

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Tesis

**Desarrollo de un aplicativo móvil para el registro
del cumplimiento y desempeño de líneas de
transporte público en la ciudad de Lima**

Eduardo Ydrogo Ramírez

Para optar el Título Profesional de
Ingeniero de Sistemas e Informática

Huancayo, 2020

Repositorio Institucional Continental
Tesis digital



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

AGRADECIMIENTO

Mediante estas líneas expreso mi profundo y sincero agradecimiento a la universidad por contribuir en mi formación profesional y de manera especial a los docentes ingenieros que contribuyeron en la realización de este trabajo, a ellos mi reconocimiento y gratitud.

Además, agradecer a las personas que en el ámbito laboral confiaron en mi y a través de sus acciones tomar decisiones correctas con profesionalismo.

Eduardo Ydrogo Ramírez

DEDICATORIA

“A mis padres, por enseñarme con el ejemplo a superarme y a seguir creciendo cada día como persona, a mi familia que siempre están brindándome su apoyo y a todos los que confiaron en que culminaría este proyecto de vida”

Eduardo Ydrogo Ramírez

ÍNDICE

| | |
|---|----|
| AGRADECIMIENTO..... | i |
| DEDICATORIA..... | ii |
| LISTA DE TABLAS..... | 7 |
| LISTA DE FIGURAS..... | 9 |
| LISTA DE ANEXOS..... | 11 |
| RESUMEN | 12 |
| ABSTRACT | 13 |
| INTRODUCCIÓN | 14 |
| CAPÍTULO I | 15 |
| Planteamiento Del Estudio | 15 |
| 1.1 Planteamiento y formulación del problema..... | 15 |
| 1.1.1 Problema general..... | 20 |
| 1.1.2 Problemas específicos | 20 |
| 1.2 Objetivos | 21 |
| 1.2.1 Objetivo general | 21 |
| 1.2.2 Objetivos específicos | 21 |
| 1.3 Justificación e importancia | 22 |
| 1.3.1 Justificación teórica..... | 22 |
| 1.3.2 Justificación práctica | 22 |
| 1.3.3 Justificación social..... | 23 |
| 1.3.4 Importancia a nivel social | 23 |
| 1.3.5 Importancia a nivel tecnológico | 24 |
| 1.4 Alcances y Limitaciones | 26 |
| 1.4.1 Alcances | 26 |
| 1.4.2 Limitaciones | 26 |
| CAPÍTULO II: | 27 |
| MARCO TEÓRICO..... | 27 |
| 2.1 Antecedentes del problema..... | 27 |
| 2.1.1 Antecedentes Internacionales | 27 |
| 2.2.2 Antecedentes Nacionales..... | 28 |
| 2.3 Bases teóricas..... | 29 |

| | | |
|-------|--|----|
| 2.1.1 | Otorgamiento de concesiones | 36 |
| 2.1.2 | Proceso de licitación de ruta | 36 |
| 2.2.2 | Proceso de autorización de servicio | 38 |
| 2.2.3 | Plazo de vigencia | 40 |
| 2.2.4 | Renovación de autorización | 40 |
| 2.2.5 | Condiciones de prestación del servicio de transporte público..... | 40 |
| 2.2.6 | Prestación del servicio de transporte público..... | 43 |
| 2.2.7 | Proceso de suspensión a la autorización de servicio..... | 44 |
| 2.4 | Definición de términos básicos | 45 |
| a) | El servicio de transporte público urbano de personas | 45 |
| b) | Dirección general de circulación terrestre | 45 |
| c) | Transporte público terrestre | 45 |
| d) | Servicio de transporte de pasajeros terrestres en Lima Metropolitana | 45 |
| e) | GTU-Gerencia de transporte urbano en Lima Metropolitana..... | 46 |
| f) | Empresa de transporte | 46 |
| g) | Administración de rutas en el transporte público..... | 46 |
| h) | Rutas | 47 |
| i) | Tráfico..... | 47 |
| j) | Frecuencia..... | 47 |
| k) | Paradero..... | 47 |
| l) | Pasajero | 48 |
| m) | Administración de horarios de transporte público..... | 48 |
| n) | Aplicativo móvil | 48 |
| o) | Tipos de aplicaciones web | 48 |
| p) | Usabilidad y funcionalidad | 49 |
| q) | Desarrollo de software predictivo | 50 |
| r) | Desarrollo de software ágil..... | 50 |
| s) | Base de datos..... | 50 |
| t) | Diseño responsivo | 51 |
| u) | Scrum | 51 |
| v) | Product backlog:..... | 52 |
| w) | Sprint backlog:..... | 52 |
| x) | Framework..... | 53 |

| | |
|---|----|
| y) Experiencia de usuario - UX | 53 |
| z) Efectividad, eficiencia y satisfacción | 54 |
| CAPÍTULO III: | 55 |
| METODOLOGÍA..... | 55 |
| 3.1 Metodología para la gestión del proyecto | 55 |
| 3.1.1 Grupos de procesos básicos | 55 |
| 3.1.2 Áreas de conocimiento del proyecto..... | 56 |
| 3.1.3 Grupo de proceso de inicio..... | 57 |
| 3.1.4 Grupo de proceso de planificación | 58 |
| 3.2 Metodología aplicada para el desarrollo de software (Scrum) | 58 |
| 3.2.1 Componente Scrum | 59 |
| 3.3 Población y muestra..... | 64 |
| CAPÍTULO IV: | 65 |
| ANÁLISIS Y DISEÑO DE LA SOLUCIÓN..... | 65 |
| 4.1 Identificación de requerimientos | 65 |
| 4.1.1 Planificación del proyecto..... | 65 |
| 4.1.2 Cronograma del proyecto | 67 |
| 4.1.3 Estructura desglosable del trabajo | 68 |
| 4.2 Alcance de la solución..... | 69 |
| 4.3 Etapas en el proceso de construcción | 69 |
| 4.4 Requerimientos no funcionales | 70 |
| 4.5 Historias de usuarios..... | 71 |
| 4.6 Planificación de Sprints | 75 |
| 4.7 Tablero de seguimiento | 77 |
| 4.8 Análisis de la solución | 78 |
| 4.9 Análisis tecnológico..... | 84 |
| 4.10 Diseño de la solución..... | 84 |
| 4.10.1 Arquitectura de la Solución | 84 |
| 4.11 Mapa de navegación del aplicativo móvil..... | 85 |
| 4.12 Plataforma de base de datos | 88 |
| 4.13 Diseño de la de base de datos..... | 90 |
| 4.14 Diseño de las tablas de almacenamiento de base de datos..... | 91 |
| 4.15 Diccionario de datos | 94 |

