios

Los servicios están relacionados directamente con las tareas, dichas tareas no son visibles y se ejecutan en segundo plano disponen de un hilo propio que permite ejecutar cualquier tarea, así sea liviana o pesada, no se requiere una interfaz a más que sea necesario, y de ser necesario la clase Service la exportaría.

El ciclo de vida de un servicio se inicia con el método onCreate(Bundle), y se libera con el método onDestroy().(*android.pdf*, s. f.)

1.1.5.3 Proceso y diseño de una aplicación móvil

El diseño y desarrollo de una aplicación tiene un proceso que seguir, empieza desde la idea prosigue con el análisis para llegar a su final con la publicación en app store y play store. En el desenvolvimiento de las diferentes etapas, programadores y diseñadores realizan un trabajo de forma conjunta, simultanea y coordinada.(BENBOURAHILA, 2015)

1.1.5.4 Conceptualización

Como resultado esta etapa es una breve idea de la aplicación móvil en funcionamiento, lo cual permite evidenciar las necesidades y problemas de los usuarios. Dicha idea hace referencia a una investigación previa y a la posterior validación de la viabilidad del concepto.(*android.pdf*, s. f.)

1.1.5.5 Definición

En la definición es el proceso para interactuar con el cliente y dar una descripción con algunos detalles de la aplicación, en la definición se comienza a definir ciertas funcionalidades, con lo cual se podrá estimar el alcance del proyecto y la complejidad y desafío que tendrá la programación y el diseño para la programación de la aplicación móvil.(*android.pdf*, s. f.)

- Definición de funcionalidades.
- Definición de usuarios.

1.1.5.6 Diseño

Para la etapa de diseño se pone en práctica las definiciones anteriores, como primera parte en forma de wireframes (es el boceto donde se representa el esquema visual de la página web), lo cual avala la creación de los primeros prototipos para ser presentado y probados con usuarios, en la segunda parte se desarrolla entorno visual que se lo entregara al programador, en forma de maquetas o modelos, para la programación del código.(*android.pdf*, s. f.)

- Wireframes.
- Prototipos.
- Test con usuarios.
- Diseño visual.

1.1.5.7 Desarrollo

El programador o programadores se encargan de crear la estructura que soportara el funcionamiento de la aplicación móvil. Al tener una versión inicial, se destina el tiempo necesario para hacer pruebas funcionales y de los resultados, si existen errores se los corrija de inmediato, esto se lo realiza con el fin de garantizar el correcto funcionamiento de la aplicación móvil. (*android.pdf*, s. f.)

- Programación del código.
- Corrección de errores.

1.1.5.8 Publicación

La aplicación finalmente se publicada y presentada a los usuarios, después de ser liberada, se realiza un seguimiento para poder evaluar el desempeño y comportamiento en un entorno real, este seguimiento ofrece información importante que sirve para corregir errores que se pueden presentar o eventualmente generar actualización de la aplicación móvil. (BENBOURAHLA, 2015)

1.1.6 Programación Web

La programación web en los diferentes años ha evolucionado hasta que en la actualidad se puede optimizar el uso de recursos para la programación, actualmente se tiene distintas formas y lenguajes para programar, el uso de framework y librerías hacen que la escritura se optimice la escritura del código y mediante clases se puedan programar partes funcionales y se pueda utilizar en varias partes para programar distintas partes funcionales sin la necesidad de escribir muchas líneas de código. (*APLICACIONES WEB*, s. f.)

En la programación de aplicaciones web se puede evidenciar 3 niveles, en el nivel superior todas las partes funcionales están destinados para el usuario, los usuarios están relacionados con el navegador web, el nivel inferior está relacionado con la base de datos, el nivel

intermedio está encargado de procesar los datos y se lo denomina servidor web.(*APLICACIONES WEB*, s. f.)

Una aplicación web está orientada y relacionada con la arquitectura cliente servidor, el cliente se lo define como el explorador o visualizador1, el servidor web se comunica mediante el protocolo (HTTP).

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son los empleados en Internet. Estos protocolos permiten la conexión de las aplicaciones heterogéneas, lo que facilita el intercambio de información entre distintos ordenadores. (*APLICACIONES WEB*, s. f,p. 48)

1.1.6.1 El cliente

El cliente web es el que interactúa con el usuario, realizar peticiones al servidor, el servidor se encarga de enviar los recursos solicitados mediante HTTP2.

La parte del cliente está formada por el código HTML, que forma la página web más algo de código ejecutable realizado en lenguaje de Script del navegador (JavaScript o VBScript).(*APLICACIONES WEB*, s. f.)

Las tecnologías más usadas para el cliente web son:

- HTML.
- PHP
- CSS.
- Lenguajes de script: JavaScript, VBScript, etc.

1.1.6.2 PHP

El funcionamiento básico de PHP es interpretado de la siguiente manera, Dentro del servidor el código PHP realiza la transformación en código HTML, en CSS y texto.

El documento así interpretado es enviado al cliente, y este lo interpreta como si fuera un documento HTML ordinario. (Arias, 2017)

PHP tiene algunas instrucciones

- Recepción de datos (formularios).
- Manipulación de datos (cálculos).
- Decisiones de datos
- Resultados de los valores del código de HTML.
- Acciones de creación, lectura, escritura, eliminación, de ficheros dentro del servidor.

1.1.6.3 CSS

CSS permite crear, diseñar y personalizar el estilo de las páginas web, es muy usado para dar una estructura ordena y bien definida que permite tener un estilo visual, para diseñar interfaces de usuario escritas en PHP, HTML.

CSS maneja una sintaxis simple que permite usar palabras claves que son propias de CSS las cuales están definidas en el lenguaje natal de la programación (ingles), que permite definir los nombres de las variables a usar en el diseño del CSS. CSS hace uso de estilos que son definidos por estándares de programación.(Tinoco & Solís, 2014)

- Font family
- Font Style
- Font Variant
- Font weight
- Font size

Los componentes que conforman el CSS son los siguientes. (Tinoco & Solís, 2014)

- Fuentes de texto
- Colores de fondo
- Elementos Sapan y Div

1.1.6.4 El Servidor

El servidor web es una aplicación que se encuentra en espera de que surjan peticiones de conexión mediante el protocolo HTTP, que son solicitadas por parte del cliente web. En los sistemas Linux son denominados demonios, a diferencia de los sistemas de Windows se los denomina servicio.(*APLICACIONES WEB*, s. f.)

Las partes del servidor web es conformados por.

- Páginas estáticas (documentos HTML) que siempre muestran el mismo contenido.(*APLICACIONES WEB*, s. f.)
- Recursos adicionales (multimedia, documentos adicionales, etc.) que se pueden emplear dentro de las páginas o estar disponibles para ser descargados y ejecutados (visualizados) en el cliente.(*APLICACIONES WEB*, s. f.)
- Scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas página, la salida de este Script suele ser una página HTML estándar que se envía al navegador del cliente.(*APLICACIONES WEB*, s. f.)

1.1.6.5 Ejecución de páginas web

Para realizar el proceso completo de ejecución de la página web, se inicia cuando el usuario realiza la petición a la página web, y finaliza cuando el navegador se la muestra con el contenido adecuado.(Arias, 2017)

- El usuario ingresa la dirección (URL) en el navegador de la página que desea visitar.
- Mediante el cliente se realiza la conexión con el servidor web.
- El cliente realiza la petición de la página web o uno de sus componentes deseados.
- El servidor se encarga de responder enviando la página o su componente deseado, de no ser acertado se genera un error, en su mayoría de veces se los envía representado por el código 400 objeto.

- Después de cumplir con la petición la conexión que se mantiene con el cliente y el servidor se termina.
- Se visualiza la página que fue solicitada por el usuario.(Arias, 2017)

1.1.7 Arquitectura MVC

El modelo MVC es una estilo de arquitectura de software que se encarga de separar la aplicación en tres partes, el modelo, la vista, el controlador, el funcionamiento del MVC se inicia cuando el usuario realiza una petición a la página web (navegador), posteriormente el controlador se encarga de responder a la petición, el controlador es el controla la lógica de la aplicación web, después que el controlador evidencia que el usuario ha realizado una petición, el controlador solicita al modelo los recursos necesarios, el modelo realiza la búsqueda de los recursos en la base de datos (videos, texto, imágenes, etc.), después de que el modelo encuentra los recursos solicitados los envía a la vista, la vista se encarga de organizar los recursos, aplica los estilos, las vista enseña los resultados solicitados en el navegador.(Pantoja, 2004a)

1.1.7.1 Modelo

El modelo accede a la capa de almacenamiento de la base de datos, está encargada de realizar las consultas a la base de datos, las actualizaciones, las eliminaciones, las inserciones.

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar.(Pantoja, 2004a)

1.1.7.2 Vista

La vista está relacionada directamente con la representación visual de los componentes de la aplicación web, es el canal entre la interfaz gráfica y el usuario.

Las vistas están representadas por el conjunto de clases, lo cual da a conocer los resultados de la información que contiene el modelo y es presentada mediante la interfaz gráfica al usuario.(Pantoja, 2004b)

1.1.7.3 Controlador

El controlador es el canal para la interconexión entre las peticiones del modelo y las representaciones visuales de la vista, el controlador recibe las peticiones del usuario, envía dichas peticiones de datos al modelo, posteriormente se las envía a la vista, el controlador se encarga de realizar las modificaciones necesarias al modelo si fuera necesario o se encarga de abrir o cerrar las vistas si el caso lo ameritara.(Pantoja, 2004a)

1.1.8 Laravel

Laravel es el framework (es la estructura del software que está compuesta de componentes que se pueden personalizar) orientado a PHP, está enfocado en un conjunto de clases que permite desarrollar una aplicación web, ahorrando líneas de código este modelo de programación hace el desarrollo sea más eficiente y fácil.(«Laravel - EcuRed», s. f.)

Cuando se programa de forma libre en PHP se generan muchas líneas de código y cada vez que se requiera hacer una nueva funcionalidad esto requiere repetir la líneas de código lo que genera muchas más líneas de código, al tener ya algunas líneas de código y se requiera una nueva funcionalidad, al hacer uso de Laravel ya no tiene que repetirlo, se llama a la clase que se requiera y la funcionalidad queda desarrollada de una forma más eficiente, Laravel tiene características importantes.(«Laravel - EcuRed», s. f.)

- Conjunto de librerías.
- Fácil enrutamiento.
- Autenticación simple.
- Motor de plantillas.

- Eloquent ORM.
- Soporte para MVC.
- Apirest.
- Incorporación para los estándares de programación PSR-2 y PSR-4.(«Laravel - EcuRed», s. f.)

1.1.8.1 Funcionamiento de framework laravel

Al momento de que un usuario realiza una petición al API laravel lo interpreta a través de HTTP, en primera instancia el servidor recibe la petición que es filtrado por el motor de PHP, consecuentemente laravel inicializa las rutinas.(«Laravel - EcuRed», s. f.)

1.1.8.2 Eloquent ORM

Eloquent ORM es una de las características funcionales más representativas ya que permite manejar la base de datos de forma sencilla y eficaz, con la gama de funciones que mantiene permite hacer complejas peticiones a la base de datos al momento que se requiera sin rescribir líneas de código. (Chen, Ji, Fan, & Zhan, 2017)

Cada tabla de base de datos tiene un modelo correspondiente que se utiliza para interactuar con esa tabla, los modelos permiten consultar los datos de las tablas, así como insertar nuevos registros en la tabla.(Chen et al., 2017)

1.1.8.3 Apirest

El Apirest es un conjunto de reglas que mantiene la comunicación entre las aplicaciones, las operaciones más importantes que permite manipular los recursos son las siguientes.

- Get: permite realizar las consultas
- Post: permite crear
- Put: para realizar las ediciones
- Delete: para realizar la eliminación

Los objetos en REST manipulados con URI es el identificador único de cada recurso de un sistema REST. Esto, facilita el acceso a la información, para poder modificarla o borrarla. También para compartir su ubicación exacta a terceros. («Qué es Api Rest y por qué debes de integrarla en tu negocio ¡Descúbrelo!», 2018)

1.1.9 MySQL

MySQL es un administrador de base de datos, es adaptable a la arquitectura cliente servidor, es amigable y fácil de usar y a su vez es robusto, una de sus ventajas es que es compatible con los lenguajes de programación, lo que lo vuelve uno de los administradores de base de datos más usados que es de fuente abierta. («¿Qué son las bases de datos?», 2007)

1.1.9.1 Base de datos

Una base de datos es un repositorio donde se puede guardar grandes cantidades de información de una forma muy segura y organizada, esto permite al momento que es deseado encontrar alguna información se la pueda encontrar de una forma rápida y precisa, las bases de datos son más usadas para el desarrollo de aplicaciones móviles que necesiten manejar cierta cantidad de información, las características de las bases de datos son las siguientes. («¿Qué son las bases de datos?», 2007)

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar. («¿Qué son las bases de datos?», 2007)

1.1.10 PSR-2

PSR-2 son los estándares de programación que permiten que la escritura del código en PHP sea de una forma universal que permite que los demás desarrolladores puedan entender el código al momento de diseñar nuevas funcionalidades, las características más relevantes para el uso del estándar de PSR-2 son las siguientes. («PSR-2: Coding Style Guide - PHP-FIG», s. f.)

- Se tiene que usar 4 espacios para la indentación, no se debe usar tabuladores
- No existir un límite para la longitud de las líneas, pero lo recomendable es que midan menos de ochenta columnas, si las columnas sobrepasan las 120 columnas se considera como un error.
- Al momento de abrir los llaves se debe colocar en la siguiente línea para la declaración de un método, clase, o de una función, y para cerrar tiene que estar en la misma línea que el cuerpo.
- Para la estructura de palabras compuestas se hace uso de los estándares de programación camel case que permite realizar la unión de las minúsculas con las mayúsculas por ejemplo (getBuses),
- Para la definición de las variables se hace uso del estándar de programación Snake case el cual permite realizar la unión con el guion bajo (_) por ejemplo (pasajero_registrado), este estándar es una buena práctica usado para programación en PHP.
- No se tiene que tener más de una sentencia por línea. («PSR-2: Coding Style Guide - PHP-FIG», s. f.)

Capítulo 2

2.1 Análisis

2.1.1 Proceso de registro manual de pasajeros

En las reuniones mantenidas con el Gerente de la cooperativa se define las historias de usuario el cual permite evidenciar el problema y poder definir la solución, en primera instancia se realiza el levantamiento del proceso manual, con el cual se está manejando el registro de los pasajeros, actualmente el registro de los pasajeros se lo realiza manualmente. El modelo del proceso permite evidenciar como se realiza el registro de pasajeros y verificar aquellos que son los participantes que están inmersos en el registro de los pasajeros y la información necesaria para el análisis y construcción.

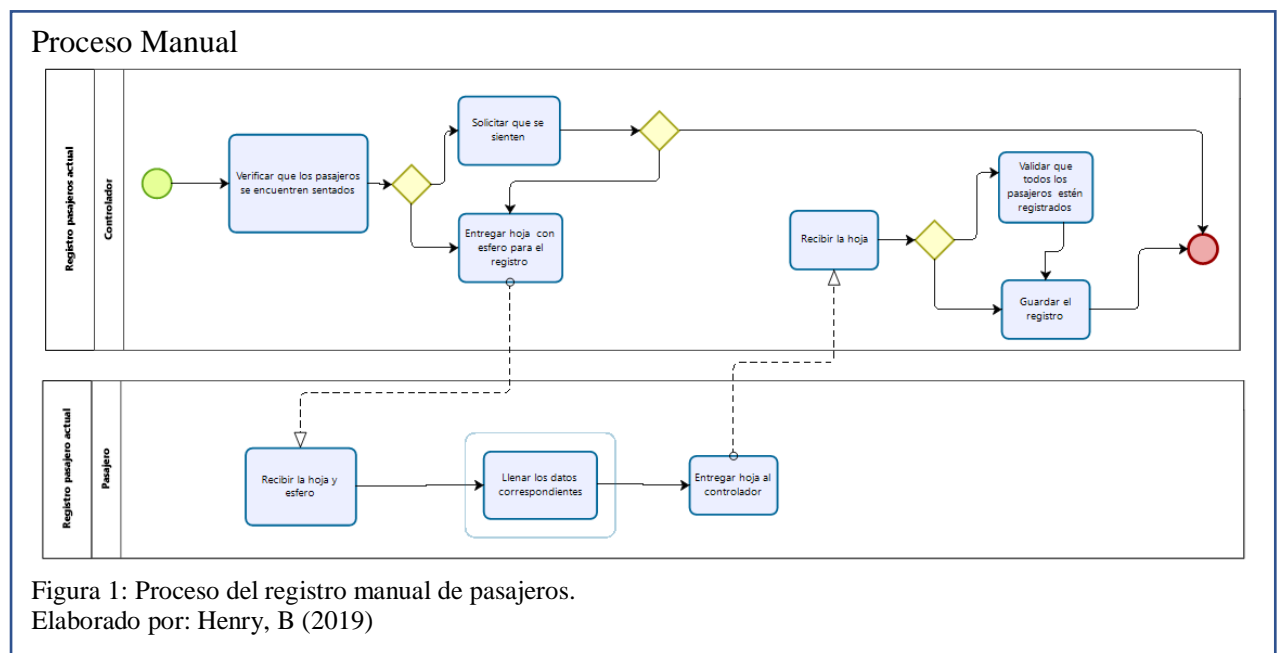


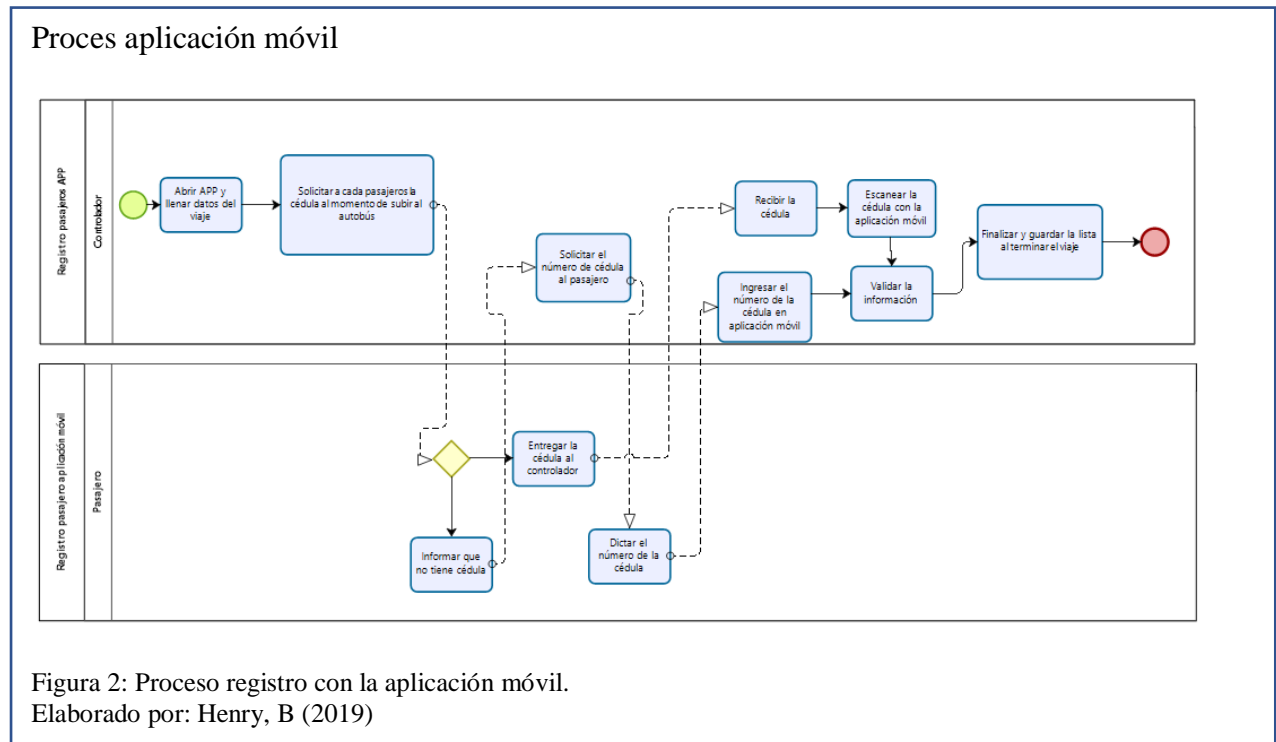
Figura 1: Proceso del registro manual de pasajeros.
Elaborado por: Henry, B (2019)

El proceso inicia cuando el controlador verifica que todos los pasajeros estén sentados y posteriormente pasa una hoja con un esfero, los pasajeros llenan la hoja con datos correspondientes y al finalizar el registro entregan al controlado, el controlador verifica que todo se encuentre correcto y guarda la hoja.