| | | |
|----------------------------------|--|-----|
| 4.16 | Desarrollo de los procedimientos almacenados de la base de datos | 97 |
| CAPÍTULO V:..... | | 100 |
| CONSTRUCCIÓN | | 100 |
| 5.1 | Plataforma de desarrollo | 100 |
| 5.1.1 | Descripción del aplicativo móvil | 100 |
| 5.1.2 | Pantallas del aplicativo móvil | 101 |
| 5.2 | Componentes usados para el desarrollo | 107 |
| 5.3 | Desarrollo del aplicativo móvil | 108 |
| 5.4 | Visualización del funcionamiento del aplicativo | 109 |
| 5.5 | Pruebas y resultados..... | 109 |
| 5.5.1 | Pruebas de funcionalidad, usabilidad y utilidad. | 110 |
| 5.5.2 | Resultado de pruebas de funcionalidad, usabilidad y utilidad..... | 112 |
| 5.5.3 | Prueba de rendimiento del aplicativo..... | 119 |
| 5.5.4 | Resultado de la prueba de rendimiento del aplicativo..... | 120 |
| 5.5.5 | Prueba de base de datos | 121 |
| 5.5.6 | Resultado de prueba de la base de datos | 124 |
| 5.5.7 | Pruebas de caja negra | 126 |
| 5.6 | Resultado de las pruebas al aplicativo | 135 |
| 5.7 | Otros resultados obtenidos durante el proceso fueron los siguientes: | 140 |
| 5.8 | Comparación de los resultados obtenidos | 140 |
| CONCLUSIONES..... | | 142 |
| TRABAJOS FUTUROS | | 144 |
| REFERENCIAS BIBLIOGRÁFICAS | | 145 |
| ANEXOS | | 149 |
| CONFORMIDADES..... | | 208 |

LISTA DE TABLAS

| | |
|--|-----|
| Tabla 1 Percepción sobre el tránsito en Lima..... | 16 |
| Tabla 2 Principales Problemas que Enfrenta el Usuario Por el Transporte Público. | 17 |
| Tabla 3 Servicios Apps relacionados al Transporte Público en Perú. | 25 |
| Tabla 4 Ejemplo de Sprint Backlog..... | 53 |
| Tabla 5 Matriz de Comparación de Metodologías..... | 58 |
| Tabla 6 Ejemplo de Product Backlog..... | 62 |
| Tabla 7: Acta del Proyecto..... | 65 |
| Tabla 8 Plan de Actividades del Proyecto..... | 67 |
| Tabla 9 Fases de la Metodología Ágil..... | 69 |
| Tabla 10 Requerimientos No Funcionales | 71 |
| Tabla 11 Historias de Usuario para el aplicativo móvil..... | 71 |
| Tabla 12 Sprint Planning | 75 |
| Tabla 13 Diseño de Pantalla - Interface Registrar Usuario al Aplicativo..... | 78 |
| Tabla 14 Diseño de Pantalla - Interface Recuperar Datos de usuario..... | 79 |
| Tabla 15 Diseño de Pantalla - Interface de Ingreso al Aplicativo | 79 |
| Tabla 16 Diseño de Pantalla - Interface para Visualizar Opciones para el Usuario..... | 80 |
| Tabla 17 Diseño de Pantalla - Interface para Registrar Incidencia..... | 80 |
| Tabla 18 Diseño de Pantalla - Interface para Visualizar Información de Empresas. | 81 |
| Tabla 19 Diseño de Pantalla - Interface para que los Fiscalizadores Visualicen Información | 81 |
| Tabla 20 Diseño de Pantalla - Interface para Registrar Usuario Fiscalizador | 82 |
| Tabla 21 Diseño de Pantalla - Interface para Registrar Empresa de Transporte | 82 |
| Tabla 22 Diseño de Pantalla - Interface para Registrar Buses..... | 83 |
| Tabla 23 Diseño de Pantalla - Interface para Registrar la descripción de problemas..... | 83 |
| Tabla 24 Especificación de Hardware | 84 |
| Tabla 25 Especificación de Software..... | 84 |
| Tabla 26 Diseño de Tablas de Base de Datos..... | 91 |
| Tabla 27 Diccionario de Datos..... | 94 |
| Tabla 28: Listado de Procedimientos Almacenados | 98 |
| Tabla 29 Pantallas del Aplicativo Móvil..... | 101 |
| Tabla 30 Resumen de Pruebas por Historia de Usuario | 107 |
| Tabla 31 Encuesta aplicada a los usuarios del servicio de transporte Público..... | 111 |

| | |
|---|-----|
| Tabla 32 Pruebas de caja negra para el procedimiento: Registrar Nuevo Usuario | 126 |
| Tabla 33 Pruebas de caja negra para el procedimiento: Recuperar datos de usuario. | 128 |
| Tabla 34 Pruebas de caja negra para el procedimiento: Registrar Incidencias | 128 |
| Tabla 35 Pruebas de caja negra para el procedimiento: Visualizar Información de Bus. | 129 |
| Tabla 36 Pruebas de caja negra del procedimiento: Visualizar Información de Empresas . | 131 |
| Tabla 37 Pruebas de caja negra para el procedimiento: Registrar Usuario Fiscalizador..... | 131 |
| Tabla 38 Pruebas de caja negra para el procedimiento: Registrar Empresa de Transporte | 133 |
| Tabla 39 Resumen de Pruebas Realizadas..... | 136 |
| Tabla 40 Resultado de las Pruebas No Funcionales | 138 |

LISTA DE FIGURAS

| | |
|--|-----|
| Figura 1 Estadística Población Urbana vs Rural..... | 32 |
| Figura 2 Medios de transporte público en Lima | 33 |
| Figura 3 Tiempos estimados por el Ciudadano en Llegar a Destino | 34 |
| Figura 4 Estadística sobre las Preferencias de los Ciudadanos | 35 |
| Figura 5 Características del Diseño Responsivo | 51 |
| Figura 6 Proceso Scrum Completo..... | 52 |
| Figura 7 Ciclo de vida del grupo de Procesos según el PMBOK..... | 57 |
| Figura 8 Ciclo de vida del desarrollo de Software – Metodología Scrum | 59 |
| Figura 9 Ejemplo de Historia de Usuario | 64 |
| Figura 10 Estructura Desglosable del Trabajo..... | 68 |
| Figura 11 Tablero Kanban..... | 77 |
| Figura 12 Componentes de la Solución..... | 85 |
| Figura 13 Mapa de Navegación del Aplicativo Móvil..... | 86 |
| Figura 14 Mapa de Navegación del Aplicativo Móvil - Sección Configuración | 87 |
| Figura 15 Ubicación de la Base de Datos..... | 88 |
| Figura 16 Plataforma de la Base de Datos | 89 |
| Figura 17 Herramienta de Conectividad a la Base de Datos..... | 89 |
| Figura 18 Estructura de Seguridad de Acceso a la Base de Datos..... | 90 |
| Figura 19 Modelo Relacional de la Base de Datos | 91 |
| Figura 20 Procedimiento Almacenado de Incidencias con Inicio de Transacción | 97 |
| Figura 21 Procedimiento Almacenado de Incidencias con Finalización de Transacción..... | 98 |
| Figura 22 Plataforma de Desarrollo Móvil..... | 100 |
| Figura 23 Código de Programación del Aplicativo Móvil | 108 |
| Figura 24 Emulador de Dispositivos Móviles Usado para el Desarrollo de la Solución | 109 |
| Figura 25 Encuesta - Pregunta 1.- Género | 112 |
| Figura 26 Encuesta - Pregunta 2: Tipo de Usuario | 113 |
| Figura 27 Encuesta – Pregunta 3 | 113 |
| Figura 28 Encuesta – Pregunta 4 | 114 |
| Figura 29 Encuesta – Pregunta 5 | 115 |
| Figura 30 Encuesta – Pregunta 6 | 115 |
| Figura 31 Encuesta – Pregunta 7 | 116 |

| | |
|--|-----|
| Figura 32 Encuesta – Pregunta 8 | 116 |
| Figura 33 Encuesta – Pregunta 9 | 117 |
| Figura 34 Encuesta – Pregunta 10 | 118 |
| Figura 35 Encuesta – Pregunta 11 | 119 |
| Figura 36 Configuración del Simulador de Pantalla Móvil | 119 |
| Figura 37 Prueba de Adecuación de Pantalla de Ingreso del Aplicativo Móvil | 120 |
| Figura 38 Resultado de Prueba de Rendimiento de Ejecución del Aplicativo | 121 |
| Figura 39 Ubicación de la Base de Datos | 121 |
| Figura 40 Prueba de Estrés para Procedimiento de Creación de Usuarios | 122 |
| Figura 41 Prueba de Estrés para Procedimiento de Consulta de Usuarios..... | 123 |
| Figura 42 Prueba de Estrés para el Registro de Incidencias | 124 |
| Figura 43 Resultado Estadístico de Uso de Base de Datos en el Servidor | 124 |
| Figura 44 Resultado de las Pruebas por Transacción Parte 1 | 125 |
| Figura 45 Resultado de las Pruebas por Transacción Parte 2 | 125 |

LISTA DE ANEXOS

| | |
|---|-----|
| Anexo A Estructura de desarrollo del aplicativo | 150 |
| Anexo B Código de Inicio del Aplicativo..... | 151 |
| Anexo C Código de Opciones del Pasajero | 155 |
| Anexo D Código de Opciones del Fiscalizador | 157 |
| Anexo E Código de Recuperar Clave | 160 |
| Anexo F Código de Registrar Incidencia de Pasajero..... | 163 |
| Anexo G Código de Reporte de Incidencia de Pasajero | 168 |
| Anexo H Código de Registrar Fiscalizador | 172 |
| Anexo I Código de Registrar Empresa..... | 176 |
| Anexo J Código de Registrar BUS..... | 179 |
| Anexo K Código de Registrar Problema | 184 |
| Anexo L Código de Reporte de Fiscalizador..... | 187 |
| Anexo M Procedimiento de Registro de Pasajero..... | 192 |
| Anexo N Procedimiento de Registro de Fiscalizador | 195 |
| Anexo O Procedimiento de Registro de Incidencias | 198 |
| Anexo P Procedimiento de Reporte de Incidencia | 201 |
| Anexo Q Procedimiento de Registro de Empresas..... | 202 |
| Anexo R Procedimiento de Registro de Buses | 204 |
| Anexo S Procedimiento de Registro de Problemas | 206 |
| Anexo T Procedimiento de Reporte del Fiscalizador | 207 |

RESUMEN

El sistema de transporte público en Lima tiene como objetivo brindar el servicio de traslado a las personas que lo requieren. Sin embargo, los usuarios de este servicio no contaban con una plataforma tecnológica en la cual pudieran registrar la calidad del servicio recibido, la ausencia de esta información generaba incertidumbre y desconfianza en los pasajeros al abordar una unidad de transporte. Esta Tesis tuvo como propósito desarrollar el prototipo de un aplicativo móvil que contribuya con la gestión del cumplimiento y desempeño de las líneas de transporte público en Lima, a través del registro de incidencias ocurridas durante la prestación del servicio. Para la estructuración de esta Tesis se usó la metodología del Project Management Institute, preparando los documentos de inicio del proyecto, cronograma de actividades y la estructura desglosable del trabajo. El desarrollo del aplicativo móvil se realizó utilizando la metodología ágil Scrum (nombre propio), con la finalidad de obtener avances incrementales en corto tiempo. Como resultado se logró desarrollar un aplicativo móvil que registró información las incidencias ocurridas durante la prestación del servicio de transporte urbano de pasajeros en la ciudad de Lima, el 91% de los usuarios encuestados logró registrar satisfactoriamente las incidencias ocurridas durante la prestación del servicio, así mismo al 94% se les facilitó la lectura de las incidencias registrada, el 77% encontró útil la información mostrada por el aplicativo móvil y para 70% la información podría ser insumo para la evaluación del servicio.

Palabras claves: transporte público, aplicativo, desempeño, pasajeros, metodología.

ABSTRACT

The public transport system in Lima aims to provide the transfer service to people who require it. However, the users of this service did not have a technological platform in which they could record the quality of the service received, the absence of this information generated uncertainty and mistrust in passengers when boarding a transport unit. The purpose of this thesis was to develop the prototype of a mobile application that contributes to the management of compliance and performance of public transport lines in Lima, through the recording of incidents that occurred during the provision of the service. For the structuring of this Thesis, the methodology of the Project Management Institute was used, preparing the project initiation documents, schedule of activities and the breakdown structure of the work. The development of the mobile application was carried out using the agile Scrum methodology (proper name), in order to obtain incremental advances in a short time. As a result, it was possible to develop a mobile application that recorded information on the incidents that occurred during the provision of the urban passenger transport service in the city of Lima, 91% of the users surveyed were able to satisfactorily record the incidents that occurred during the provision of the service, thus Even 94% were facilitated to read the recorded incidents, 77% found the information shown by the mobile application useful and for 70% the information could be an input for evaluating the service.