En la figura No. 1 se observa el proceso que realiza el controlador y los pasajeros para el registro de los mismos, al realizar el análisis del flujo pertinente se presentaron algunos

cambios al ser un proceso de registro manual la información es vulnerable el mismo proceso es ineficiente ya que algunas ocasiones no se realiza el registro, con este proceso manual no se puede garantizar ninguna acción oportuna al surgir algún problema en el transcurso del viaje, con las modificaciones al proceso se tiene:

2.1.2 Proceso registro con la aplicación móvil

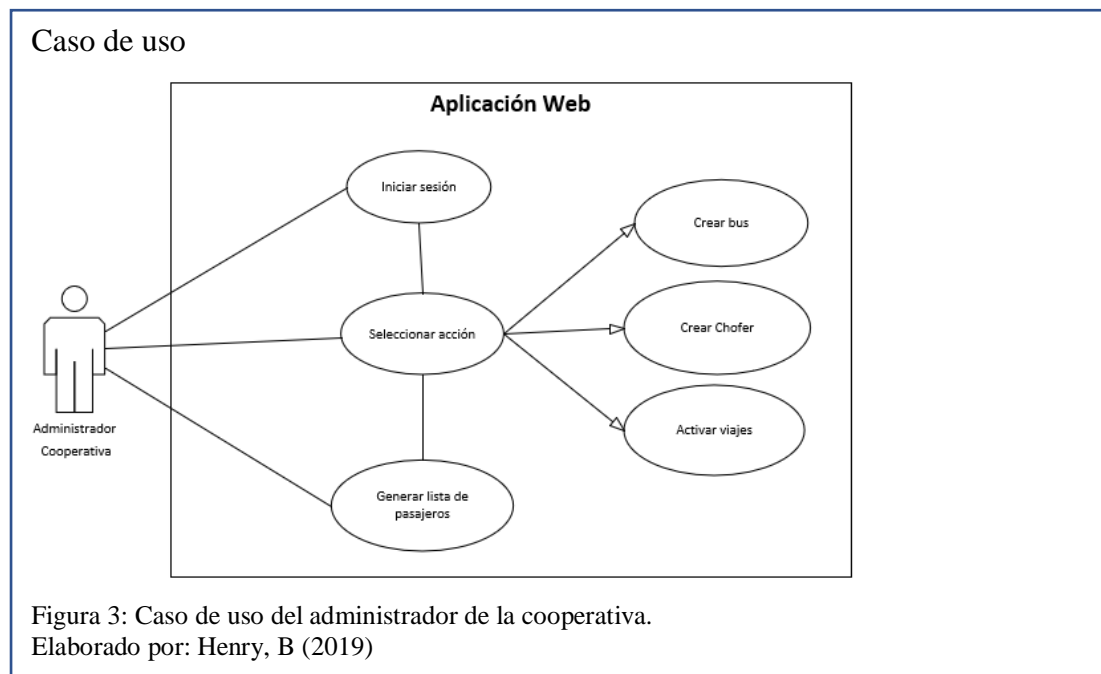


El proceso optimizado ofrece una estructura funcional y adecuada para garantizar la información, el proceso inicia al abrir la aplicación móvil y el controlador realiza la selección de la hora del viaje, solicita la cédula al pasajero si el pasajero tiene la cédula el proceso solo se enfoca en el uso de la aplicación móvil, posteriormente se escanea el código de barras, se verifica los datos se guarda el registro del pasajero, si el pasajero no tiene la cédula se solicita el número y se procede con el registro, se verifica los datos se guarda el registro del pasajero, se optimizo el tiempo de registro y se garantiza que la información es fiable y oportuna

2.2 Casos de uso

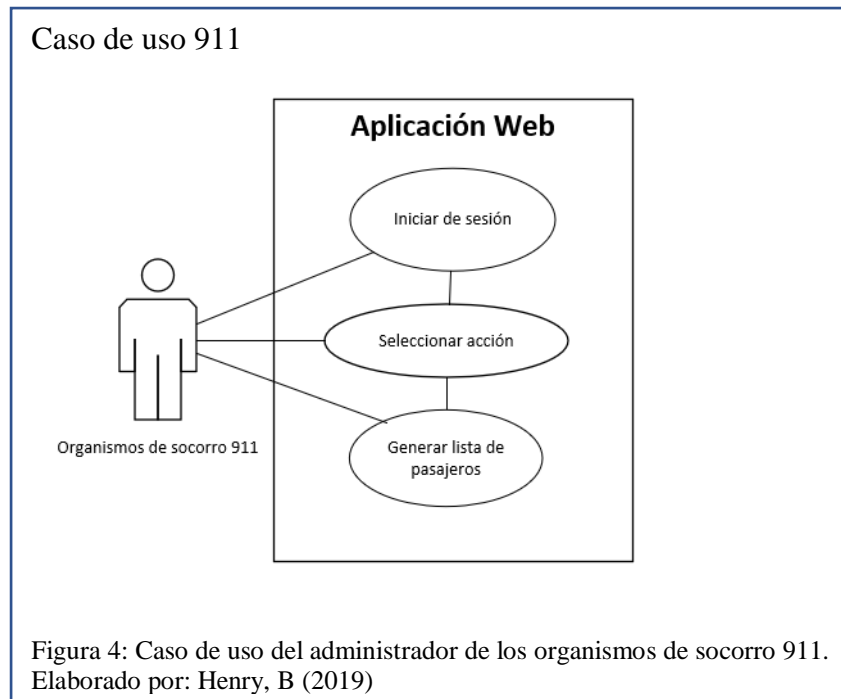
Para poder optimizar el proceso de análisis del proyecto, los diagramas de casos de uso permitieron tener una idea más clara de los componentes, los roles y características de cada usuario que fue generado para su perfil, lo cual son usados para la aplicación web y la aplicación móvil, después de haber levantado todas las historias de usuario y tener definidos los procesos se torna fácil entender los roles de cada usuario y para su mayor entendimiento se realizan los casos de uso que a continuación son detallados.

2.2.1 Caso de uso Administrador de Cooperativa



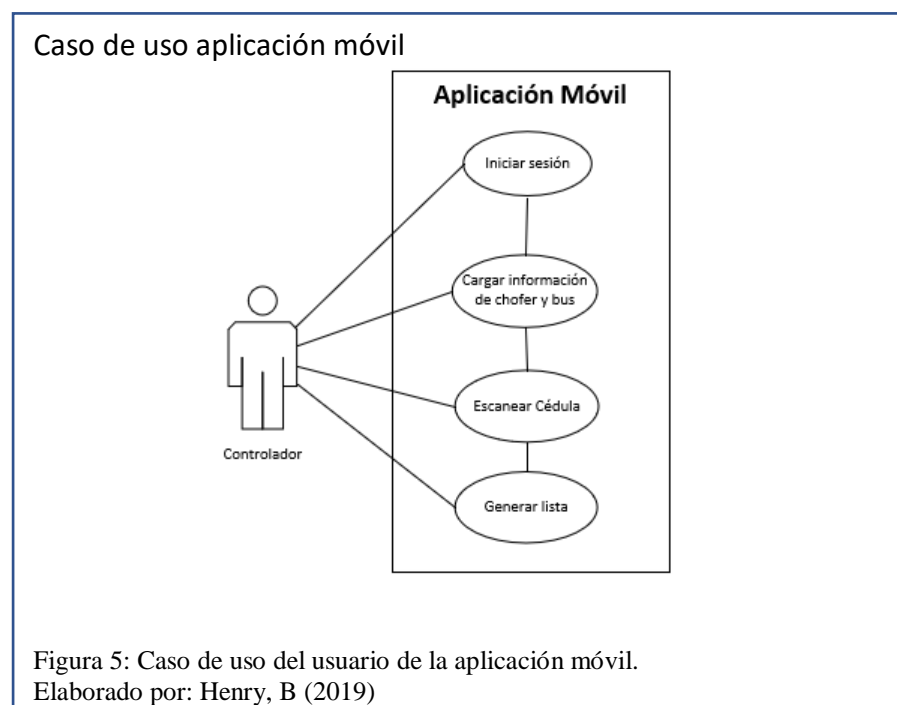
En la figura No. 3 se demuestra el perfil que tiene el administrador de la cooperativa, está diseñado con el inicio de sesión para ingresar a la aplicación web, está contemplado por el menú que podrá usar administración con el submenú crear chofer crear bus, activar viajes, generar lista de pasajeros.

2.2.2 Caso de uso organismos de socorro 911



En la figura No. 4 se demuestra el perfil que tiene los organismos de socorro, está diseñado con el inicio de cesión para ingresar a la aplicación web, tienen el menú, generar lista de pasajeros.

2.2.3 Caso de uso para usuario de aplicación móvil



En la figura No. 5 se demuestra el perfil que tiene el controlador, está diseñado con el inicio de cesión para ingresar a la aplicación móvil, cargar pasajeros, seleccionar horarios, escanear cédula, confirmar pasajero, pausar, cierre definitivo.

2.3 Resumen de las tablas de la base de datos.

Se define el diagrama de la base de datos para garantizar el funcionamiento correcto y el uso de la información que se manejará entre la aplicación web y la aplicación móvil, ya que la aplicación móvil consume información de la aplicación web y viceversa.

Tabla 2: Componentes de la base de datos.

Tablas	Descripción	Campo
Buses	Contiene el registro de todos los buses, sin distinción de la cooperativa a la cual pertenecen. Es un maestro de Buses.	-Marca -Modelo -Año -Placa -Número de Chasis -Número de Unidad
choferes_cooperativa	Es una tabla relacional, la cual permite asociar un chofer con un bus y la cooperativa a la cual pertenece	-Id Usuario -Id Bus -Id Cooperativa
Cooperativas	Contiene el registro de todas las cooperativas.	-Nombre -RUC
cooperativa_rutas	Es una tabla relacional, la cual permite asignarle un horario de salida a determinada ruta de una cooperativa. Esta tabla contiene la información de todas las rutas de una cooperativa para determinada hora. De aquí se toman los horarios que estarán disponibles para la aplicación móvil (siempre y cuando la ruta este activa para el día de salida del bus)	-Id Cooperativa -Id Ruta -Hora Salida (HH:MM)
cooperativa_rutas_activa	Es una tabla de control, la cual facilita la activación de rutas por cooperativa. Es una de las tablas más importantes del proceso, puesto que mantiene el control de carga de pasajeros; es decir, es quien decide si una ruta estará activa para un bus en determinado día y hora. En esta tabla también se almacena la fecha y hora de activación, así como también quien realizo la activación de la ruta.	-Id Chofer -Id Bus -Id Cooperativa -Id Ruta -Hora Salida -Fecha Salida -Id Admin Cooperativa

licencias	Contiene el registro de todas las licencias de conducir disponibles para los choferes al momento de su registro	-Código -Descripción
migrations	Esta no es una tabla propia de la aplicación, pero si del framework de Laravel, sobre el cual se desarrolló la aplicación Web. Contiene simplemente el nombre de la migración ejecutada por Laravel para montar cada tabla/proceso de base de datos	
pasajeros	Contiene el registro de todos los pasajeros que han sido escaneados por la aplicación móvil. Esta tabla permite hacer un uso más eficiente de los datos; es decir, que, por cada escaneo de código de barra al momento de cargar al pasajero, primero se verifica que el pasajero no se encuentre en la base de datos. Si no se encuentra, entonces se consultan los datos a un webservices del registro civil de Ecuador.	-Cédula -Nombres -Apellidos -Fecha Nacimiento
pasajeros_buses	Es una tabla de control, la cual permite saber el itinerario final de un bus al momento de su carga. Esta tabla asocia una ruta activa con el chofer que realiza la carga del bus, la fecha de salida del bus y un identificador del pasajero, el cual se registra previamente en la aplicación. También posee un atributo de control, que permite mantener abierta o cerrada una lista de pasajeros.	-Id Ruta Activada por Administrador de Cooperativa. -Id Chofer -Id Pasajero -Fecha Salida -Control Lista
password_resets	Esta es una tabla del framework de Laravel para el control de reseteo de contraseña de un usuario	-Email -Token
rutas	Contiene el registro de todas las rutas disponibles en la aplicación.	-Origen -Destino
users	Contiene todos los datos relacionados con los usuarios de la aplicación (Super Usuario, Administradores de Cooperativa, Choferes y Usuarios del 911). Para el caso del Chofer, se almacena el tipo de licencia y la cooperativa a la cual pertenece. Para el caso del Administrador de la Cooperativa, se asocia la cooperativa que administrara. Para los usuarios del 911 y Super Admin, todos los demás datos requeridos por la aplicación.	-IdCooperativa (Opcional) -IdLicencia (Opcional) -Cédula -Nombres -Apellidos -Email -Password -Role -Fecha Nacimiento -Celular -Dirección Domicilio

Nota: Esta tabla contiene todas las tablas de la base de datos.

2.3.1 Diagrama entidad relación de la base de datos

Se define el diagrama de entidad relación de la base de datos para entender las relaciones y funcionamiento que lleva a cabo la aplicación móvil y la aplicación web, tanto en el uso de datos y consumo, el diagrama está diseñado de una forma simple y flexible, para garantiza que la aplicación web y la aplicación móvil interactúen de una forma eficiente, contiene tablas enfocadas en la funcionalidad de las aplicaciones.

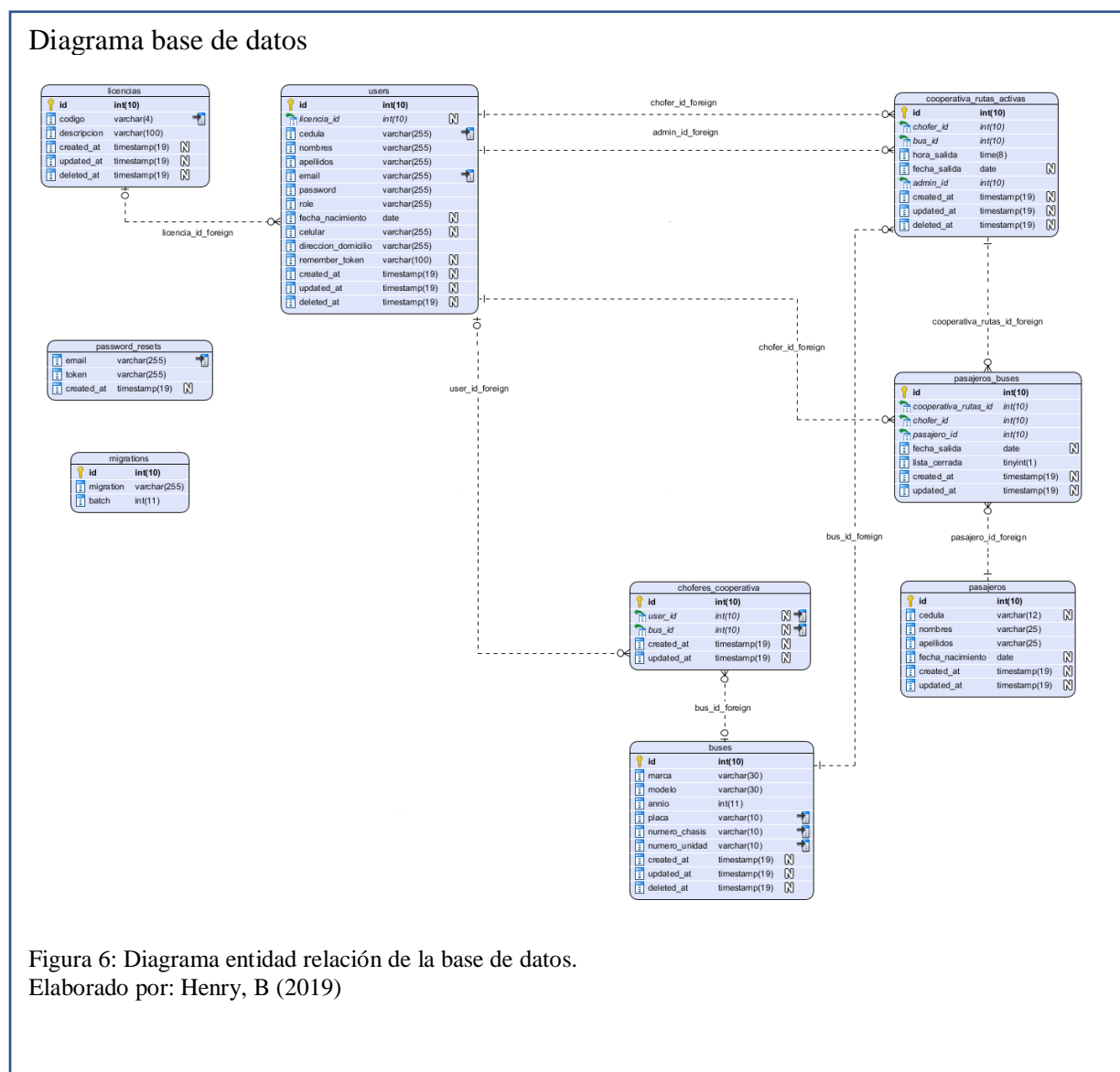


Figura 6: Diagrama entidad relación de la base de datos.
Elaborado por: Henry, B (2019)

Capítulo 3

3.1 Estándares de programación

Los estándares de programación se definieron para el nombre de las tablas, nombres de las variables, nombres de clases, nombres de métodos, el desarrollo de la aplicación web y aplicación móvil fueron realizados haciendo uso de las buenas prácticas de programación basados en PSR-2, PSR-2 permite estandarizar la codificación, garantizar que el desarrollo, permite evidenciar la correcta manera para que sea más fácil poder leer y entender por el código escrito por cualquiera desarrollador que desee evidenciar el trabajo realizado y pueda hacer uso como referencia para siguientes desarrollos.

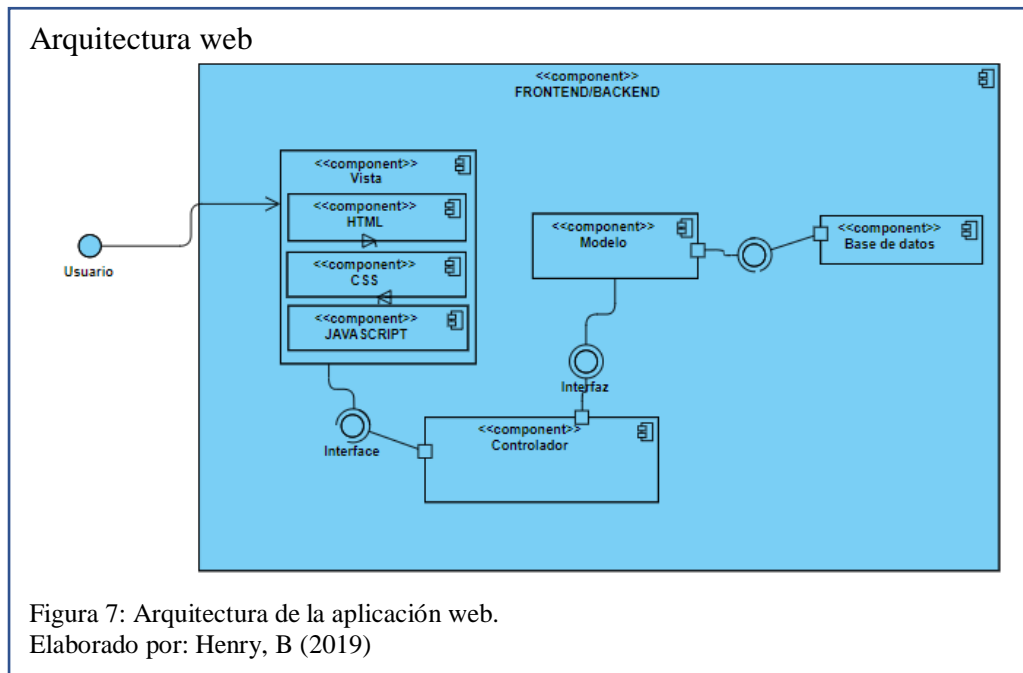
- Para la definición de las funciones que son palabras compuestas se usa los estándares de programación camel case que permite realizar la unión de las minúsculas con las mayúsculas por ejemplo (getBuses), este estándar es una buena practica usado para programación en PHP.
- Para la definición de las variables se hace uso del estándar de programación Snake case el cual permite realizar la unión con el guion bajo (_) por ejemplo (pasajero_registrado), este estándar es una buena practica usado para programación en PHP.
- Para la escritura del código se usa 4 espacios para sangrar
Para abrir y cerrar las llaves de las clases se usa de la siguiente manera, para abrir las clases van en la siguiente línea, y para cerrar van en la siguiente línea después del cuerpo.
- Para el uso de las palabras claves como (true, false, null) se las escribe en minúsculas.