Key words: public transport, application, performance, passengers, methodology.

INTRODUCCIÓN

En Lima metropolitana operan diversas empresas que brindan el servicio de transporte de pasajeros, a las cuales la Municipalidad de la ciudad les asigna rutas de circulación con el fin de satisfacer la demanda de transporte de los usuarios en la totalidad del territorio Limeño. Los vehículos de transporte público obtienen su permiso de circulación luego de cumplir con los requisitos que la Municipalidad les exige para autorizar su funcionamiento. Luego de ello, las empresas inician sus operaciones de transporte público y entran a un proceso de monitoreo, seguimiento y control en el desempeño de sus actividades, la información de gestión y desempeño es obtenida mayormente a través de las multas o infracciones impuestas por incumplimiento de reglas o normativas de tránsito. Sin embargo, existen datos que no se registran de manera automatizada u oportuna, tales como cumplimiento de rutas y paraderos, cumplimiento de horario, servicio prestado por conductor o cobrador, estado del vehículo y calidad del servicio prestado, entre otros.

En este sentido, se vienen realizando diversos estudios orientados a conocer el estado del transporte público en Lima, con la finalidad de tener una línea base que permita implementar adecuadamente las reformas necesarias para resolver esta problemática. Existen también distintas instituciones públicas y privadas que analizan y emiten documentos técnicos sobre los problemas del transporte público.

Ante este panorama, surge el objetivo de la presente Tesis, el cual consiste en el desarrollo de un aplicativo móvil que permita registrar información por parte de los usuarios y en tiempo real de la calidad de prestación del servicio de transporte urbano.

El desarrollo de esta Tesis se desglosa en 5 capítulos estructurados según la metodología PMI. En el capítulo 1 se describe el problema a resolver, los objetivos y preguntas planteadas y la justificación e importancia. En el capítulo 2 se desarrollan el marco teórico de la solución propuesta, los conceptos y las bases teóricas sobre la cual se sustenta la propuesta de solución. En el capítulo 3 se presenta la metodología de investigación y la metodología a implementar en la solución planteada. En el capítulo 4 se exponen los requerimientos, el análisis y la arquitectura de la solución. En el capítulo 5 se detalla cómo se construyó y probó el aplicativo en la solución implementada.

CAPÍTULO I

Planteamiento Del Estudio

1.1 Planteamiento y formulación del problema

La ciudad de Lima alberga más de nueve millones de personas y las pérdidas económicas generadas por el excesivo tiempo que toma el trasladarse de un lugar a otro debido a la congestión vehicular, pueden ser muy significativas. Para el 2030, World Urbanization Prospects proyecta que Lima tendrá 12,2 millones de habitantes y ello ocasionaría mayor dificultad para movilizarse. (1)

El transporte urbano en la ciudad de Lima es una actividad que se desarrolla en condiciones muy riesgosas y precarias para sus ciudadanos y constituye una fuente significativa de contaminación ambiental. Esta situación pone en grave riesgo los derechos fundamentales de la vida, la integridad personal y la salud pública, además, la calidad del servicio de transporte de pasajeros es pésima. (2)

Para los ciudadanos que deben usar el transporte público en la ciudad de Lima el tránsito vehicular es un problema significativo (Tabla1). El incremento desmedido del parque automotor es uno de los principales motivos de la saturación de las vías de Lima, ello lo muestra la Asociación Automotriz del Perú – AAP quien indica que en el 2018 se importaron 141,733 vehículos livianos entre ellos autos y satino vagón, a ello se suma la importación de vehículos menores que en el 2018 asciende a 259,753 entre las cuales encontramos motos y trimotos, además de la importación de vehículos pesados que asciende a 17,594 entre ellos encontramos a buses para transporte público, camiones y tractocamiones; todos estos vehículos que en breve entrarán en circulación en Lima y provincias. (3)

Según el observatorio ciudadano Lima Cómo Vamos, indica que el 75.6% de limeños se moviliza en transporte público o colectivo (Metropolitano, Metro de Lima, Corredores Complementarios, Bus, Coasters, Combi y Colectivo informal), el 25% de la población usa más de 2 horas al día en trasladarse a su destino y el 49% considera al transporte público como el

principal problema de Lima. El problema del transporte en nuestra ciudad es complejo y tiene muchas aristas. (4)

Resulta importante resaltar que el 25 % de la población dedica hasta dos horas al día a trasladarse desde su hogar hasta su centro de trabajo o estudios, además del costo de traslado podría llegar alcanzar el 25% del sueldo mínimo mensual de una persona. (5)

En efecto, el tráfico genera pérdidas económicas para el ciudadano y para el país ya que reduce su nivel de productividad y competitividad, también afecta la salud, produce estrés y reduce los tiempos que una persona tiene para pasar con su familia. (5)

Por otro lado, según Diego Vargas, Secretario Técnico del Consejo Nacional de Seguridad Vial, “uno de los principales problemas de tránsito es el volumen del parque automotor del país y en especial la antigüedad de los vehículos”. Por otro lado, el Ing. Johan Moran Muñoz que es Subgerente de Transporte y Transito de la Municipalidad de la Molina realiza un estudio sobre la problemática del transporte y tránsito y de acuerdo con ello obtiene algunos resultados. (6)

Tabla 1

Percepción sobre el tránsito en Lima

| ¿CÓMO DESCRIBIRÍA EL TRÁNSITO EN LIMA? | |
|---|--------|
| Caótico | 65.5% |
| Desordenado y desorganizado | 15.5 % |
| Congestionado | 9.0% |
| Genera malestar físico a la población | 7.8% |
| Pérdida de tiempo | 4.5% |
| Otros | 8.5% |

Fuente: (6)

Tabla 2
Principales Problemas que Enfrenta el Usuario Por el Transporte Público.

| ¿CUÁL ES EL PRINCIPAL PROBLEMA DE TRÁNSITO QUE ENFRENTA LA CIUDAD? | |
|---|-------|
| Excesiva congestión vehicular | 48.3% |
| Falta de un sistema de transporte público adecuado | 22.0% |
| Siniestralidad (accidentes de tránsito) | 10.0% |
| Corrupción en la gestión del sector | 10.0% |
| Contaminación ambiental | 6.8% |
| Otros | 3.3% |

Fuente: (6)

Como se puede apreciar en la (Tabla 1) los ciudadanos piensan mayoritariamente que el tránsito en Lima es desordenado y desorganizado, además, ellos opinan que les genera pérdida de tiempo y dinero. Asimismo, han identificado que el principal problema que enfrenta la ciudad es la excesiva congestión vehicular y que carece de un sistema de transporte público adecuado. (6)

Respecto a las cifras de registro automotor para el servicio de transporte público, en Lima Metrópoli de Lima y Callao al 2018, se tienen inscritos 26,805 vehículos para transporte de pasajeros. En la capital se tienen 16,751 unidades en 372 rutas, y en la Municipalidad Provincial de Callao 10,054 vehículos en 144 rutas, siendo este último municipio el que; aun cuando solo representa el 5% del territorio y tan solo el 10% de población, ha autorizado el 37.5% del total de la flota circulante en Metrópoli. (7)

Asimismo, dentro de la jurisdicción de la Municipalidad de Lima Metropolitana, los denominados “Minibús o Coasters” (entre 17 hasta 33 asientos) son los mayoritarios con 7,211 unidades, seguidos de los ómnibus (con más de 33 asientos) con 4,827 unidades y las Combis o Microbuses (camionetas rurales entre 10 y 16 asientos) con 3,509 vehículos. Por su parte, en el Callao, las Combis siguen siendo las unidades de transporte urbano mayoritarias con 7,492 vehículos, seguidas muy de lejos por las Coasters con 1,025 unidades y los Buses con tan solo 1,537. (7)

A pesar de que existen líneas de transporte público de diverso tamaño y en diversas rutas, no se tiene información de tiempos de salida o llegada de los buses, además, se evidencia que los conductores de los buses hacen uso de paraderos no autorizados para el recojo de pasajeros, mientras que en otros casos los conductores realizan trayectos segmentados causando malestar a sus usuarios. Estos y otros incidentes no son detectados oportunamente por las municipalidades o los organismos encargados del control de transporte público.

La Ley N° 27181, Ley General de Transporte y Tránsito Terrestre , así como la Ley N° 27972, Ley Orgánica de Municipalidades establecen que las municipalidades tienen potestad para regular el transporte público dentro de su jurisdicción, así como el Decreto Supremo 003-2014-MTC, considerando que dicha regulación se ejerce a su vez a través de la ordenanza 020-2007-MPC en el caso de la Municipalidad Provincial del Callao y de la ordenanza N° 1613 y posteriores, Ordenanza 1876 y Ordenanza 1893, en el caso de la Municipalidad Metropolitana de Lima.

La ley 27181 modificado por la ley 28172 (8), la cual en su artículo 5 señala que el Estado tiene la responsabilidad de garantizar la vigencia de reglas claras y eficaces en la actividad de transporte. Por lo tanto, debe asegurar la existencia de una fiscalización eficiente y protectora en favor de los intereses de los usuarios.

En relación con la ley 27181 el artículo 15 señala que las autoridades competentes respecto al transporte y tránsito terrestre son los siguientes:

- El Ministerio de Transporte, Comunicaciones, Vivienda y Construcción (MTC).
- Las Municipalidades Provinciales.
- Las Municipalidades Distritales.
- La Policía Nacional del Perú
- El Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual – INDECOPI.

El Ministerio de Comunicaciones, Vivienda y Construcción (MTC) es el órgano rector a nivel nacional en materia de transporte y tránsito terrestre a nivel nacional.

De acuerdo con ley 271811 (8), en materia de transporte y tránsito terrestre el estado peruano orienta sus normativas a la satisfacción de las necesidades de los usuarios y al resguardo de las condiciones de seguridad y salud, así como a la protección del medio ambiente y a la comunidad. En el caso del servicio de transporte urbano es la Municipalidad de Lima la responsable de actuar para el cumplimiento de lo señalado.

La Sub Gerencia de Transporte Urbano de la Municipalidad de Lima es la responsable de Programar, dirigir, ejecutar, coordinar y controlar las actividades de registro y autorizaciones para el servicio de transporte y para el uso especial de las vías y de autorizar y proponer la regulación del servicio de transporte público en el ámbito de la provincia de Lima (9). Para la renovación de licencia de conductor de transporte de pasajeros se indican como requisitos los siguientes:

- Formato de solicitud
- Fotografía digital.
- Copia de licencia de conducir vigente.
- Carta de presentación de la persona jurídica autorizada y
- Pago de derecho a trámite.

La oficina de Gerencia de Transporte de la Municipalidad de Lima Metropolitana es la encargada de iniciar y conducir este proceso. (10)

La ley 271811 (8) en su artículo 17 “De las competencias de las Municipalidades Provinciales” es la de Supervisar, detectar infracciones e importar sanciones por incumplimiento de los dispositivos legales vinculados al transporte y el tránsito terrestre.

Así mismo, el artículo 24 “De la responsabilidad administrativa por las infracciones” señala lo siguiente: “El propietario del vehículo y, en su caso, el prestador del servicio de transporte es solidariamente responsables ante la autoridad administrativa de las infracciones vinculadas a las condiciones técnicas del vehículo, incluidas las infracciones a las normas relativas a la protección del ambiente y seguridad, según lo que establece esta Ley 271811 y los reglamentos nacionales.”

Para las inspecciones de cumplimiento de las normas, la Municipalidad realiza las acciones de fiscalización del servicio de transporte que consiste en la detección de infracciones, imposición

de sanciones y la ejecución de estas. La Municipalidad utiliza inspectores los cuales son acompañados con miembros de la Policía Nacional del Perú. En el caso que los inspectores detecten infracciones impondrán la multa correspondiente.

Las imposiciones de estas sanciones requieren necesariamente la presencia de un fiscalizador el cual realiza este proceso de manera manual. Además, estas acciones comúnmente se realizan en operativos que conllevan a desplegar personal fiscalizador en la vía pública.

Es evidente que las municipalidades tienen problemas de registro de información del desempeño y cumplimiento de servicio de buses de pasajeros que complican una adecuada gestión del sistema de transporte pública.

Debido a las dificultades que tiene la Municipalidad de Lima para el registro oportuno de las incidencias durante la prestación de servicio de transporte público es que se plantea el uso de un aplicativo móvil que permita obtener información relevante sobre la gestión de cumplimiento y desempeño del transporte público en Lima de tal manera que sirva de complemento al momento de verificar y analizar las solicitudes de renovación de licencias.

1.1.1 Problema general

¿De qué manera se puede obtener, almacenar y mostrar información del cumplimiento del servicio brindado por las empresas de transporte urbano de pasajeros en la ciudad de Lima?

1.1.2 Problemas específicos

¿De qué manera se puede registrar las incidencias ocurridas durante el servicio de transporte urbano de pasajeros en la ciudad de Lima?