- ¿Para la escritura del código se hace uso de etiquetas largas `<?php ?>` y para cortas `<?= ?>`.
- Todas las constantes están escritas en minúsculas.
- Para la creación de las clases se usan nombres propios a los cuales se representaron de la siguiente manera (rutas, pasajeros) y para combinaciones (pasajeros_buses, choferes_coperativas).
- Para la creación de los métodos se usan nombres propios a los cuales se representaron de la siguiente manera (login) y para combinaciones (guardarItinerario).
- Para los campos de las tablas se usan nombres propios a los cuales se representaron de la siguiente manera (nombres, apellidos, email) y para combinaciones (fecha_salida, lista_cerrada).

3.2 Arquitectura

3.2.1 Arquitectura aplicación web

Para la implementación de la aplicación web se ha utilizado el patrón arquitectónico MVC (modelo, vista, controlador), el desarrollo funcional está separado en modelos vistas y controladores, la arquitectura establecida permite brindar una disponibilidad de 24/7, la arquitectura MVC permite realizar cambios en partes específicas sin afectar a cualquier otro componente.



Se representa como se comunica el Front-End con el Back-End en la aplicación web, los cuales interactúan de la siguiente manera: El usuario realiza una petición (GET, POST, DELETE, PUT), la cual es recibida por el servidor mediante un archivo de rutas (web.PHP), el cual se encarga de direccionar a cada petición a un método perteneciente a un controlador.

El controlador procesa la petición, ejecutará los procesos solicitados, y en caso de alguna afectación en base de datos, este se comunica con la misma mediante un modelo de datos. Cada modelo de la aplicación es una representación de las tablas de base de datos MySQL de la aplicación. Desde el modelo se realizan las inserciones, actualizaciones, ediciones o eliminación de datos de la aplicación, para esto, Laravel hace uso de su ORM llamado Eloquent, el cual ayuda a manejar la base de datos mediante objetos, para esto, Eloquent mapea todos los datos de nuestra base de datos, evitando todo el trabajo de escribir las consultas para el acceso de la información. El acceso a sus relaciones también se realiza por medio de propiedades de objetos, lo que facilita mucho las dinámicas de acceso a la información, siempre que haya una relación de dependencia declarada entre las distintas entidades de nuestro modelo de datos.

Una vez que se ejecute la acción sobre la base de datos, el controlador valida la ejecución de la misma y devuelve a la vista un mensaje de éxito en formato JSON con código 200, o un error con código 400 indicando en donde se presentó el error.

El controlador también se encargará de devolver las vistas solicitadas por el usuario mediante el mismo archivo de rutas.

3.2.2 Arquitectura aplicación móvil

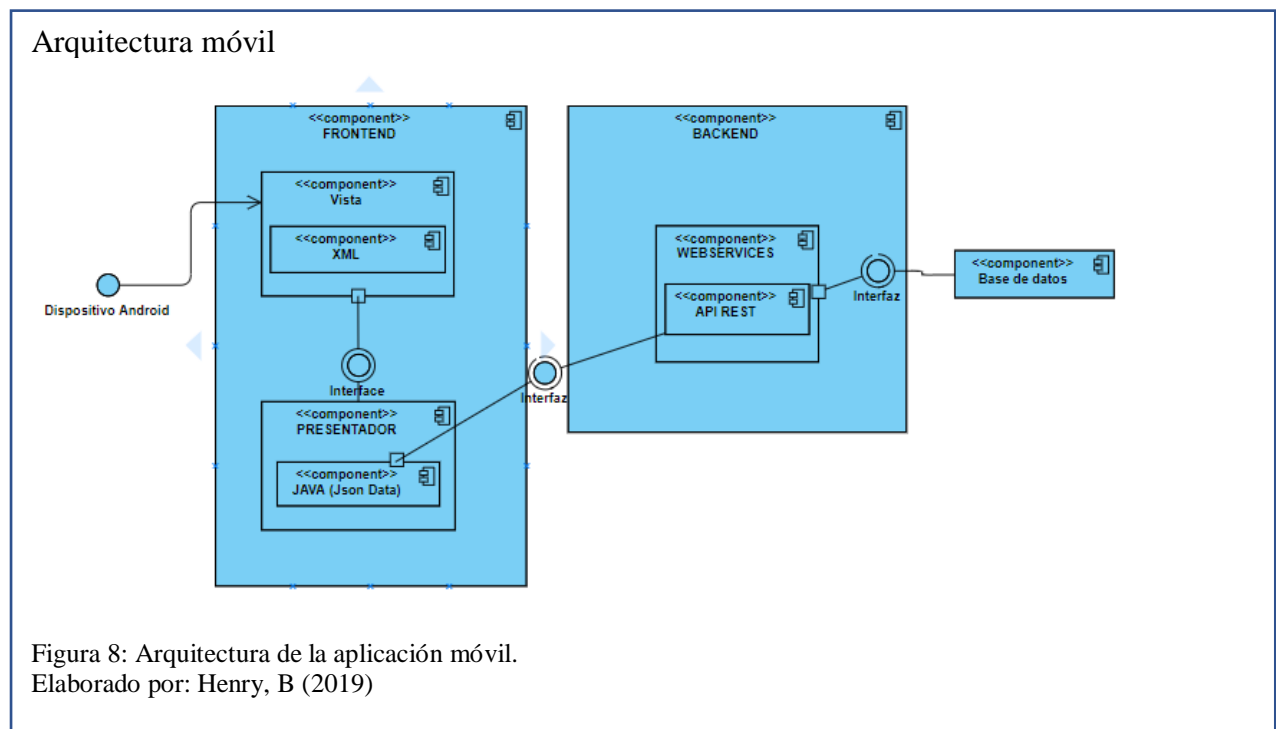


Figura 8: Arquitectura de la aplicación móvil.
Elaborado por: Henry, B (2019)

Se representa la comunicación entre la aplicación Android con la aplicación web los cuales interactúan de la siguiente manera: El usuario realiza una petición (GET, POST) a través de una vista, la cual es recibida por el servidor mediante un servicio web de tipo APIREST. Para la interpretación de la petición, Laravel hace uso de un archivo de ruta llamado api.php, la cual contiene las rutas disponibles para la aplicación móvil, mediante API. Por ejemplo, para la autenticación, la aplicación móvil crea un objeto JSONObjectRequest hacia una ruta perteneciente a la API de la aplicación web de tipo POST (<http://www.transportesec.com/api/login>).

El servidor mediante la API procesa la petición por medio de un método perteneciente a un controlador, ejecutará los procesos solicitados, y en caso de alguna afectación en base de datos, este se comunica con la misma mediante un modelo de datos haciendo uso de Eloquent de Laravel. Cada modelo de la aplicación es una representación de las tablas de base de datos MySQL de la aplicación. Desde el modelo se realizan las inserciones, actualizaciones, ediciones o eliminación de datos de la aplicación. Una vez que se ejecute la acción sobre la base de datos, el controlador valida la ejecución de la misma y devuelve a través de la API un mensaje de éxito en formato JSON con código 200, o un error con código 400 indicando en donde se presentó el error.

3.3 Clases y Métodos

3.3.1 Clases aplicación web

Tabla 3: Clases de la aplicación web.

Clases		
Clases aplicación web		
Nombre Clase	ubicación	Sirve para
CooperativaRuta	App/	Permite asignar un horario de salida a determinada ruta de una cooperativa. Gestiona la información de todas las rutas de una cooperativa para determinada hora.
Cooperativa	App/	Gestiona el registro de todas las cooperativas
User	App/	Gestiona el registro de todos los usuarios
Licencia	App/	Gestiona el registro de todas las licencias de conducir disponibles para los choferes al momento de su registro
ChoferesCooperativa	App/	Permite asociar un chofer con un bus y la cooperativa a la cual pertenece
CooperativaRutaActiva	App/	Permite la activación de rutas por cooperativa. También ayuda a decidir si una ruta estará activa para un bus en determinado día y hora
Bus	App/	Gestiona el registro de todos los buses, sin distinción de la cooperativa a la cual pertenecen. Es un maestro de buses.
Ruta	App/	Gestiona el registro de todas las rutas disponibles en la aplicación.
PasajerosBus	App/	Es un modelo de control, la cual permite saber el itinerario final de un bus al momento de su carga. Permite asociar una ruta activa con el chofer que realiza la carga del bus, la fecha de salida del bus y un identificador del pasajero, el cual se registra previamente en la aplicación. También posee un atributo de control, que permite mantener abierta o cerrada una lista de pasajeros.

Pasajero	App/	Gestiona el registro de todos los pasajeros que han sido escaneados o cargados por la aplicación móvil.
----------	------	---

Nota: Esta tabla contiene todas las clases de la aplicación web.

3.3.2 Métodos aplicación web

Tabla 4: Métodos de la aplicación web.

Métodos		
Métodos aplicación web		
Nombre Método	Pertenece a la clase	Sirve para
login	AndroidController	Utilizada en la autenticación del chofer desde la aplicación Android. Este método también devuelve todos los datos relacionada al chofer (horarios permitido de carga, pasajeros ya registrados, datos de la cooperativa a la que pertenece el chofer.
getDatosChofer	AndroidController	Genera un array (almacenamiento contiguo de elementos) que contiene toda la información del chofer. Son los datos que se devuelve en el login desde la aplicación Android.
guardarItinerario	AndroidController	Utilizado en el proceso de sincronización desde la aplicación móvil hacia la web. Es el método que utiliza la aplicación móvil mediante ApiRest para guardar toda la lista de pasajeros cargados en un bus a determinada hora.
index	BusController, CooperativaController, LicenciaController, RutaController, UserController	Permite consultar los datos ya registrados de los buses, cooperativas licencias, rutas o usuarios, y los retorna a la vista para mostrarlo como una lista para su mantenimiento
edit	BusController, CooperativaController, LicenciaController, RutaController, UserController	Permite consultar los datos ya registrados del bus, cooperativa, licencia, ruta o usuario. Esta información la envía al formulario de edición
store	BusController, CooperativaController, LicenciaController, RutaController, UserController	Almacena los datos del bus, cooperativa, licencia, ruta o usuario que se intenta registrar desde el formulario que corresponda
create	BusController, CooperativaController, LicenciaController, RutaController, UserController	Devuelve la vista que permite registrar un bus, cooperativa, licencia, ruta o usuario
update	BusController, CooperativaController, LicenciaController, RutaController, UserController	Actualiza los datos de un bus, cooperativa, licencia, ruta o usuario.
active	BusController, CooperativaController, LicenciaController,	Devuelve a estado activo una cooperativa, bus, ruta o usuario

	RutaController, UserController	
destroy	BusController, CooperativaController, LicenciaController, RutaController, UserController	Elimina de forma lógica (no definitiva) un registro de cooperativa, bus, ruta o usuario

Nota: Esta tabla contiene todos los métodos de la aplicación web.

3.2.3 Clases aplicación móvil

Tabla 5: Clases de la aplicación móvil.

Clases aplicación Móvil		
Nombre Clase	ubicación	Sirve para
Bus	app/java/com.barragan.transportesec/models	Serializar datos (almacenar datos en un archivo) de Buses
Cooperativa	app/java/com.barragan.transportesec/models	Serializar datos de la Cooperativa del usuario autenticado
Horario	app/java/com.barragan.transportesec/models	Serializar datos de Horarios permitidos para la carga de los pasajeros en determinado bus
Licencia	app/java/com.barragan.transportesec/models	Serializar datos de Licencia del chofer
Pasajero	app/java/com.barragan.transportesec/models	Serializar datos de Pasajeros (maestro)
PasajeroBus	app/java/com.barragan.transportesec/models	Modelo que permite mantener la información de los pasajeros que se cargaron en un bus a determinada hora.
Ruta	app/java/com.barragan.transportesec/models	Serializar datos de rutas disponibles
Usuario	app/java/com.barragan.transportesec/models	Serializar datos del chofer autenticado

Nota: Esta tabla contiene todas las clases de la aplicación móvil

3.2.4 Métodos aplicación móvil

Tabla 6: Métodos de la aplicación móvil.

Métodos aplicación Móvil		
Nombre Método	Pertenece a la clase	Sirve para
attemptLogin	LoginActivity	Proceso de autenticación. Para esto, envía una petición POST a la aplicación web. Una vez validado los datos para autenticación, la aplicación web envía a la aplicación móvil toda la información relacionada al chofer, así como también las rutas y horarios disponibles.
loadRutaData	RutaActivity	Carga los datos de rutas disponibles al control Spinner (Control de tipo ListBox en Android SDK)

loadHoraData	RutaActivity	Carga los datos de horarios disponibles al control Spinner (ListBox en Android SDK), según sea la ruta seleccionada.
requestCameraPermission	BarcodeCaptureActivity	Al momento de invocar la captura del código de barra, la aplicación requiere de permisos por parte del usuario del celular para poder hacer uso de la cámara del celular. Este método solicita estos permisos mediante el SDK de Android
onTouchEvent	BarcodeCaptureActivity	Es un Evento que se asocia al proceso de captura de código de barra de la cédula de identidad. Este evento solo escucha cuando el usuario pulsa la pantalla cuando la aplicación detecte el código de barra
onRequestPermissionsResult	BarcodeCaptureActivity	Es un evento que recibe el resultado de conceder o no los permisos de Android para la utilización de la cámara.
onClick	MainActivity	Evento asociado al botón que escanea la cédula.
onActivityResult	MainActivity	Evento que recibe la respuesta del escaneo de la cédula. Dentro de este evento se encuentra la consulta de los datos del pasajero en el Registro Civil. Antes de consulta en el registro civil, se valida que el usuario este registrado en base de datos mediante un json recibido desde la aplicación web.
guardarLista	MainActivity	Envía todos los datos recolectados en la carga de pasajeros a una ruta y hora específica. Sincroniza los datos enviados con los recibidos en la aplicación web. Es el proceso en donde se cierra la lista de pasajeros.

Nota: Esta tabla contiene todos los métodos de la aplicación móvil.

3.3 Diagrama de clases

3.3.1 Diagrama de clase aplicación web

El diagrama de clases de la aplicación web permite evidenciar la interacción de la aplicación web con la aplicación móvil al momento.

Diagrama de calases web

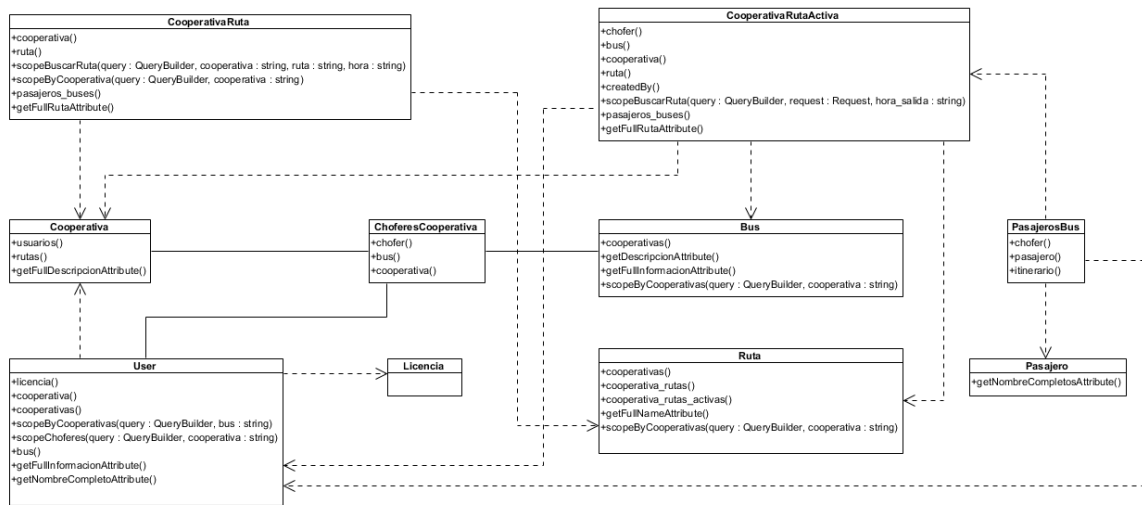


Figura 9: Diagrama de clases de la aplicación web.
Elaborado por: Henry, B (2019)

3.3.2 Diagrama de clase aplicación móvil

El diagrama de clases de la aplicación móvil permite evidenciar el funcionamiento de la aplicación móvil ya que es programado en un lenguaje diferente.

Diagrama de classes móvil

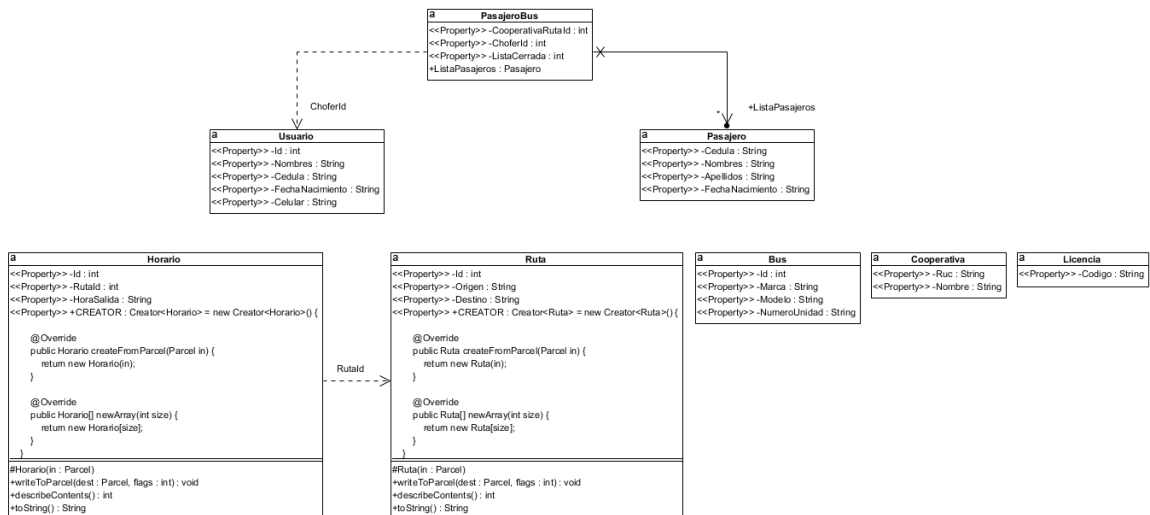


Figura 10: Diagrama de clases de la aplicación móvil.
Elaborado por: Henry, B (2019)

3.4 Código relevante

3.4.1 Manejo de la información y escaneada del código de barras

La funcionalidad detectada surge al momento de escanear el código de barras que es la parte más relevante de la aplicación es el siguiente, encontrar un método que controle la cámara para que capture el código de barras y extraiga el número de cédula que contiene dicho código, se lo resuelve de la siguiente manera.

Dodigo número de cédula

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    try {
        if (requestCode == RC_BARCODE_CAPTURE) {
            if (resultCode == CommonStatusCodes.SUCCESS) {
                if (data != null) {
                    //Obtencion del valor leído con la camara desde la otra actividad (Pantalla BarcodeCapture)
                    Barcode barcode = data.getParcelableExtra(BarcodeCaptureActivity.BarcodeObject);
                    Toast.makeText(getApplicationContext(), R.string.barcode_success, Toast.LENGTH_LONG).show();
                    barcodeValue.setText(barcode.displayValue);

                    //Busqueda de la cedula en el maestro de Pasajeros.
                    for(Pasajero pasajero: ListaPasajeros){
                        if (pasajero.getCedula().equals(barcode.displayValue)) {
                            tvNombresPasajero.setText(pasajero.getNombres());
                            tvApellidosPasajero.setText(pasajero.getApellidos());
                        }
                    }

                } else {

                    tvNombresPasajero.setText("");
                    tvApellidosPasajero.setText("");
                    Toast.makeText(getApplicationContext(), R.string.barcode_failure, Toast.LENGTH_LONG).show();

                }
            } else {
                Toast.makeText(getApplicationContext(), String.format(getString(R.string.barcode_error),
                    CommonStatusCodes.getStatusCodeString(resultCode)), Toast.LENGTH_LONG).show();
            }
        } else {
            super.onActivityResult(requestCode, resultCode, data);
        }
    } catch (Exception e) {
        // This will catch any exception, because they are all descended from Exception
        System.out.println("Error " + e.getMessage());
    }
}

/**
 * Starts or restarts the camera source, if it exists. If the camera source doesn't exist yet
 * (e.g., because onResume was called before the camera source was created), this will be called
 * again when the camera source is created.
 */
private void startCameraSource() throws SecurityException {
    // check that the device has play services available.
    int code = GoogleApiAvailability.getInstance().isGooglePlayServicesAvailable(
        getApplicationContext());
    if (code != ConnectionResult.SUCCESS) {
        Dialog dlg =
            GoogleApiAvailability.getInstance().getErrorDialog(this, code, RC_HANDLE_GMS);
        dlg.show();
    }

    if (mCameraSource != null) {
        try {
            mPreview.start(mCameraSource, mGraphicOverlay);
        } catch (IOException e) {
            Log.e(TAG, "Unable to start camera source.", e);
            mCameraSource.release();
            mCameraSource = null;
        }
    }
}
```

```

/**
 * onTap returns the tapped barcode result to the calling Activity.
 *
 * @param rawX - the raw position of the tap
 * @param rawY - the raw position of the tap.
 * @return true if the activity is ending.
 */
private boolean onTap(float rawX, float rawY) {
    // Find tap point in preview frame coordinates.
    int[] location = new int[2];
    mGraphicOverlay.getLocationOnScreen(location);
    float x = (rawX - location[0]) / mGraphicOverlay.getWidthScaleFactor();
    float y = (rawY - location[1]) / mGraphicOverlay.getHeightScaleFactor();

    // Find the barcode whose center is closest to the tapped point.
    Barcode best = null;
    float bestDistance = Float.MAX_VALUE;
    for (BarcodeGraphic graphic : mGraphicOverlay.getGraphics()) {
        Barcode barcode = graphic.getBarcode();
        if (barcode.getBoundingBox().contains((int) x, (int) y)) {
            // Exact hit, no need to keep looking.
            best = barcode;
            break;
        }
        float dx = x - barcode.getBoundingBox().centerX();
        float dy = y - barcode.getBoundingBox().centerY();
        float distance = (dx * dx) + (dy * dy); // actually squared distance
        if (distance < bestDistance) {
            best = barcode;
            bestDistance = distance;
        }
    }

    if (best != null) {
        Intent data = new Intent();
        data.putExtra(BarcodeObject, best);
        setResult(CommonStatusCodes.SUCCESS, data);
        finish();
        return true;
    }
    return false;
}

```

Figura 11: Código para extraer en número de cédula.
Elaborado por: Henry, B (2019)

Al hacer click en el botón “Escanear Cédula”, se instancia la vista o actividad que inicializa la cámara. Luego se invoca al método `startCameraSource()` el cual, es el encargado de inicializar la cámara del teléfono. Para esto, se usa las librerías nativas del SDK de Android. Básicamente lo que ocurre es que, se crea una capa que intercepta cada pulsación sobre la pantalla (Tap). Mientras esto ocurre, hay un proceso que se encarga de detectar la presencia del código de barras de la cédula de identidad, lo cual ocurre gracias al SDK de Android. Una vez se detecta el código de barras, se imprime en pantalla la numeración correspondiente al código de barras leído. Cuando se muestra esta numeración, es cuando el usuario pulsa la pantalla para enviar la numeración detectada a la actividad o vista que invoco a la cámara.

Para capturar finalmente este valor, se utiliza la función “onTap” (permite la capturar el número del código de barras), la cual pasa el número de cédula identificado a la vista “MainActivity” (recibe el número del código de barras). Esta actividad recibe este valor

mediante la función “onActivityResult”. Dentro de esta función se verifica el resultado de la lectura mediante cámara. Si el resultado fue correcto, se escribe el número de cédula en el campo correspondiente y luego se buscan los datos del pasajero en un maestro de datos. Si ya el usuario se encontraba registrado, se muestran los nombres y apellidos del mismo en la pantalla y se cargan a los campos correspondientes.

3.4.2 Consultar datos del chofer, conexión entre la aplicación móvil y web.

La funcionalidad detectada, es realizar la conexión entre la aplicación móvil y la aplicación web, y que la aplicación móvil pueda consumir datos de la aplicación web al momento de que el usuario ingrese a la aplicación móvil con sus credenciales, de ser exitoso el ingreso cargará los datos correspondientes al usuario (chofer, licencia, cooperativa, bus, ruta), se lo resuelve de la siguiente manera.

Conexión web y móvil

```
private void attemptLogin() {  
    // Reset errors.  
    mEmailView.setError(null);  
    mPasswordView.setError(null);  
  
    // Store values at the time of the login attempt.  
    String email = mEmailView.getText().toString();  
    String password = mPasswordView.getText().toString();  
  
    boolean cancel = false;  
    View focusView = null;  
  
    // Check for a valid password, if the user entered one.  
    if (!TextUtils.isEmpty(password) && !isPasswordValid(password)) {  
        mPasswordView.setError(getString(R.string.error_invalid_password));  
        focusView = mPasswordView;  
        cancel = true;  
    }  
  
    // Check for a valid email address.  
    if (TextUtils.isEmpty(email)) {  
        mEmailView.setError(getString(R.string.error_field_required));  
        focusView = mEmailView;  
        cancel = true;  
    } else if (!isEmailValid(email)) {  
        mEmailView.setError(getString(R.string.error_invalid_email));  
        focusView = mEmailView;  
        cancel = true;  
    }  
  
    if (cancel) {  
        // There was an error; don't attempt login and focus the first  
        // form field with an error.  
        focusView.requestFocus();  
    } else {  
        // Muestra el Spinner de progreso, y envia al background la tarea para realizar consulta de autenticacion con el servidor web  
        showProgress(true);  
        RequestQueue requestQueue = Volley.newRequestQueue(this);  
        final String url = getString(R.string.servidor).concat(getString(R.string.login_url));  
        Map<String, String> parametros = new HashMap();  
        parametros.put("email", mEmailView.getText().toString());  
        parametros.put("password", mPasswordView.getText().toString());  
  
        JSONObject parameters = new JSONObject(parametros);  
        JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(Request.Method.POST, url, parameters, new Response.Listener<JSONObject>() {  
  
            @Override  
            public void onResponse(JSONObject response) {  
                try {  
                    //StringBuilder formattedResult = new StringBuilder();  
                    if (response.isNull("error") == true) {  
  
                        Gson gson = new Gson();  
  
                        //Datos del Usuario autenticado  
                        usuario = gson.fromJson(response.getJSONObject("usuario").toString(), Usuario.class);  
  
                        //Datos de la Licencia  
                        licencia = gson.fromJson(response.getJSONObject("licencia").toString(), Licencia.class);  
  
                        //Datos de la Cooperativa  
                        cooperativa = gson.fromJson(response.getJSONObject("cooperativa").toString(), Cooperativa.class);  
  
                        //Datos del Bus asignado  
                        bus = gson.fromJson(response.getJSONObject("bus").toString(), Bus.class);  
  
                        //Listado de rutas disponibles  
                        rutas = response.getJSONArray("rutas");  
                        ListaRutas = Arrays.asList(gson.fromJson(rutas.toString(), Ruta[].class));  
                        Type typeListaRutas = new TypeToken<List<Ruta>>().getType();  
                        String jsonListaRutas = gson.toJson(ListaRutas, typeListaRutas);  
  
                        //Listado de horarios disponibles por ruta  
                        horarios = response.getJSONArray("horarios");  
                        ListaHorarios = Arrays.asList(gson.fromJson(horarios.toString(), Horario[].class));  
                        Type typeListaHorarios = new TypeToken<List<Horario>>().getType();  
                        String jsonListaHorarios = gson.toJson(ListaHorarios, typeListaHorarios);  
  
                        //Listado de pasajeros registrados hasta el momento en el sistema  
                        pasajeros = response.getJSONArray("pasajeros");  
                        ListaPasajeros = Arrays.asList(gson.fromJson(pasajeros.toString(), Pasajero[].class));  
                        Type typeListaPasajeros = new TypeToken<List<Pasajero>>().getType();  
                        String jsonListaPasajeros = gson.toJson(ListaPasajeros, typeListaPasajeros);  
  
                        itinerario.setListaCerrada(0);  
  
                        //Ventana (actividad) que muestra un menu con informacion del chofer  
                        Intent myMenu = new Intent(getApplicationContext(), MenuActivity.class);  
                    }  
                }  
            }  
        });  
        requestQueue.add(jsonObjectRequest);  
    }  
}
```

```

//Parametrización de valores entre pantallas (datos del chofer, itinerario, pasajeros, etc
myMenu.putExtra("usuario", (new Gson()).toJson(usuario));
myMenu.putExtra("licencia", (new Gson()).toJson(licencia));
myMenu.putExtra("cooperativa", (new Gson()).toJson(cooperativa));
myMenu.putExtra("bus", (new Gson()).toJson(bus));
myMenu.putExtra("hora", (new Gson()).toJson(hora));
myMenu.putExtra("ruta", (new Gson()).toJson(ruta));
myMenu.putExtra("itinerario", (new Gson()).toJson(itinerario));
myMenu.putExtra("rutas", jsonListaRutas);
myMenu.putExtra("horarios", jsonListaHorarios);
myMenu.putExtra("pasajeros", jsonListaPasajeros);
myMenu.putExtra("fecha_servidor", response.get("fecha_servidor").toString());

//Iniciar MENU
startActivity(myMenu);

} else {
    Toaster.makeLongToast(getApplicationContext(), response.getString("error"), 10000);
}

} catch (JSONException e) {
    Toaster.makeLongToast(getApplicationContext(), e.getMessage(), 6000);
}
showProgress(false);
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        showProgress(false);
        Toast.makeText(getApplicationContext(), error.toString(),
            Toast.LENGTH_LONG).show();
    }
});
requestQueue.add(jsonObjectRequest);
}
}

```

Figura 12: Conexión entre la aplicación web y la aplicación móvil.
Elaborado por: Henry, B (2019)

Para el funcionamiento de la aplicación móvil, la misma solicita el email y password del usuario (Chofer). Una vez se validan los datos, se realiza una petición POST hacia un API (aplicación WEB), mediante el uso de un JsonObjectRequest, el cual invoca la ruta <http://www.transportesec.com/api/login>. La aplicación web detecta la petición, respondiendo a la misma con un JSON con código 200, la cual contiene todos los datos del chofer, cooperativa a la que pertenece, rutas y horarios disponibles para cargar, y el maestro de pasajeros.

Luego, la aplicación móvil recibe esta respuesta en formato JSON, la decodifica, separando cada conjunto de datos en modelos de datos pertenecientes a la aplicación móvil. Estos modelos se pasan posteriormente entre cada vista o actividad.

3.4.3 Generar la lista de pasajeros en la aplicación móvil.

El desafío de programación está en enviar al servidor la lista generada, y que esté disponible para que puedan descargar en la aplicación web, con todos los datos

correspondientes como por ejemplo (la fecha, ruta, hora del viaje, datos del chofer, datos del bus, datos del pasajero).

Lista de pasajeros

```
private void guardarLista() {
    showProgress(true);
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    final String url = getString(R.string.servidor).concat(getString(R.string.guardar_itinerario));
    Map<String, String> parametros = new HashMap();
    parametros.put("usuario", (new Gson()).toJson(usuario));
    parametros.put("itinerario", (new Gson()).toJson(itinerario));

    JSONObject parameters = new JSONObject(parametros);
    JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(Request.Method.POST, url, parameters, new Response.Listener<JSONObject>() {

        @Override
        public void onResponse(JSONObject response) {
            try {
                if (response.isNull("error")) {

                    Gson gson = new Gson();

                    //Datos del Usuario autenticado
                    usuario = gson.fromJson(response.getJSONObject("usuario").toString(), Usuario.class);

                    //Datos de la Licencia
                    licencia = gson.fromJson(response.getJSONObject("licencia").toString(), Licencia.class);

                    //Datos de la Cooperativa
                    cooperativa = gson.fromJson(response.getJSONObject("cooperativa").toString(), Cooperativa.class);

                    //Datos del Bus asignado
                    bus = gson.fromJson(response.getJSONObject("bus").toString(), Bus.class);

                    //Listado de rutas disponibles
                    rutas = response.getJSONArray("rutas");
                    ListaRutas = Arrays.asList(gson.fromJson(rutas.toString(), Ruta[].class));
                    Type typeListaRutas = new TypeToken<List<Ruta>>() {}.getType();
                    String jsonListaRutas = gson.toJson(ListaRutas, typeListaRutas);

                    //Listado de horarios disponibles por ruta
                    horarios = response.getJSONArray("horarios");
                    ListaHorarios = Arrays.asList(gson.fromJson(horarios.toString(), Horario[].class));
                    Type typeListaHorarios = new TypeToken<List<Horario>>() {}.getType();
                    String jsonListaHorarios = gson.toJson(ListaHorarios, typeListaHorarios);

                    //Listado de horarios disponibles por ruta
                    horarios = response.getJSONArray("horarios");
                    ListaHorarios = Arrays.asList(gson.fromJson(horarios.toString(), Horario[].class));
                    Type typeListaHorarios = new TypeToken<List<Horario>>() {}.getType();
                    String jsonListaHorarios = gson.toJson(ListaHorarios, typeListaHorarios);

                    //Listado de pasajeros registrados hasta el momento en el sistema
                    pasajeros = response.getJSONArray("pasajeros");
                    ListaPasajeros = Arrays.asList(gson.fromJson(pasajeros.toString(), Pasajero[].class));
                    Type typeListaPasajeros = new TypeToken<List<Pasajero>>() {}.getType();
                    String jsonListaPasajeros = gson.toJson(ListaPasajeros, typeListaPasajeros);

                    //Inicializamos nuevamente el objeto Itinerario
                    itinerario.setListaCerrada(0);

                    //Ventana (actividad) que muestra un menu con informacion del chofer
                    Intent myMenu = new Intent(getBaseContext(), MenuActivity.class);

                    //Parametrizacion de valores entre pantallas (datos del chofer, itinerario, pasajeros, etc
                    myMenu.putExtra("usuario", (new Gson()).toJson(usuario));
                    myMenu.putExtra("licencia", (new Gson()).toJson(licencia));
                    myMenu.putExtra("cooperativa", (new Gson()).toJson(cooperativa));
                    myMenu.putExtra("bus", (new Gson()).toJson(bus));
                    myMenu.putExtra("hora", (new Gson()).toJson(hora));
                    myMenu.putExtra("ruta", (new Gson()).toJson(ruta));
                    myMenu.putExtra("itinerario", (new Gson()).toJson(itinerario));
                    myMenu.putExtra("rutas", jsonListaRutas);
                    myMenu.putExtra("horarios", jsonListaHorarios);
                    myMenu.putExtra("pasajeros", jsonListaPasajeros);
                    myMenu.putExtra("fecha_servidor", response.get("fecha_servidor").toString());
                }
            }
        }
    });
    requestQueue.add(jsonObjectRequest);
}
```

Figura 13: Código para crear la lista de pasajeros.
Elaborado por: Henry, B (2019)

Cuando el chofer da por finalizada la carga de pasajeros para un bus en una ruta y hora determinada, la aplicación móvil arma un JSON con datos del usuario y del itinerario. En este itinerario se envía los datos de todos los pasajeros cargados al bus, para una ruta y hora.

Para el envío, se realiza una petición POST hacia una API (aplicación WEB), mediante el uso de un `JsonObjectRequest`, el invoca la ruta `http://www.transportesec.com/api/guardar/itinerario`. La aplicación web detecta la petición, respondiendo con un JSON con código 200, la cual contiene un mensaje de respuesta de éxito y también una nueva lista de pasajeros actualizada.

En la aplicación web se recibe todo el itinerario, decodificando los datos de los pasajeros, e insertando en base de datos a cada pasajero, asociándole la ruta y hora de salida del bus, junto con el chofer que presto el servicio.

Luego, la aplicación móvil recibe esta respuesta en formato JSON, la decodifica, separando cada conjunto de datos en modelos de datos pertenecientes a la aplicación móvil. Estos modelos se pasan posteriormente entre cada vista o actividad.

3.4.4 Validación para activar y desactivar los viajes

La funcionalidad detectada está basada en la seguridad de la aplicación móvil, al tener una aplicación móvil de libre acceso al momento de generar un usuario con el registro de un chofer, los choferes podrían acceder en cualquier momento a la aplicación y hacer uso indebido de la misma al registrar pasajeros fuera de la hora de los viajes, se lo resuelve de la siguiente manera.

Activar y desactivar viajes

```
/**
 * Remove the specified resource from storage.
 *
 * @param App\Licencia $licencia
 * @return Illuminate\Http\Response
 */
public function destroy(CooperativaRuta $ruta)
{
    $ruta->delete();
    return redirect()->route('rutas.index')->with('msg', trans('globals.success_alert_content'));
}

/**
 * Active the specified resource from storage
 * @param int $id Licencia Id
 * @return Illuminate\Http\Response
 */
public function active($id) {
    $ruta = CooperativaRuta::onlyTrashed()->where('id',$id);
    $ruta->restore();
    return redirect()->route('rutas.index');
}
```

Figura 14: Validación para activar y desactivar los viajes.
Elaborado por: Henry, B (2019)

Dentro del controlador RutaController, existen dos métodos comunes para algunos otros controladores. Estos métodos se llaman “destroy” y “active”. El método “destroy” recibe una instancia del viaje de la cooperativa que se desea eliminar de forma lógica (no permanente). Para eliminar un registro de manera lógica, Laravel mediante el modelo CooperativaRuta, actualiza el registro de la ruta insertando una fecha y hora en el campo “deleted_at”. Esto hace que el registro sea invisible para cualquier consulta realizada respecto a rutas activas.

El método “active” recibe una instancia del viaje de la cooperativa que se desea activar de forma lógica (no permanente). Para activar un registro de manera lógica, Laravel mediante el modelo CooperativaRuta, actualiza el registro de la ruta insertando un valor ‘null’ en el campo “deleted_at”. Esto hace que el registro vuelva a ser visible para cualquier consulta realizada respecto a rutas activas.

3.4.5 Generar la lista de pasajeros

El desafío de programación es generar la lista en la aplicación web después de ser generada por la aplicación móvil y extraer todos los datos como por ejemplo (la fecha, ruta, hora del viaje, datos del chofer, datos del bus, datos del pasajero) que se encuentran guardados en la base de datos, todos estos campos tienen que estar bien definidos, ordenados y tener simetría, se lo resuelve de la siguiente manera.

Lista de pasajeros

```
public function reportePasajeros(Request $request, $ruta) {
    $pasajeros = PasajerosBus::where('cooperativa_rutas_id', $ruta)
        →where('fecha_salida', $request→fecha_salida)→get();

    $chofer = ChoferesCooperativa::with(['chofer', 'bus', 'cooperativa'])→where('user_id', $pasajeros→first()→chofer_id)→first();

    $data_pdf['pasajeros'] = $pasajeros;
    $data_pdf['chofer'] = $chofer;

    $pdf = \PDF::loadView('pdf.lista_pasajeros', $data_pdf);
    return $pdf→download('listado.pdf');
}

public function reportePasajeros(Request $request, $ruta) {
    $pasajeros = PasajerosBus::where('cooperativa_rutas_id', $ruta)
        →where('fecha_salida', $request→fecha_salida)→get();

    $chofer = ChoferesCooperativa::with(['chofer', 'bus', 'cooperativa'])→where('user_id', $pasajeros→first()→chofer_id)→first();

    $data_pdf['pasajeros'] = $pasajeros;
    $data_pdf['chofer'] = $chofer;