¿De qué manera se puede almacenar información sobre las incidencias ocurridas durante la prestación del servicio de transporte público en la ciudad de Lima?

¿De qué manera se puede mostrar a los usuarios información de las incidencias ocurridas en los buses de servicio de transporte urbano de pasajeros en la ciudad de Lima?

¿De qué manera se puede mostrar información que podría ser insumo para la evaluación del desempeño del servicio de transporte urbano de pasajeros en la ciudad de Lima?

¿De qué manera se pueden desarrollar funcionalidades de utilidad en un aplicativo móvil para el registro y visualización de las incidencias ocurridas durante el servicio de transporte urbano en la ciudad de Lima?

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un aplicativo móvil que permita registrar, almacenar y mostrar información del cumplimiento y desempeño del servicio brindado por las empresas de transporte público de pasajeros en la ciudad de Lima.

1.2.2 Objetivos específicos

Desarrollar una aplicación móvil que registre las incidencias ocurridas durante el servicio de transporte urbano de pasajeros en la ciudad de Lima.

Desarrollar una base de datos que almacene información sobre las incidencias que se registren en el servicio de transporte urbano de pasajeros en la ciudad de Lima.

Desarrollar una aplicación móvil que muestre información de incidencias de los buses ocurridas en el servicio de transporte urbano de pasajeros en la ciudad de Lima.

Desarrollar una aplicación móvil que muestre información de incidencias de las empresas de buses ocurridas en el servicio de transporte urbano de pasajeros en la ciudad de Lima.

Desarrollar funcionalidades en una aplicación móvil que sea de utilidad a los usuarios para el registro y visualización de las incidencias ocurridas durante el servicio de transporte público en la ciudad de Lima.

1.3 Justificación e importancia

1.3.1 Justificación teórica

Los usuarios de transporte público de Lima no cuentan con un aplicativo que les permita visualizar información sobre la calidad del servicio brindado por las diferentes unidades de transporte. El desarrollo del aplicativo móvil objetivo de esta tesis es importante por ser una herramienta para la recolección y visualización de información referente a las incidencias ocurridas durante la prestación del servicio de transporte urbano, se convierte en una base referencial y documentada para las empresas fiscalizadoras que sustenta y facilita la toma de decisiones en pro de la mejora continua del servicio. La información recopilada durante el proceso de desarrollo de este aplicativo cimienta las bases teóricas para mejoras y aplicaciones complementarias futuras.

1.3.2 Justificación práctica

Lograr la satisfacción del cliente en términos de cumplir con un buen servicio debe ser la premisa de todos los transportistas públicos del país, al medir la frecuencia con la que ocurren los problemas sobre el mal trato del conductor o cobrador hacia el pasajero, incumplir con la ruta o paraderos establecidos, irrespetar las luces de semaforización o mal estado físico de la unidad de transporte, se obtiene un panorama de cómo se está llevando a cabo la tarea de transportar pasajeros por la ciudad de Lima y si los usuarios de transporte público están satisfechos con el servicio brindado, proporcionando a los organismos reguladores una herramienta de control y seguimiento y a los usuarios un canal de comunicación con dichos organismos para reportar las incidencias ocurridas; con las pruebas realizadas en este trabajo de investigación se podrá

determinar el impacto positivo que podría tener la implementación del aplicativo para mejorar el desempeño del servicio de transporte público.

1.3.3 Justificación social

Será importante contar con la información que registre la sociedad a través de los usuarios que usan el servicio de transporte de pasajeros, debido a que son necesarias sus observaciones y sus reclamos para mejorar, corregir o sancionar a los responsables de brindar este servicio público.

1.3.4 Importancia a nivel social

El problema del transporte público en Lima tiene muchos años sin ser resuelto y las distintas soluciones que las autoridades vienen implementando no dan resultados efectivos, la pérdida de tiempo todos los días genera mucho malestar, por otro lado, el desorden e informalidad es continuo y se acrecienta día a día.

Por otro lado, del total de viajes en Lima y Callao, casi el 25% de los mismos ocurre principalmente en dos horas, entre las 7:00 a.m. y las 9:00 a. m., y son los distritos de Lima Centro, en especial el Cercado de Lima, Miraflores y San Isidro, los que más viajes atraen. Estos viajes provienen en su mayoría de zonas periféricas como Comas, La Molina y Lurigancho. Esto demuestra la importancia de la planificación urbana y el modelo de ciudad compacta para disminuir los viajes largos que atraviesan la ciudad, generan pérdidas de horas/persona e impactan negativamente en la calidad de vida. (11)

Además, se debe tener en cuenta que entre las 7:00 – 9:00 de la mañana, es la hora de entrada, y entre las 17:00-19:00 horas, son los dos momentos que existe mayor demanda de viajes, por lo que una falta de una mayor dosificación de unidades de transporte en esas horas genera saturación excesiva en los puntos más concurridos de Lima, ocasionando alto tránsito de unidades de transporte y genera por consecuencia, mayores tiempos de desplazamiento de la oficina o universidad a casa. (12)

Esta investigación evidencia que la baja calidad en el servicio de transporte público que se brinda obedece sustancialmente a que las empresas de transporte no suelen controlar el desempeño de sus trabajadores principales, como son el conductor y el cobrador. Los incidentes atribuibles al servicio que ocurren durante el viaje de un pasajero usualmente no son registrados, por lo que finalmente, la municipalidad de Lima no tiene la información relevante y oportuna para tomar decisiones proactivas o analizarlas y decidir la renovación de las licencias del conductor o de la empresa de transporte.

Allí radica la importancia de contar con una herramienta que registre la calidad del servicio de transporte público de pasajeros, el cumplimiento de las rutas y horarios y las incidencias que ocurren durante la prestación del servicio.

De ahí que, la presente investigación propone desarrollar un aplicativo móvil que contribuya en registrar la información necesaria para la mejora de la calidad del transporte público y sea insumo en el proceso de cumplimiento y desempeño para toma de decisiones de los organismos encargados de renovación de licencias.

1.3.5 Importancia a nivel tecnológico

Esta investigación toma en cuenta el continuo avance tecnológico, sobre todo las herramientas de tecnología que permiten la construcción de herramientas o servicios que contribuyen a gestionar de manera óptima los procesos de negocio, que son parte fundamental de las organizaciones, estas interacciones con el cliente generan información que posteriormente puede traducirse en conocimiento, el cual permite determinar acciones de corrección o ajustes a los procesos.