    $pdf = \PDF::loadView('pdf.lista_pasajeros', $data_pdf);
    return $pdf→download('listado.pdf');
}
```

Figura 15: Generar de la lista de pasajeros.
Elaborado por: Henry, B (2019)

Para la generación de la lista de pasajeros se recibe la hora de salida y ruta desde la vista la lista de pasajeros. Luego, se hace una consulta en base de datos mediante el Modelo PasajeroBus, la cual retorna todos los datos relacionados a los pasajeros cargados al bus. Estos datos se encuentran en la tabla “pasajeros_buses”.

3.4.6 Llenar la cédula manualmente y al poner el último número automáticamente sale los datos de los pasajeros.

Funcionalidad destacada, al momento de solicitar la cédula y el pasajero no la disponga se pueda solicitar el número de cédula manualmente y al terminar de ingresar el último dígito se puedan llenar los campos (cédula, nombres, apellidos) automáticamente sin necesidad de presionar un botón, se lo resuelve de la siguiente manera.

Ingreso de cédula

```
//Cuando se ingresa una cedula manualmente, se realiza una busqueda de la misma en la lista de pasajeros actual
barcodeValue.addTextChangedListener(new TextWatcher() {

    public void afterTextChanged(Editable s) {}

    public void beforeTextChanged(CharSequence s, int start, int count, int after) {}

    public void onTextChanged(CharSequence s, int start, int before, int count) {

        if (s.length() == 10) {
            for (Pasajero pasajero : ListaPasajeros) {
                if (pasajero.getCedula().equals(s.toString())) {
                    tvNombresPasajero.setText(pasajero.getNombres());
                    tvApellidosPasajero.setText(pasajero.getApellidos());
                }
            }
        } else {
            tvNombresPasajero.setText("");
            tvApellidosPasajero.setText("");
        }
    }
});
```

Figura 16: Lenado automático para ingresar del número de la cédula manualmente.
Elaborado por: Henry, B (2019)

En la vista de carga de pasajeros, el campo “cédula de identidad” es editable. Para validar una cédula de identidad, existe un evento atado al campo llamado “addTextChangedListener”. Este evento recibe un parámetro de tipo TextWatcher, el cual se encarga de inspeccionar cada carácter ingresado. Cuando el usuario termina de ingresar los diez dígitos de la cédula, la aplicación móvil válida que esta cédula se encuentre en la tabla de pasajeros. Si la cédula existe, la aplicación móvil escribe los nombres y apellidos a los campos correspondientes.

3.5 Pruebas

Para afianzar el correcto desempeño y funcionamiento de la aplicación web y la aplicación móvil, y poder evaluar la su funcionalidad independiente y a su vez su funcionalidad en conjunto, para satisfacer y dar cumplimiento a los requerimientos, se someten a diferentes tipos de pruebas que permite verificar el nivel del trabajo.

3.5.1 Plan de pruebas

Para validar el correcto funcionamiento de las aplicaciones web y móvil y a su vez cumplan con los requerimientos, se diseñan tres tipos de pruebas (pruebas funcionales, pruebas de usabilidad, pruebas de rendimiento) que permite evaluar su correcto funcionamiento, en el anexo 2 se encuentra a detalle todos los resultados del plan de pruebas.

3.5.1.1 Pruebas funcionales

Sometiendo la aplicación móvil y la aplicación web a las pruebas funcionales, centrándose en que las aplicaciones tienen un adecuado funcionamiento de acuerdo a las funcionalidades expuestas en los requerimientos, se busca valorar cada uno de las componentes funcionales que manejan las aplicaciones web y móvil, funcionen y respondan de acuerdo a lo expuesto, garantizando que todos los requerimientos funcionales fueron atendidos y dados la importación del caso.

Tabla 7: Resumen de las pruebas funcionales aplicación web.

Resultados de pruebas funcionales				
Aplicación web				
Descripción		Resultado		Número de pruebas
General	Específica	Correcto	Error	
1. Inicio de sesión, administrado de cooperativa (5 inicios de sesión)	a. Contraseña ingresada erróneamente (mensaje de error)	X		5
	b. Contraseña ingresada correctamente	X		5
2. Crear un chofer nuevo	a. Verificar mensaje de error si la cédula tiene menos de 10 dígitos	X		5
	b. Verificar mensaje de error si la cédula tiene más de 10 dígitos	X		5
	c. Validar que la cédula tenga 10 dígitos	X		5
	d. Verificar mensaje de error si la contraseña no está ingresada.	X		5
	e. Validar que el email cumpla con el formato de correo electrónico	X		5

	f. Validar mensaje de error si faltan llenar los campos del formulario	X		5
3. Desactivar y activar choferes creados	a. Activar choferes creados	X		5
	b. Desactivara choferes creados	X		5
4. Editar un chofer creado	a. Cambiar el tipo de licencia	X		5
	b. Cambiar el número de celular	X		5
5. Crear un nuevo bus	a. Verificar validación para que el número de unidad sea único (mensaje de error)	X		5
	b. Verificar validación para que el chasis sea único (mensaje de error)	X		5
	c. Validar que el número de la placa tenga una longitud máxima de 7 y mínimo 6 (mensaje de error)	X		5
6. Asignar un chofer a un bus		X		5

7. Desactivar y activar un bus creado.		X		5
8. Editar un bus.		X		5
9. Activar un viaje para diferentes buses	a. Activar un viaje para las 10:30	X		5
	b. Activar un viaje para las 11:30	X		5
	c. Desactivar el viaje de las 10:30	X		5
	d. Activar un viaje para las 11:30	X		5
10. Generar la lista de pasajeros	a. Generar la lista de pasajeros para las 10:30	X		5
	b. Generar la lista de pasajeros para las 11:30	X		5
	c. Descargar el PDF de la lista de pasajeros para las 10:30	X		5
	d. Descargar el PDF de la lista de pasajeros para las 11:30	X		5
11. Inicio de sesión, 911-Policía nacional inicio de sesión, (5 inicios de sesión)	a. Contraseña ingresada erróneamente (mensaje de error)	X		5

	b. Contraseña ingresada correctamente	X		5
12. Generar la lista de pasajeros iniciada cesión 911- Policía nacional	a. Generar la lista de pasajeros cooperativa 10:30	X		5
	b. Generar la lista de pasajeros cooperativa 11:30	X		5
	c. Generar el PDF de la lista de pasajeros cooperativa 10:30	X		5
	d. Generar el PDF de la lista de pasajeros cooperativa 11:30	X		5

Nota: Esta tabla contiene todos los resultados de las pruebas funcionales

Teniendo como resultado un 100%, después de ser sometida la aplicación web a las distintas pruebas funcionales lo cual corrobora que la aplicación web ha cumplido satisfactoriamente los requerimientos establecidos.

Tabla 8: Resumen de las pruebas funcionales aplicación móvil.

Resultados Pruebas funciones				
Aplicación móvil				
Descripción		Resultado		Número de pruebas
General	Especifica	Correcto	Error	
1. Inicio de cesión, con el usuario asignado al chofer (5 inicios de sesión)	a. Contraseña ingresada erróneamente (mensaje de error)	X		5
	b. Que el viaje no esté activado (mensaje informativo)	X		5
2. Realizar el registro de pasajeros para los horarios 10:30 y 11:30	a. Registrar pasajero manualmente y guardar registro	X		5
	b. Registrar pasajero escaneado la cédula y guardar el registro	X		5
	c. Registrar pasajero manualmente y guardar registro	X		5
	d. Registrar pasajero escaneado la cédula y guardar el registro	X		5
	e. Cerrar la lista definitivamente	X		5

Nota: Esta tabla contiene todos los resultados de las pruebas funcionales

Teniendo como resultado un 100% después de ser sometida la aplicación móvil a las distintas pruebas funcionales lo cual corrobora que la aplicación móvil ha cumplido satisfactoriamente los requerimientos establecidos.

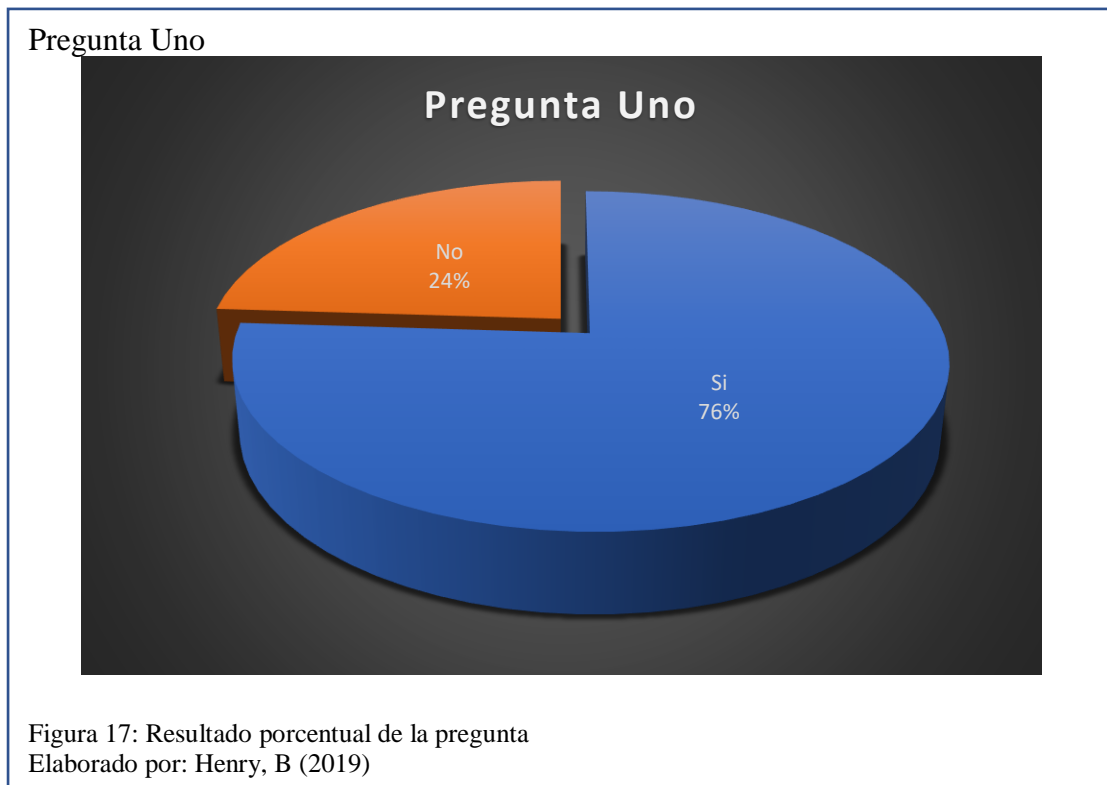
3.5.1.2 Pruebas de usabilidad

Para ser llevadas a cabo las pruebas de usabilidad se diseña una encuesta estructurada de tal forma ayude a evidenciar la aceptación por los usuarios que están conformados con los perfiles de choferes y administrativos.

Tabla 9: Resultados la pregunta 1.

Pregunta número 1		
Descripción:	1. ¿Considera que el diseño de la interfaz, estructura, organización de la aplicación web son adecuados?	
Número Encuestas	Resultado	
	SI	NO
Muestreo	19	6

Nota: Esta tabla contiene el resultado de la pregunta 1

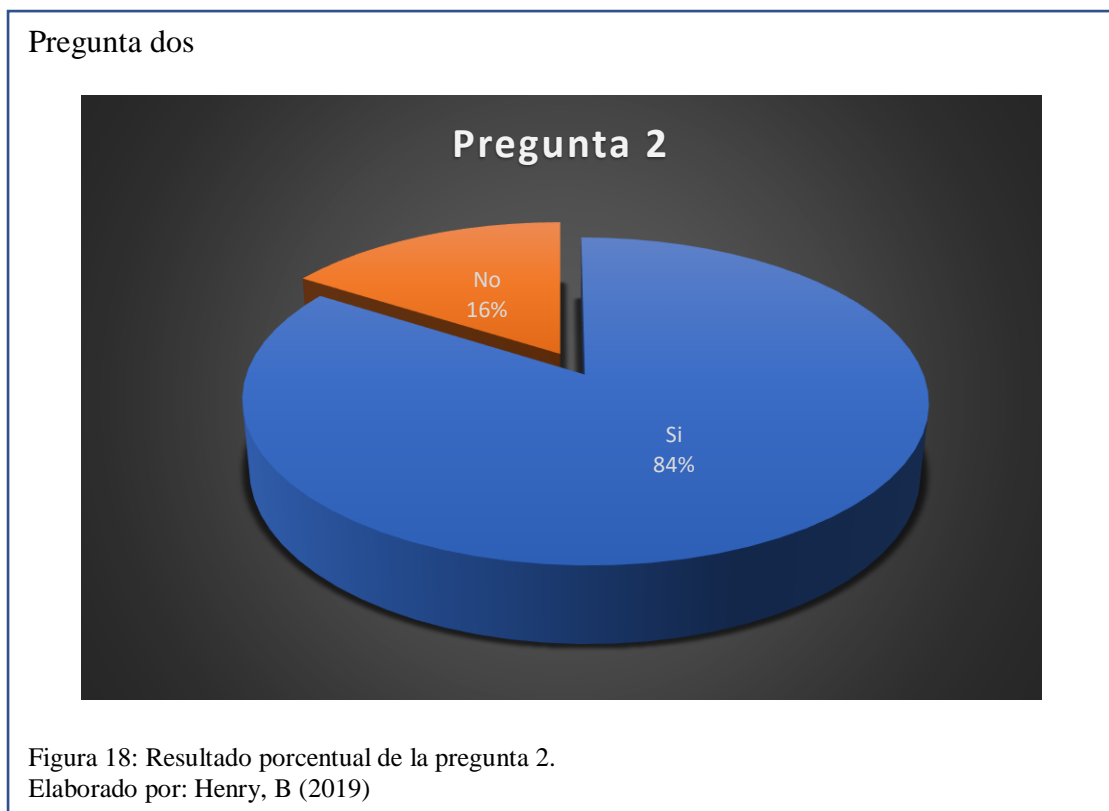


En la representación porcentual del resultado de la pregunta 1 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 76%, de aceptación es considerado que el diseño de la interfaz gráfica enfocado en la estructura y organización de la aplicación web, es adecuado para los 19 usuarios que representa el 76%, y 6 usuarios que representan el 24% que no le pareció adecuado.

Tabla 10: Resultados la pregunta 2.

Pregunta número 2		
Descripción:	2. ¿Consideras que el diseño de la interfaz: ¿accesibilidad, navegación, de la página web son adecuados?	
Número Encuestas	Resultado	
	SI	NO
Muestreo	21	4

Nota: Esta tabla contiene el resultado de la pregunta 2

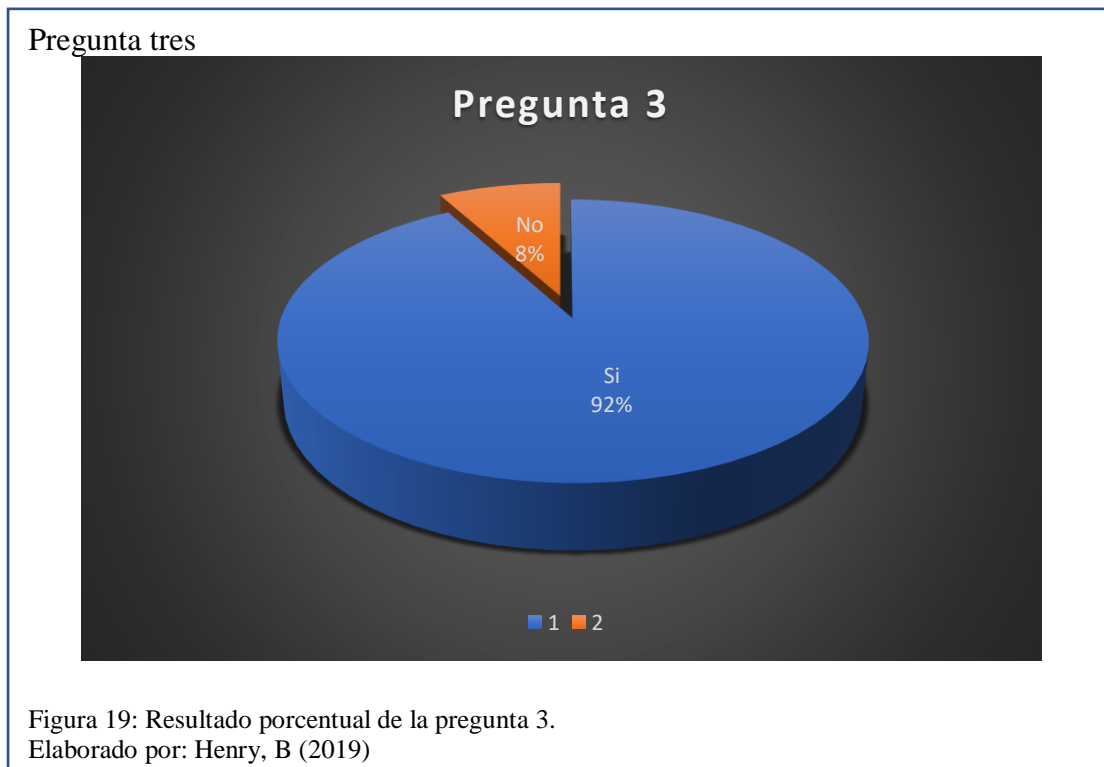


En la representación porcentual del resultado de la pregunta 2 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 84% de aceptación es considerado que el diseño de la interfaz gráfica enfocado en la accesibilidad y navegación de la aplicación web, es adecuado para los 21 usuarios que representa el 76%, y 4 usuarios que representan el 16% que no le pareció adecuado.

Tabla 11: Resultados la pregunta 3.

Pregunta número 3		
Descripción:	3. Le pareció interesante la aplicación web	
Número Encuestas	Resultado	
	SI	NO
Muestreo	23	2

Nota: Esta tabla contiene el resultado de la pregunta 3



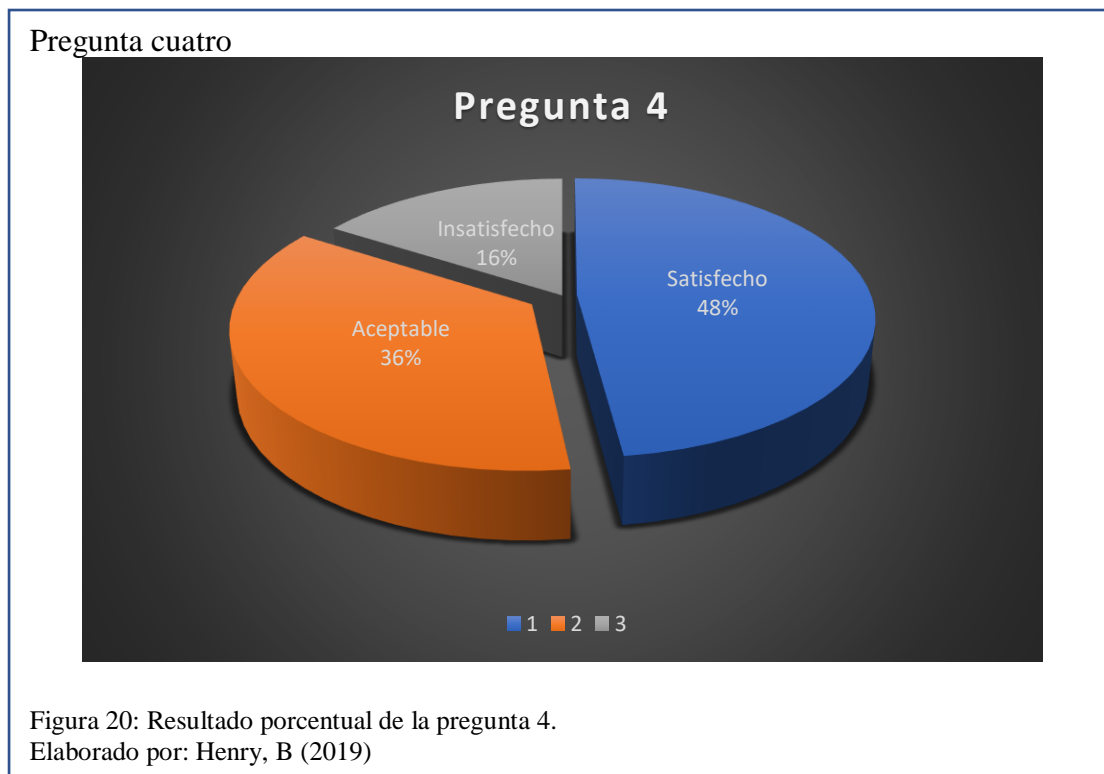
En la representación porcentual del resultado de la pregunta 3 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un

92% de aceptación es considerado que a los usuarios encuestados les pareció interesante la aplicación web, para los 23 usuarios que representa el 76%, y 2 usuarios que representan el 8% que no le pareció interesante.

Tabla 12: Resultados la pregunta 4.

Pregunta número 4			
Descripción:	4. ¿La aplicación web es fácil de usar?		
Número Encuestas	Resultado		
	Satisfecho	Aceptable	Insatisfecho
Muestreo	12	9	4

Nota: Esta tabla contiene el resultado de la pregunta 4

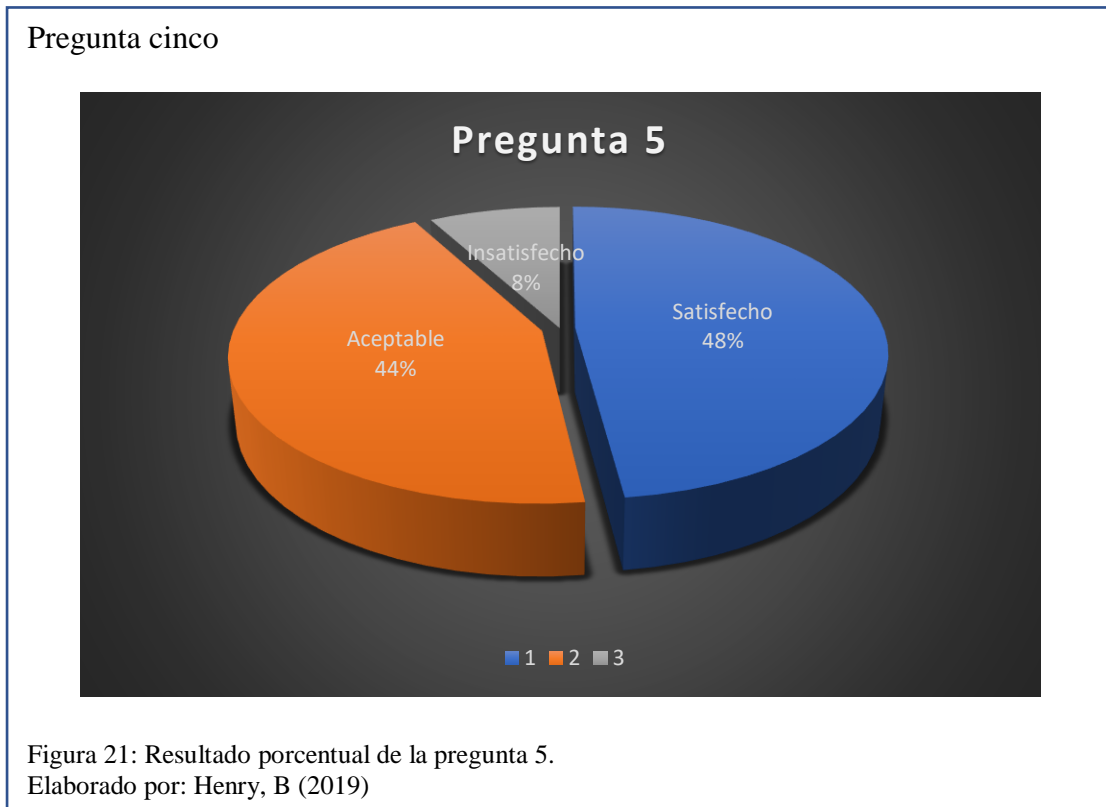


En la representación porcentual del resultado de la pregunta 4 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 48% de satisfacción es considerado que a los usuarios encuestados les pareció fácil de usar la aplicación web, para los 12 usuarios que representa el 48%, 9 usuarios que representan el 36% les pareció aceptable, y para un 16% no les pareció fácil de usar.