En el escenario de la gestión del transporte urbano se utilizan las tecnologías innovadoras como GPS, comunicación móvil (por voz, texto y/o video), Cloud Computing, redes 4G y 5G, etc. Para el desarrollo de nuevos servicios se deberán considerar la utilización de todas o algunas de estas tecnologías, ello dependerá de los objetivos que se quiere alcanzar como servicio.

Por otro lado, a nivel mundial se tienen muchas experiencias sobre servicios que sirven para mejorar la movilidad del usuario del transporte urbano como, por ejemplo; TuRuta, Moovit, City

Mapper, Trafi, etc. (Tabla 3). También existen algunos servicios estatales que apoyan a la seguridad de las empresas de transporte público en carreteras como la app Comobity que fue diseñada por la Dirección General de Tráfico (DGT) de España, también existen otras experiencias como My Truck diseñada por la multinacional Volvo, ActiRuta que sirve para optimizar los recorridos y las rutas para la entrega de mercancías, OnTruck que promueve el transporte colaborativo ya que permite a las empresas de transporte recibir pedidos directamente a su móvil con la información de la carga, horarios de entrega, destino, precios, etc. (13)

Realizando un levantamiento de información en Google Play, medios de difusión digital peruanos sobre los servicios móviles que se brindan en el país sobre gestión del transporte público o servicios de transporte público, se encontró la siguiente información:

Tabla 3
Servicios Apps relacionados al Transporte Público en Perú.

| APP | TIPO DE SERVICIO | CANTIDAD |
|--------------|------------------------------|-----------------|
| Turuta | Transporte público | 1 |
| Trys Mi Bus | Transporte público | 1 |
| Moovit | Transporte público | 1 |
| BUS.ALTOKE | Transporte público | 1 |
| Blablacar | Asientos libres en vehículos | 1 |
| Efletex | Transporte de carga | 1 |
| Cabify | Servicio de Taxi | 1 |
| Uber | Servicio de Taxi | 1 |
| MyTaxi | Servicio de Taxi | 1 |
| Easy taxi | Servicio de Taxi | 1 |
| TOTAL | | 10 |

Fuente: Elaboración Propia

Cabe resaltar que todos ellos brindan servicios dirigidos al ciudadano o usuario del servicio público, por lo cual se abre la oportunidad de poder realizar o desarrollar un servicio dirigido a la gestión y control del servicio de transporte público para la Municipalidad de Lima Metropolitana. (14)

La importancia de este proyecto radica en el desarrollo de un servicio tecnológico que permitirá a los usuarios del servicio registrar información de las empresas de transporte, de sus conductores y cobradores, permitiendo posteriormente evaluar el desempeño y la calidad del servicio que brinda a sus usuarios. La información resultante de este servicio podrá ser el insumo básico para la renovación de las licencias a las empresas de transporte público.

Esta investigación propone el desarrollo de un aplicativo móvil usando un modelo tecnológico, sistematizado y automatizado de registro de información que los usuarios del servicio de transporte público de pasajeros utilicen y que posteriormente pueda ser de utilidad para las empresas y autoridades involucradas en los procesos de renovación de licencia de transporte de pasajeros.

1.4 Alcances y Limitaciones

1.4.1 Alcances

El presente trabajo de investigación contempla el prototipo y el desarrollo de un aplicativo móvil que recogerá información sobre la calidad del servicio, incidencias cumplimiento y desempeño del servicio de transporte de pasajeros en la ciudad de Lima. Las pruebas se realizarán funcionalmente a nivel tecnológico garantizando el almacenamiento de la información y permitiendo la visualización de estos a través de una plataforma móvil.

1.4.2 Limitaciones

Debido al difícil acceso a las instituciones encargadas de las gestiones de transporte público, el trabajo de investigación no contempla la implementación y difusión del aplicativo propuesto en la municipalidad de Lima Metropolitana.

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedentes del problema

2.1.1 Antecedentes Internacionales

La tesis “Calidad del Servicio de Transporte Urbano en la Ciudad de Cuenca” (15) realizada en la ciudad de Cuenca – Ecuador, buscó como objetivo encontrar los aspectos fundamentales de la calidad del transporte urbano de pasajeros en Cuenca utilizando la metodología de seis pasos para obtener una medida de disgusto de los usuarios en un sistema de transporte en particular. Para ello, se realizó entrevistas a las personas que usan el transporte público de la ciudad de Cuenca y el 41% indicó que el servicio del transporte es bueno y el restante 59% de usuarios indicaron que es malo debido a que sienten inconformes. La investigación realizada determinó que una serie de mejoras permitirán brindar una excelente calidad del servicio de transporte urbano de pasajeros que incluyen por ejemplo capacitaciones al conductor sobre el trato al usuario mejora de los paraderos con una ubicación estratégica, otorgar información acerca de las rutas y tiempos, implementación de bitácoras para una higiene excelente, al igual que exigir el mantenimiento de los autobuses para evitar la contaminación. Esta investigación permitió incorporar a la presente Tesis nuevos factores para evaluar la calidad del servicio en el transporte urbanos en Lima e incorporarlo como criterio al momento de revalidar la licencia de las empresas de transporte de pasajeros.

Por otro lado, en la tesis “Modelo de Satisfacción de los Usuarios de Transporte Público Tipo Bus Integrando Variables Latentes” (16) se planteó como objetivo desarrollar un modelo que represente la satisfacción de los usuarios de transporte público tipo bus en la ciudad de Medellín, incluyendo variables latentes de percepción. Para esta investigación se siguió la metodología de modelación a partir de ecuaciones estructurales utilizando el software AMOS. Se obtuvo como resultado que la satisfacción de los usuarios no sólo depende de lo que el modo pueda ofrecer (atributos), sino también de variables propias del individuo: imagen, norma social, valor percibido; y de variables externas a ambos como el ambiente, es decir, el contexto urbano en el que se desarrolla este modo. Se concluye que los viajes de los pasajeros deben entenderse no sólo

como llevarlos de un lugar a otro, sino como una experiencia que empieza desde que el usuario pisa el primer escalón para ingresar al bus teniendo en mente esto ya cambia la visión de cumplir la necesidad de transporte de las personas por una que comprende la importancia del trabajo, no se ofrece la satisfacción de una obligación sino un adecuado servicio.

Dicha investigación aportó a la presente Tesis el factor de perceptibilidad de los usuarios de transporte de pasajeros como clientes y no como usuarios de este servicio.