Tabla 13: Resultados la pregunta 5.

Pregunta número 5			
Descripción:	5. ¿Cuál es la impresión general de la aplicación web?		
Número Encuestas	Resultado		
	Satisfecho	Aceptable	Insatisfecho
Muestreo	12	11	2

Nota: Esta tabla contiene el resultado de la pregunta 5

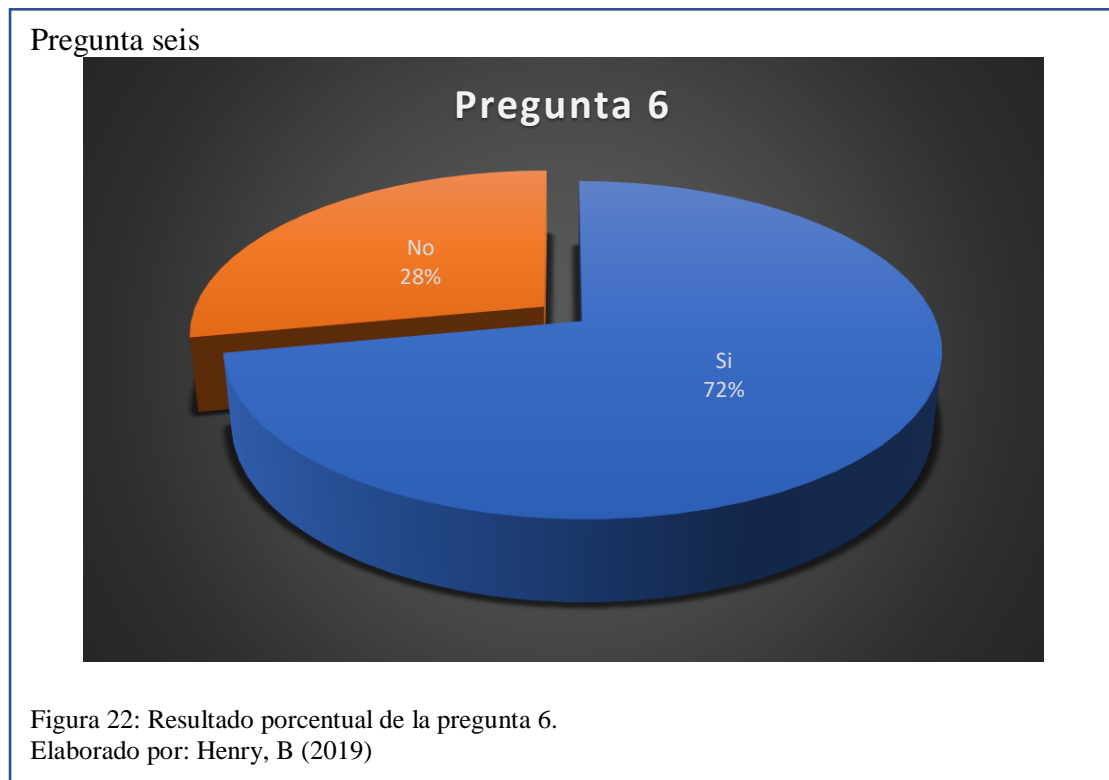


En la representación porcentual del resultado de la pregunta 5 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 48% están satisfechos considerado que a los usuarios encuestados tiene una buena impresión de la aplicación web, para los 12 usuarios que representa el 48%, 11 usuarios que representan el 44% les pareció aceptable, y para un 8% no tienen una buena impresión.

Tabla 14: Resultados la pregunta 6.

Pregunta número 6	
Descripción:	6. ¿Considera que el diseño de la interfaz, estructura, organización de la aplicación móvil son adecuados?
Número Encuestas	Resultado
Muestreo	18 7

Nota: Esta tabla contiene el resultado de la pregunta 6

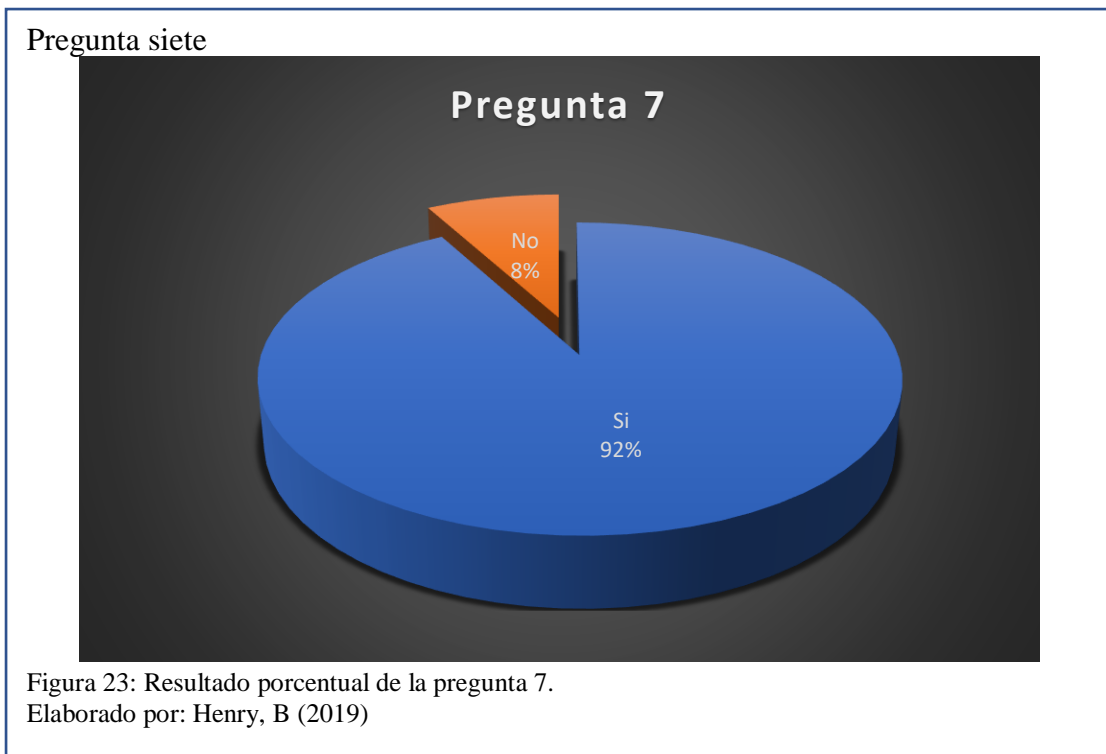


En la representación porcentual del resultado de la pregunta 6 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 72% de aceptación es considerado que el diseño de la interfaz gráfica enfocado en la estructura y organización de la aplicación móvil es adecuado para los 18 usuarios que representa el 72%, y 7 usuarios que representan el 28% que no le pareció adecuado.

Tabla 15: Resultados la pregunta 7.

Pregunta número 7	
Descripción:	7. ¿Consideras que el diseño de la interfaz: ¿accesibilidad, navegación, de la página móvil son adecuados?
Número Encuestas	Resultado
Muestreo	23 2

Nota: Esta tabla contiene el resultado de la pregunta 7

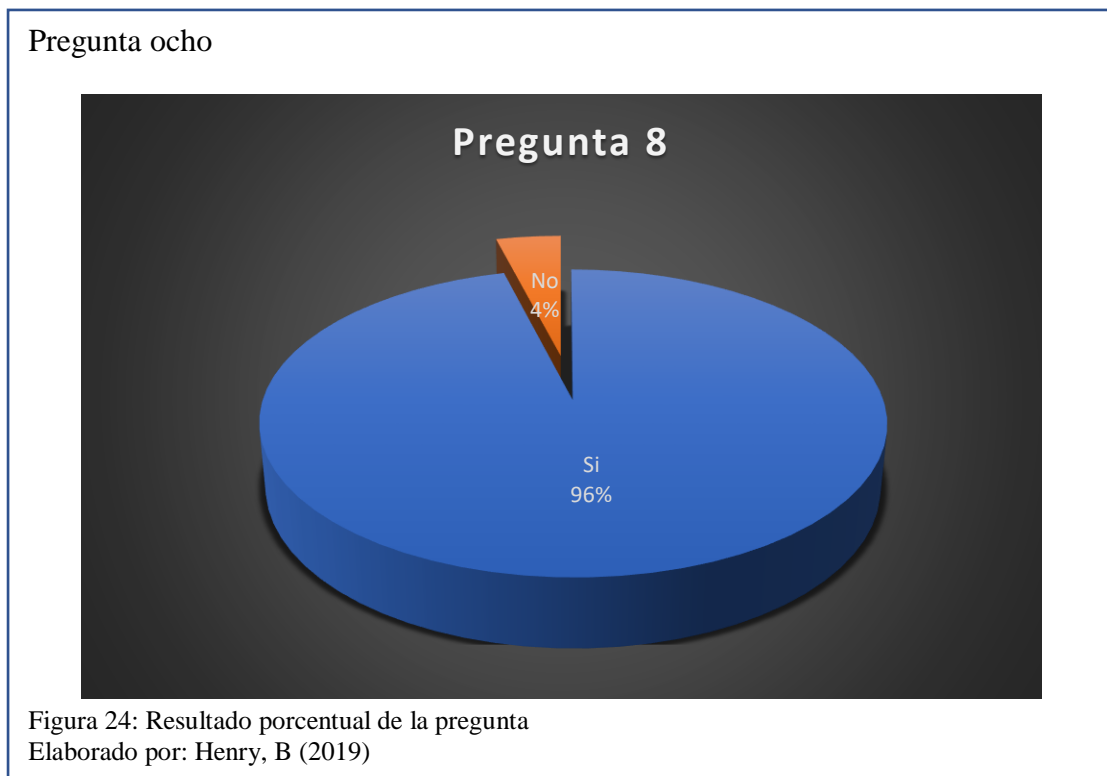


En la representación porcentual del resultado de la pregunta 2 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 92% de aceptación es considerado que el diseño de la interfaz gráfica enfocado en la accesibilidad y navegación de la aplicación móvil es adecuado para los 23 usuarios que representa el 92%, y 2 usuarios que representan el 8% que no le pareció adecuado.

Tabla 16: Resultados la pregunta 8.

Pregunta número 8		
Descripción:	8. Le pareció interesante la aplicación móvil	
Número Encuestas	Resultado	
	SI	NO
Muestreo	24	1

Nota: Esta tabla contiene el resultado de la pregunta 8

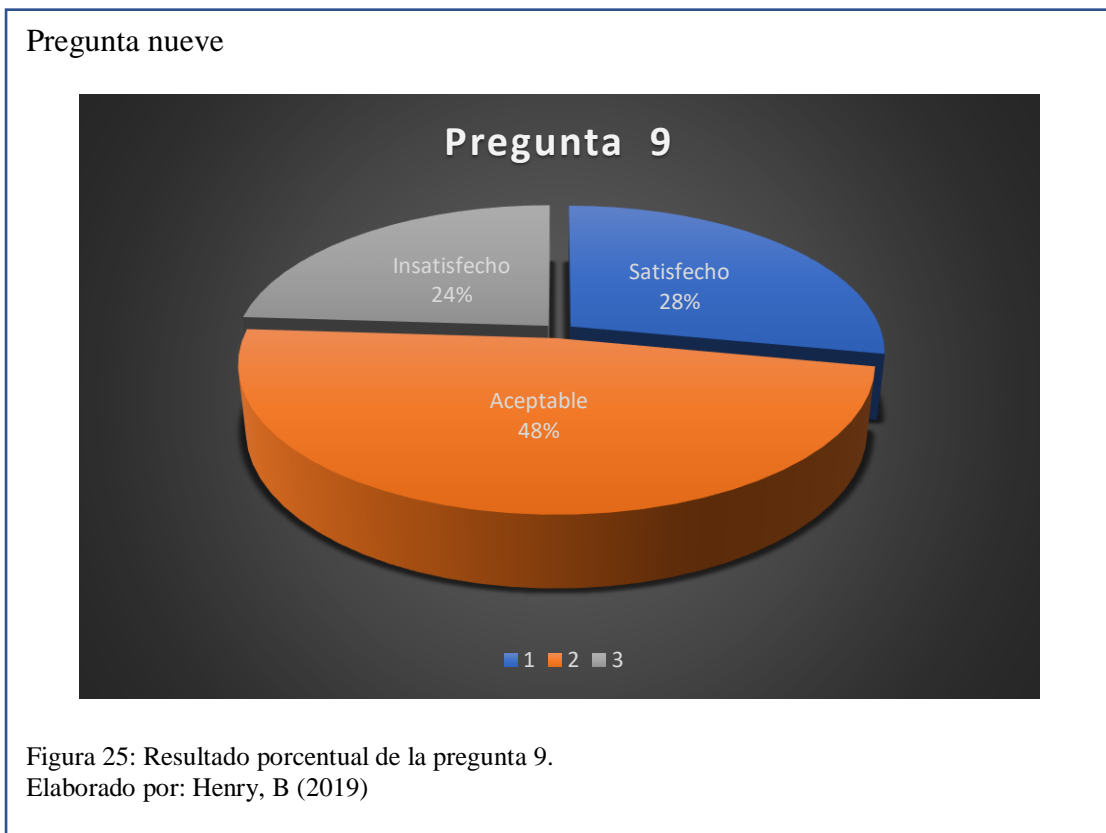


En la representación porcentual del resultado de la pregunta 8 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 96% de aceptación es considerado que a los usuarios encuestados les pareció interesante la aplicación móvil, para los 24 usuarios que representa el 96%, y 1 usuarios que representan el 4% que no le pareció interesante.

Tabla 17: Resultados la pregunta 9.

Pregunta número 9			
Descripción:	9. ¿La aplicación web es fácil de usar?		
Número Encuestas	Resultado		
	Satisfecho	Aceptable	Insatisfecho
Muestreo	7	12	6

Nota: Esta tabla contiene el resultado de la pregunta 9

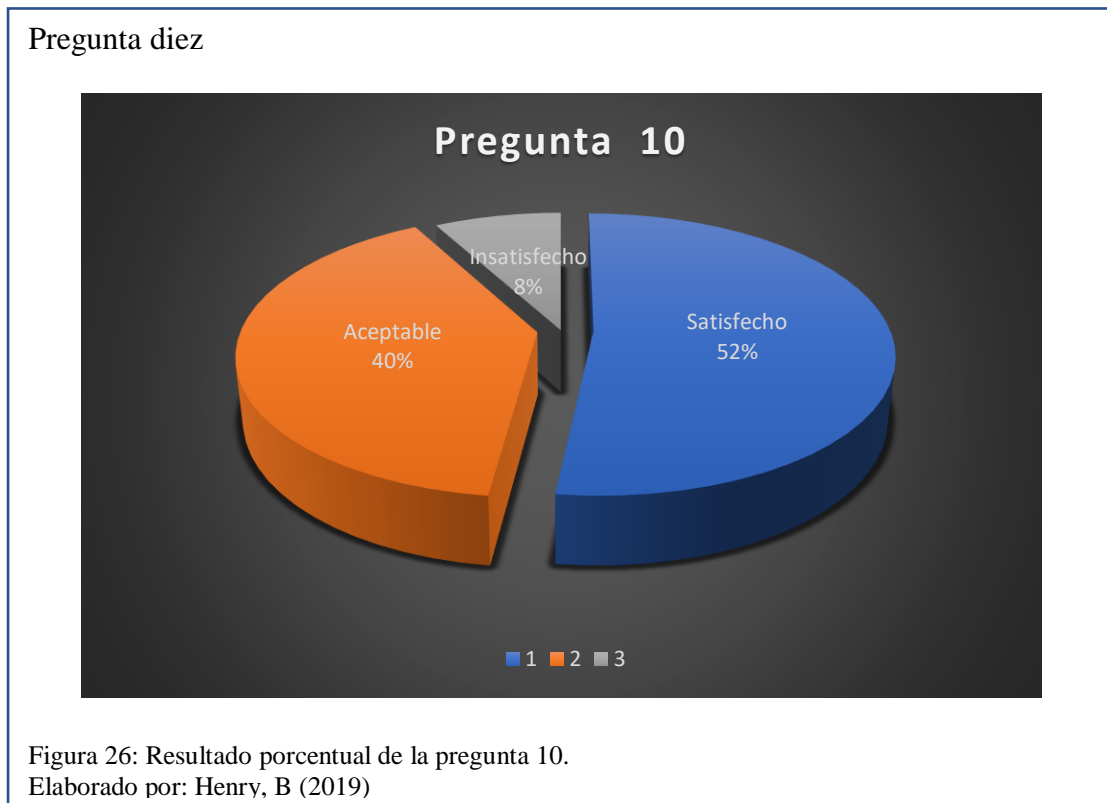


En la representación porcentual del resultado de la pregunta 9 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 48% de aceptación es considerado que a los usuarios encuestados les pareció fácil de usar la aplicación móvil, para los 12 usuarios que representa el 48%, 7 usuarios que representan el 28% están satisfechos, y para un 24% no les pareció fácil de usar.

Tabla 18: Resultados la pregunta 10.

Pregunta número 10			
Descripción:	10. ¿Cuál es la impresión general de la aplicación web?		
Número Encuestas	Resultado		
	Satisfecho	Aceptable	Insatisfecho
Muestreo	13	10	2

Nota: Esta tabla contiene el resultado de la pregunta 10



En la representación porcentual del resultado de la pregunta 10 que corresponde a la encuesta que fue realizada a 25 usuarios (personal administrativo), se evidencia que con un 52% están satisfechos considerado que a los usuarios encuestados tiene una buena impresión de la aplicación móvil, para los 13 usuarios que representa el 52%, 10 usuarios que representan el 40% les pareció aceptable, y para un 8% no tienen una buena impresión.

3.5.1.3 Pruebas de rendimiento

La finalidad de realizar las pruebas de rendimiento es asegurar que los tiempos de respuestas al realizar la conexión con los servicios sean óptimas, haciendo uso de la herramienta java (JMeter), que permite hacer pruebas de carga y estrés.

Las pruebas realizadas fueron enfocadas en peticiones GET, al estudiar el segmento que hará uso de la aplicación se diseñó los distintos escenarios para las pruebas.

3.5.1.3.1 Pruebas de carga

Escenario 1.

Se desarrolla el escenario uno, teniendo como referencia 10 usuarios en un rango de 5 segundos que será el tiempo que se demora en hacer la peticiones.

Los resultados para el escenario uno realizado con los 10 usuarios son satisfactorias ya que los tiempos de respuesta son aceptables, dentro de los 5 segundos se evidencia que los intervalos son (200 ms) entre cada petición y el estatus es positivo, el consumo de bytes enviados son 247 constantes, las respuestas de servidor 108 a 212 ms, el porcentaje de error es de 0.00%.

Escenario 2.

Se desarrolla el escenario dos, teniendo como referencia 50 usuarios en un rango de 5 segundos que será el tiempo que se demora en hacer la peticiones.

Los resultados para el escenario 2 son satisfactorias ya que los tiempos de respuesta son aceptables, dentro de los 5 segundos se evidencia que los intervalos son (300 ms) entre cada petición y el estatus es positivo, el consumo de bytes enviados son 247 constantes, las respuestas de servidor 115 a 286 ms, el porcentaje de error es de 0.00%.

3.5.1.3.2 Pruebas de rendimiento aplicación móvil

El objetivo para realizar las pruebas a la aplicación móvil es para poder evidenciar el rendimiento y consumo de recursos en los teléfonos inteligentes que son más usados por los

choferes, haciendo uso de aplicaciones móviles Tune-up kit, esta aplicación permite medir uso de recursos de las aplicaciones instaladas en los teléfonos inteligentes, se realizan pruebas completas de uso de la aplicación móvil.

Se seleccionan 2 celulares para realizar las pruebas de rendimiento.

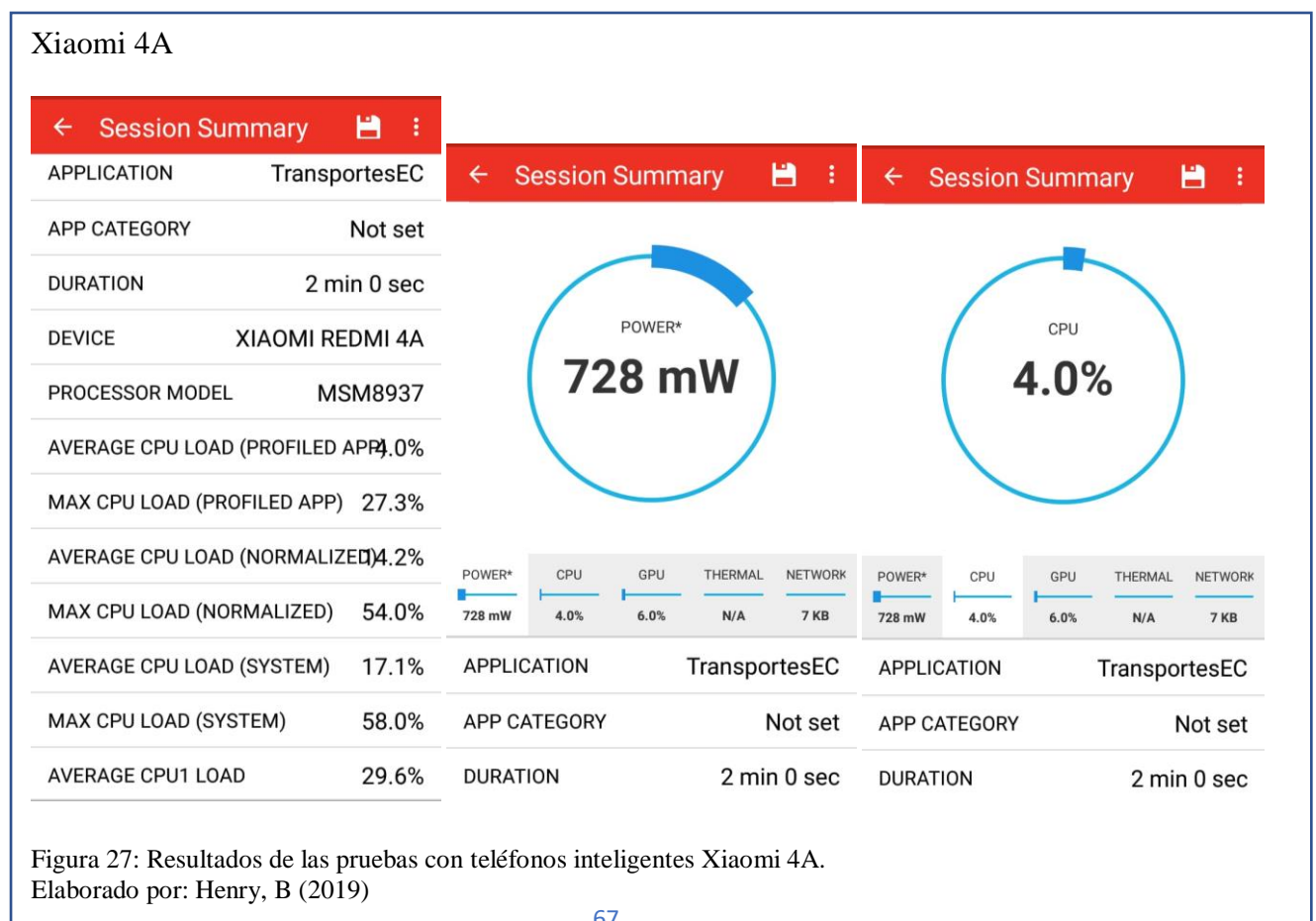
Tabla 19: Teléfonos inteligentes usados para las pruebas.

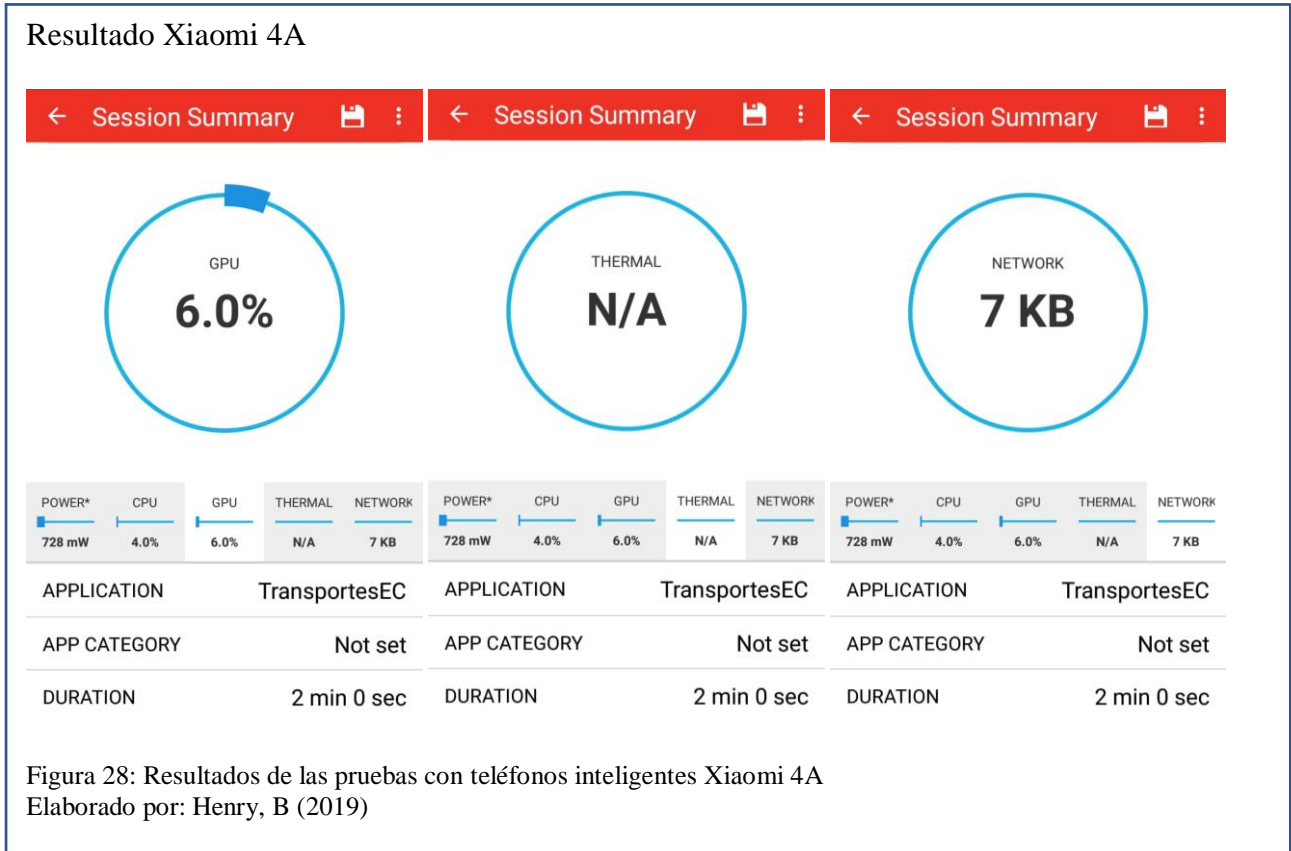
Teléfonos inteligentes			
Marca	Modelo	Sistema Operativo	Memoria RAM
Xiaomi	A4	Android 6	2 GB
Samsung Galaxy	S5	Android 8	2 GB

Nota: Esta tabla contiene los teléfonos inteligentes que se usaran para las pruebas

Xiaomi A4

Para realizar las pruebas se hace uso de la aplicación móvil Tune-up kit con la cual se realiza un testeo de la aplicación móvil programando 2 minutos para poder verificar el consumo de recursos, para ello se plantea hacer que la aplicación móvil realice todas sus funciones (ingresar usuario, seleccionar hora de viaje, escanear cédula, pausar viaje, continuar con escaneo de cédula, guardar la lista de pasajeros).



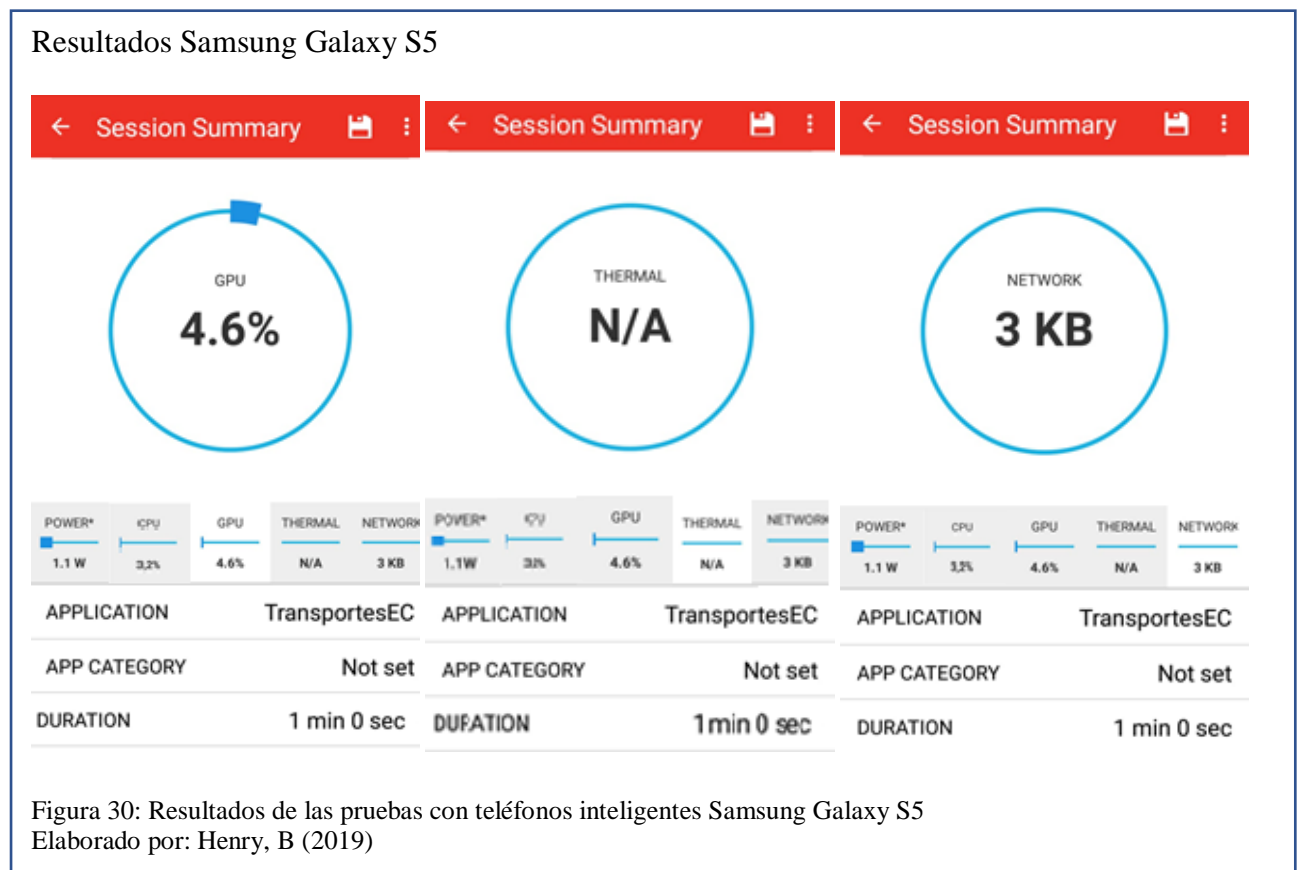
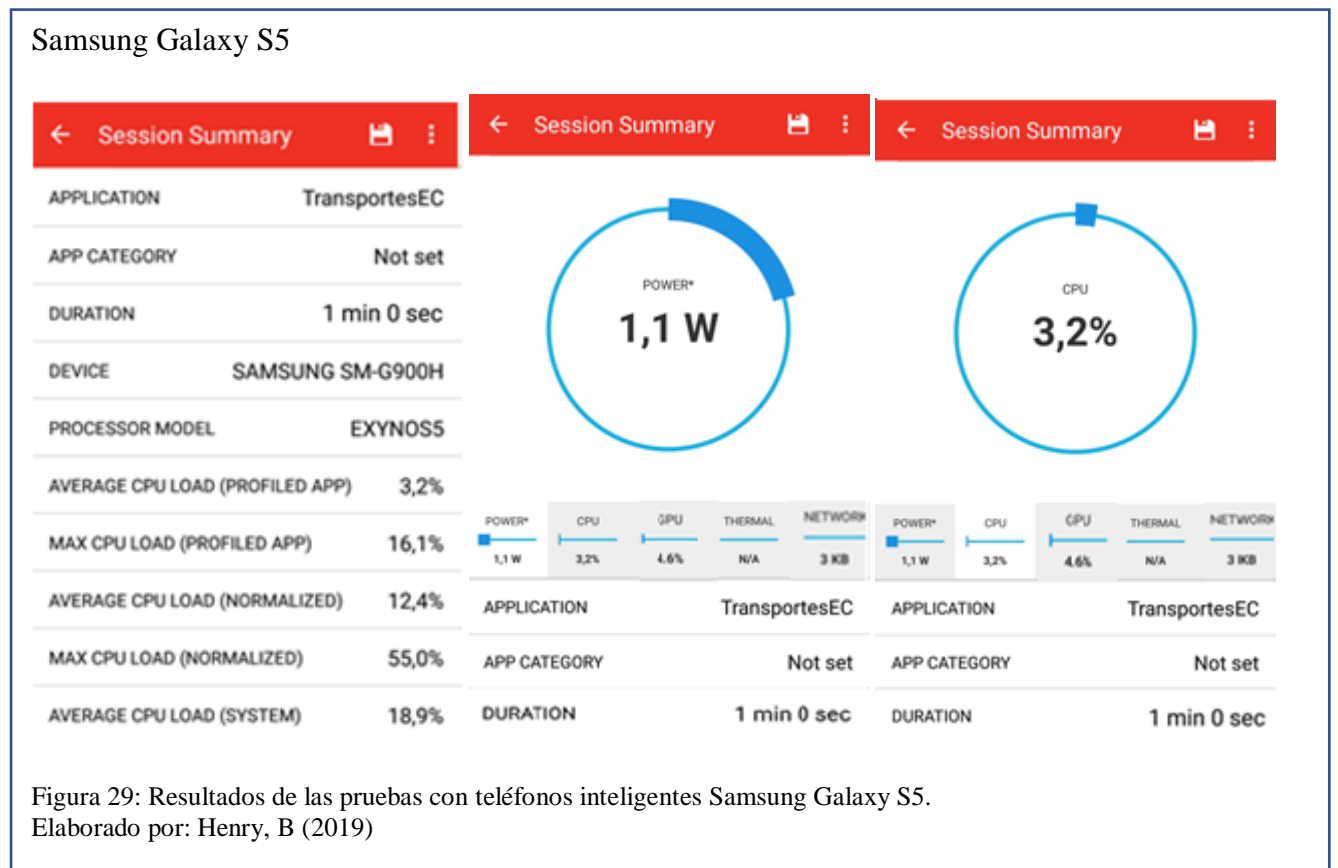


Se evidencia que los recursos que consume la aplicación móvil no afecta al rendimiento de los teléfonos inteligentes, consume 728 MW de batería por cada 2 minutos de funcionamiento, el consumo de batería no es elevado, usa un 4% de la memoria física y 6% de memoria gráfica lo cual garantiza el funcionamiento normal de los teléfonos inteligentes, para los minutos de funcionamiento garantiza que la temperatura es de 0% , el consumo de datos es de 7 KB, con todos los resultados en evidencia se puede garantizar que el consumo de recursos es bajo de la aplicación móvil.

Samsung Galaxy S5

Para realizar las pruebas se hace uso de la aplicación móvil Tune-up kit con la cual se realiza un testeo de la aplicación móvil programando 1 minutos para poder verificar el consumo de recursos, para ello se plantea hacer que la aplicación móvil realicé todas sus

funciones (ingresar usuario, seleccionar hora de viaje, escanear cédula, pausar viaje, continuar con escaneo de cédula, guardar la lista de pasajeros).

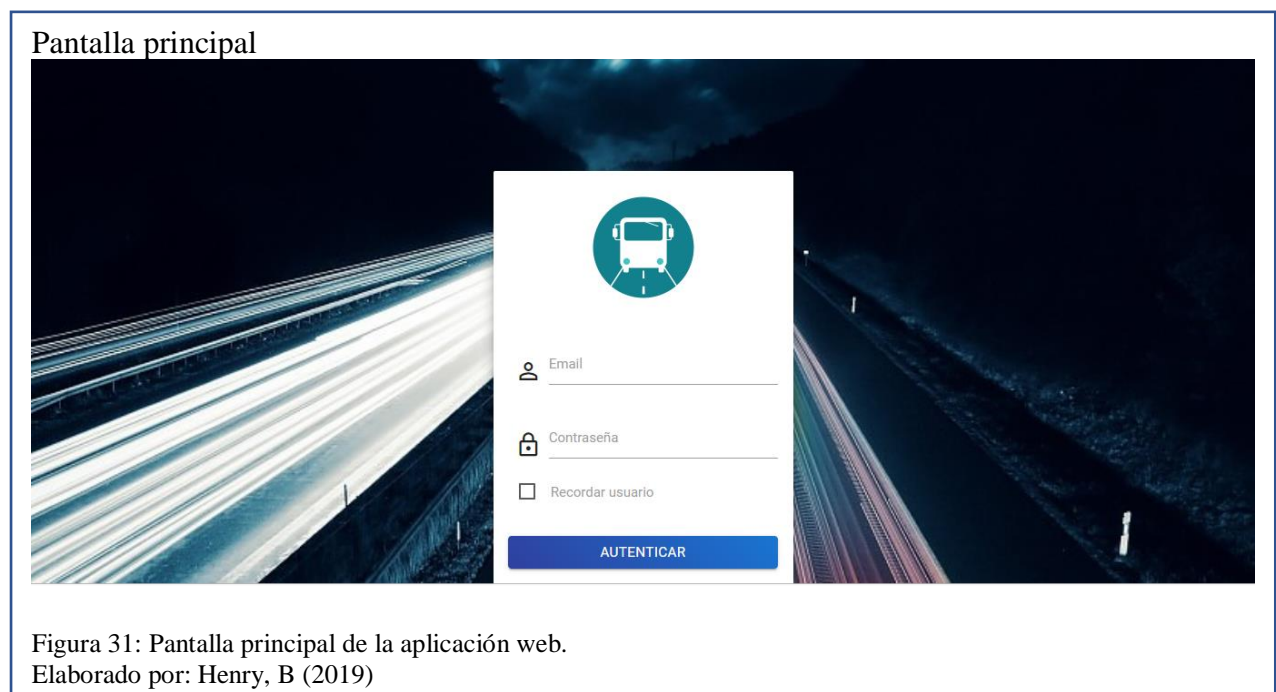


Se evidencia que los recursos que consume la aplicación móvil no afecta al rendimiento de los teléfonos inteligentes, consume 1,1 W de batería por cada 1 minutos de funcionamiento, el consumo de batería es mínimo, usa un 3,2% de la memoria física y 4,6% de memoria gráfica lo cual garantiza el funcionamiento normal de los teléfonos inteligentes, para los minutos de funcionamiento garantiza que la temperatura es de 0%, el consumo de datos es de 3 KB, con todos los resultados en evidencia se puede garantizar que el consumo de recursos es bajo de la aplicación móvil.

3.6 Perspectiva visual del software

3.6.1 Perspectiva visual de la aplicación web

Se realiza una breve representación de la ampliación web, donde los resultados finales obtenidos después de que la aplicación sea sometida a las distintas pruebas y se pueda tener una garantía de que la misma es funcional y aceptado por el segmento de usuarios que fueron presentados.



El diseño minimalista para el login para poder obtener una excelente percepción y que la experiencia del usuario sea agradable, la imagen fue seleccionada cuidadosamente para que tenga un contraste acertado con lo que comunica la aplicación.

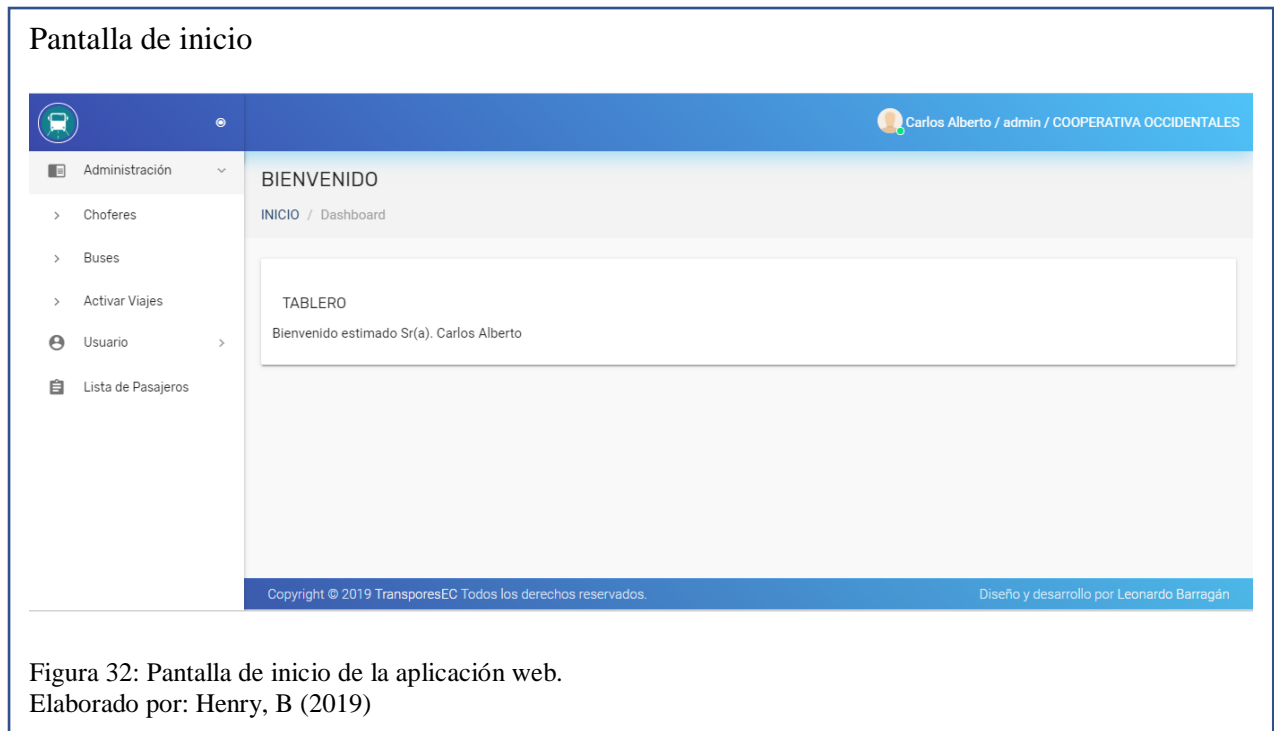


Figura 32: Pantalla de inicio de la aplicación web.
Elaborado por: Henry, B (2019)

La pantalla de inicio fue pensada de una modo simple y llamativo para que los usuarios puedan hacer uso de la ampliación web de una forma fácil y eficiente, lo cual garantiza la optimización de recursos en tiempo hombres, la pantalla de inicio consta de un menú (Choferes, Buses, Activar Viajes, Lista de pasajeros) que está enfocado en la funcionalidad de cada componente expuesto en los requerimientos.

Crear choferes

ROL DEL USUARIO ASIGNADO
Seleccione un Rol
Chofer

NUEVO CHOFER

Cédula Nombres Apellidos

Email Contraseña Fecha de Nacimiento

Dirección Domicilio Teléfono Celular

Tipo de Licencia Cooperativa
SIN ASIGNAR COOPERATIVA OCCIDENTALES

GUARDAR

Figura 33: Formulario para crear choferes.
Elaborado por: Henry, B (2019)

Los formularios fueron pensados para poder llenarlos de la forma más precisa y eficiente, los campos (cédula, nombres, apellidos, email, contraseña, fecha de nacimiento, dirección de domicilio, teléfono celular, tipo de licencia, cooperativa), están correctamente validados para que no exista algún error y demore el trabajo a realizar.

Descargar lista de pasajeros

Cooperativa
COOPERATIVA TRANSPORTES PANAMERICANA

Ruta
QUITO - ATACAMES (10:30:00)

Fecha de Salida

BUSCAR LISTADO

Pasajeros Registrados

COOPERATIVA TRANSPORTES PANAMERICANA
CHOFER: 1243567891 - Fabricio Cristina Cruz Aguilar
BUS: Hino Fast (2014) - PLACA: PYH0157 | NRO UNIDAD: 02
RUTA: QUITO - ATACAMES HORA: 10:30:00 FECHA DE SALIDA: 2019-05-14

DESCARGAR

Cédula	Nombres y Apellidos	Fecha de Nacimiento	Hora
1708705536	RAUL HERNAN BARRAGAN PACHECO	15-05-1965	21:47:38
1721867891	HENRY LEONARDO BARRAGAN PEREZ	22-12-1988	21:47:38

Mostrando 1 a 2 de 2 registros

Anterior 1 Siguiente

Figura 34: Consular y descargar lista de pasajeros.
Elaborado por: Henry, B (2019)

La lista de pasajeros es una de las partes más fundamentales de la aplicación web, dicha funcionalidad permite llevar el control total de los viajes y poder descargar la lista de pasajeros, filtrándolos por fechas, permite tener una previsualización de la lista para optimizar el tiempo, para no descargar una lista innecesaria.

El manual de aplicación web se encuentra en el anexo 1.

3.6.2 Perspectiva visual de la aplicación móvil

Se realiza una breve representación visual de la ampliación móvil, donde los resultados finales obtenidos después de que la aplicación pasa los controles de calidad y se evalué que su funcionalidad es adecuada para su fin, y que asegura que la misma que cumple con la satisfacción de usuarios que a los fueron presentados.

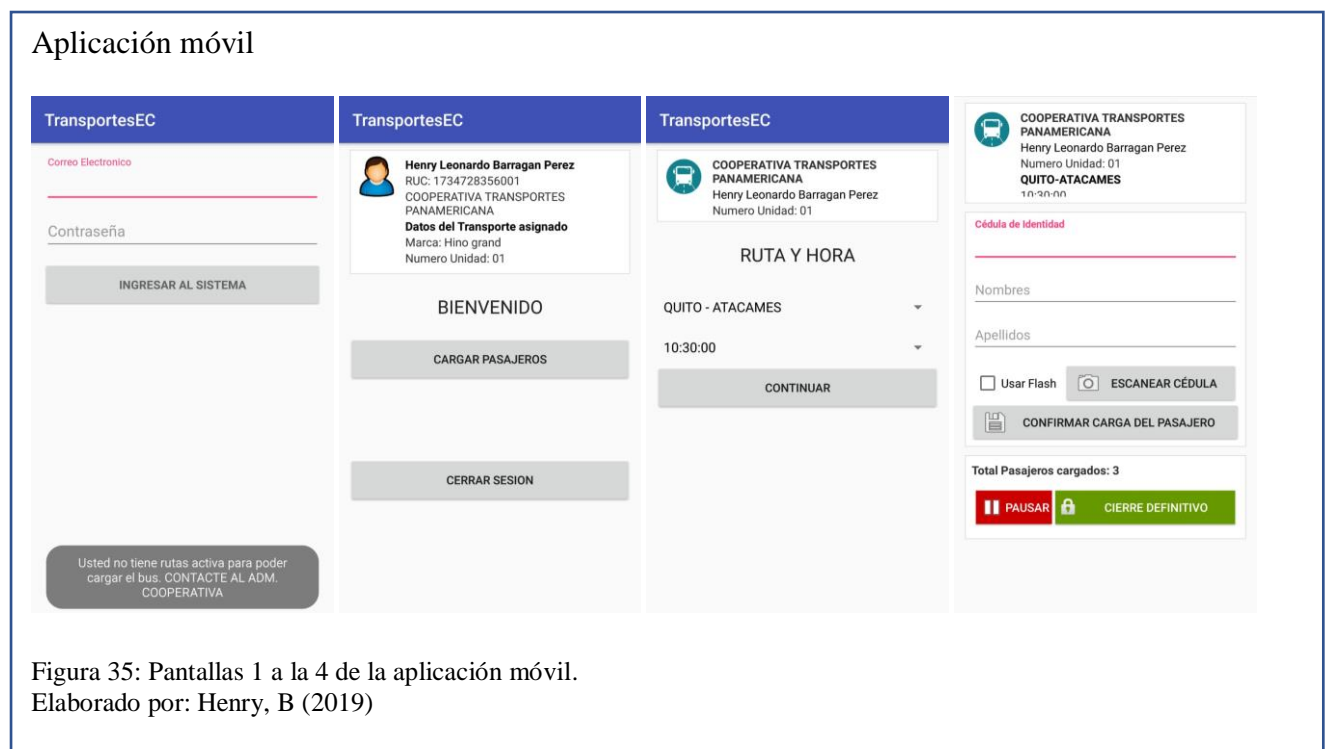


Figura 35: Pantallas 1 a la 4 de la aplicación móvil.
Elaborado por: Henry, B (2019)

Aplicación móvil

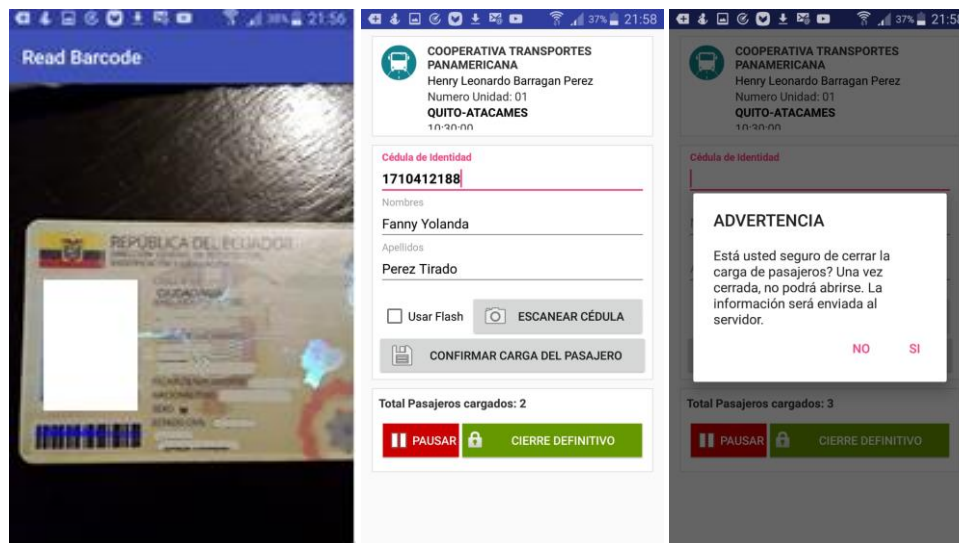


Figura 36: Escaneo de la cédula.
Elaborado por: Henry, B (2019)

El diseño de la aplicación móvil es los más simple y manejable posible para que los usuarios (choferes) puedan hacer uso de ella sin ninguna complicación y no se confundan, en la figura No. 35 se puede evidenciar en la pantalla uno el (login), en la dos (la información del chofer), en la tres (selección de usuario), en la cuatro (el formulario donde se cargarla la información de los pasajeros al escanear la cédula). En la figura No. 36 se puede evidenciar en la pantalla uno (proceso de escaneo de la cédula) en la dos (ingreso y autocompletado manual de la cédula) en la tres (la confirmación del cierre de carga de pasajeros).

El manual de aplicación móvil se encuentra en el anexo 1.

CONCLUSIONES

- Con la implementación de la aplicación web se evidencia que el proceso recomendado por la normativa (RESOLUCIÓN N° 161-DIR-2013-ANT), ha sido automatizado y se ejecuta de una manera ágil y segura.
- La identificación y documentación de procesos posibilita la implementación de los requerimientos del software de una forma eficiente.
- Con la automatización de la lista de pasajeros se genera un beneficio para los involucrados, para los pasajeros se garantiza seguridad, para las autoridades se pone a disposición una herramienta fiable y oportuna para responder adecuadamente a cualquier requerimiento de información.

RECOMENDACIONES

- Frente al desafío del desarrollo de un software, se deben identificar todos procesos manuales que puedan ser susceptibles de automatización para mejorar la eficiencia en el proceso de ingeniería de software.
- Es necesario desarrollar una segunda etapa del software, que tenga como fin incrementar nuevas funcionalidades para que las aplicaciones puedan ser usadas por todas las cooperativas en todas sus rutas, por ejemplo, el módulo para crear y administrar las rutas con sus horarios.
- En la segunda etapa se tendrá que evaluar el crecimiento del software, según las necesidades de momento y con ello se debe considerar la migración de la aplicación a un servidor más robusto, con mayor seguridad, espacio de almacenamiento, y una mejor capacidad de respuesta.
- Se tiene que diseñar un plan de capacitación cuando el software salga a producción, dicho plan tiene que estar dividido en dos etapas, en la etapa uno se debe preparar una reunión con el personal administrativo que fungirá como administrador de la cooperativa, en la etapa dos se debe preparar algunas reuniones, coordinar con el personal administrativo, para indicar el funcionamiento de la aplicación móvil a los controladores; esta capacitación tiene que ser lo más simple posible, sin usar lenguaje técnico, por las características de los usuarios.

LISTA DE REFERENCIAS

Agencia Nacional de Tránsito, D. de secretaria general. (2013). *RESOLUCIÓN No 161-DIR-2013-ANT*.

android.pdf. (s. f.). Recuperado de

https://s3.amazonaws.com/academia.edu.documents/34556195/android.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1558117490&Signature=4bPkjpLOaMJom5wocUxMWjt2ESM%3D&response-content-disposition=inline%3B%20filename%3DG_-Te_C_Introduccion_a_Android.pdf

ANTECEDENTES DE LA AUTOMATIZACION. (s. f.). Recuperado 1 de junio de 2019, de

prezi.com website: https://prezi.com/p1ijlnq_dqxc/antecedentes-de-la-automatizacion/

APLICACIONES WEB. (s. f.). Recuperado de

https://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf

Arias, M. Á. (2017). *Aprende Programación Web con PHP y MySQL: 2ª Edición*. IT Campus Academy.

BENBOURAHLA, N. (2015). *Android 5: Principios del desarrollo de aplicaciones Java*. Ediciones ENI.

BONE VELÀQUEZ, I. C., CASTILLO AMPUDIA, G. E., & SANDOVAL EGAS, J. C.

(2015). *ANÁLISIS DE LA RESPONSABILIDAD SOCIAL EN LAS EMPRESAS DE TRANSPORTE PÚBLICO INTERPROVINCIAL DEL ECUADOR, CASO*

“COOPERATIVA DE TRANSPORTES OCCIDENTALES” (TESIS, UNIVERSIDAD POLITÈCNIA SALESIANA). Recuperado de

<https://dspace.ups.edu.ec/bitstream/123456789/10134/1/UPS%20-%20ST001687.pdf>

BORJA LÓPEZ, Y. (s. f.). *Metodología Ágil de Desarrollo de Software – XP*. Recuperado de DOCUMENTO.

- Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (2017). Restful API Architecture Based on Laravel Framework. *Journal of Physics: Conference Series*, 910, 012016.
<https://doi.org/10.1088/1742-6596/910/1/012016>
- Extreme Programming: A Gentle Introduction. (s. f.). Recuperado 16 de mayo de 2019, de <http://www.extremeprogramming.org/>
- José Joskowicz. (2008, febrero 10). *Reglas y Prácticas en eXtreme Programming*. Recuperado de Department of Computer Science. (University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2)
- Juan, C. S. (1993). *La Revolución industrial*. Ediciones AKAL.
- Laravel - EcuRed. (s. f.). Recuperado 20 de mayo de 2019, de <https://www.ecured.cu/Laravel>
- Metodologías del Desarrollo de Software. (2016, octubre 18). Recuperado 17 de mayo de 2019, de OK HOSTING | Hospedaje Web, Dominios, Desarrollo de Software, Marketing Online, SEO website: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>
- Pantoja, E. B. (2004a). *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. 2, 15.
- Pantoja, E. B. (2004b). *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. 2, 15.
- PSR-2: Coding Style Guide - PHP-FIG. (s. f.). Recuperado 4 de mayo de 2019, de <https://www.php-fig.org/psr/psr-2/>
- Qué es Api Rest y por qué debes de integrarla en tu negocio ¡Descúbrelo! (2018, mayo 29). Recuperado 20 de mayo de 2019, de Blog de IEBSchool website: <https://www.iebschool.com/blog/que-es-api-rest-integrar-negocio-business-tech/>

¿Qué son las bases de datos? (2007, octubre 26). Recuperado 20 de mayo de 2019, de Maestros del Web website: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

Tinoco, E. E. C., & Solís, I. S. (2014). *Programación Web con CSS, JavaScript, PHP y AJAX*. Iván Soria Solís.