2.2.2 Antecedentes Nacionales

En la tesis “Simulación del Sistema de Transporte Urbano de la Ciudad de Tarapoto para Planificar Escenarios Futuros” (17), en la Universidad Nacional de San Martín - Facultad de Ingeniería de sistemas e informática tuvo como objetivo buscar predecir escenarios en función a valores cuantitativos de la aglomeración urbana de Tarapoto. Para ello, usa indicadores como la cantidad de unidades móviles motorizadas, cantidad de conductores, precio, población, precio de la gasolina, congestión de tráfico e incidencias de accidentes. La investigación indicó que sí se presenta mayor malestar social a causa del inadecuado desarrollo del sistema de transporte urbano y mayor inseguridad causada por los accidentes de tránsito. Se logró modelar que el crecimiento poblacional y el crecimiento automotriz influyen directamente en los accidentes de tránsito en este medio de transporte. Además, se determinó que el factor socioeconómico es el más sensible y afecta directamente al servicio de transporte de pasajeros.

Esta investigación contribuyó a la presente Tesis para incorporar las incidencias y accidentes como reporte e insumo de verificación al momento de evaluar la renovación de licencia de conducir del conductor y de la empresa de transporte.

Adicionalmente, en el informe de investigación denominado “Transporte Público de Pasajeros en Lima” (18) se indicó que desde hace varios años se vienen realizando diversos estudios orientados a conocer como estaba el servicio del transporte público en la ciudad de Lima, con el objetivo saber cómo se encontraban y con ello permitir realizar adecuadamente los cambios que contribuyan en solucionar los problemas y uno de los primeros documentos de gestión que trató de manera integral la situación del transporte en la ciudad capital es el denominado «Plan Maestro de Transporte Urbano de Lima y Callao (2005)». Según la empresa Protransporte, antes

de empezar el proceso de cambio, el servicio de transporte en la ciudad de Lima tenía como atributo ser deficiente, con servicio pausado, demasiado inseguro y desaseado. Además, el transporte tenía demasiada oferta y esto genera tráfico, lentitud en el recorrido y desorden entre los mismos buses. Debido a esta problemática era necesario realizar mayores estudios orientados a conocer aspectos importantes para enfrentar el caos vehicular, como son: Población y demanda, Transporte público – rutas y flota, Impacto del transporte sobre la población.

Dicha investigación concluye que el sistema de transporte público en Lima Metropolitana está afectado por una serie de factores que repercuten en su calidad y eficiencia, los aspectos que ocasionan ello son la alta población en la ciudad, el crecimiento permanente de venta de otros vehículos motorizados, alta incidencia de accidentes de tránsito y la contaminación ambiental. Actualmente se tienen bastante cantidad de vehículos de transporte público que no tienen las mínimas características necesarias para hacer más eficiente el traslado de pasajeros (antigüedad, tamaño, parámetros técnicos), lo cual genera una mayor congestión vehicular en las principales vías de la capital. Este informe nos permitió incorporar datos adicionales como atributos del bus y del servicio al momento de recopilar la información del servicio de transporte de pasajeros.

2.3 Bases teóricas

En la ciudad de Lima la autoridad competente para regular y reglamentar este servicio es la Municipalidad Metropolitana de Lima y lo hace mediante los siguientes entes:

- Concejo de la Municipalidad.
- La Municipalidad de Lima.
- La GTU (Gerencia de Transporte Urbano) de Lima.
- Instituto PROTRANSPORTE de Lima.

Actualmente en Lima no se difunde información relevante sobre las rutas y paraderos de las empresas de transporte de pasajeros lo que con lleva a que el ciudadano se entere de esta información a través de consultar a otro ciudadano, al cobrador o al chofer del bus.

Según un estudio de la empresa CPI y el INEI en el 2019 se estima una población de 11'591,400 aproximadamente en Lima (19). Además, se estima que en Lima se concentra el 65,6% del parque automotor nacional, lo que vendría representado aproximadamente 2'423,696 vehículos, según estudio del INEI en el 2014.

La Municipalidad de Lima realiza esfuerzos dirigidos a la reforma del transporte público y su éxito o fracaso gira en torno a la adecuación y toma de decisiones con respecto a los puntos enumerados. No obstante, es importante mencionar que existen otros factores exógenos que podrían perjudicar el desarrollo de la reforma del sistema de transporte público en Lima Metropolitana. (20)

El primer factor está vinculado al crecimiento explosivo que viene teniendo el sector informal de transporte público. Se debe implementar algunas medidas, regulaciones y prohibiciones en paralelo a la reforma del sistema de transporte público para desalentar el crecimiento explosivo que viene teniendo la flota de taxis informales. (20)

El segundo factor será la capacidad que tenga la demanda laboral del sector formal de la ciudad de absorber la mano de obra no calificada que actualmente está relacionada al sistema de transporte público (choferes, cobradores, "dateros", entre otros) y a los taxis formales que dejarán de circular por la sustitución de esta oferta de transporte de pasajeros por el nuevo sistema de transporte público articulado de calidad. En la medida que la demanda laboral del sector formal no sea capaz de absorber de manera eficiente a esta mano de obra, las presiones políticas y sociales que generará la reforma serán difíciles de sostener para las autoridades del gobierno. Esto muestra la necesidad de realizar una transición o migración lenta del sistema de transporte público actual al sistema de transporte público reformado. (20)

El tercer factor está vinculado a las mayores facilidades que tienen los usuarios en la actualidad de adquirir créditos vehiculares y vehículos privados propios. Se debe encontrar formas de desalentar la adquisición de vehículos privados para desacelerar el crecimiento de la motorización en Lima Metropolitana en el corto, mediano y largo plazo. Las soluciones en esta línea son diversas, pero claramente es necesario tomar decisiones en este sentido para evitar que el número de vehículos motorizados privados per cápita siga incrementándose. (20)

El factor número cuatro se relaciona con el aumento de inseguridad que se manifiesta en las ciudades. El gobierno central debe incorporar leyes para mitigar esta percepción y, además, incorporar los controles pertinentes para realizar el seguimiento y monitoreo de la seguridad de las personas durante el uso del servicio de transporte público. De no hacerlo se desalentará el uso del sistema de transporte público de calidad implementado en Lima Metropolitana. (20)

El quinto factor es aumento poblacional urbano que se viene dando en forma desordenada y que la Municipalidad de Lima enfrenta. La ciudad debe crecer de manera planificada a través de una institución independiente del gobierno central o municipal de tal manera que se realice de manera sostenible acorde a un sistema de transporte público. Por ello, es necesario contemplar variables como como la densidad poblacional mínima que permitan justificar la generación o ampliación de rutas de transporte público que se forman o se formarán en los años venideros. (20)

Considerando todas las medidas que se aplicarán en los años venideros es evidente que las autoridades deben contemplar procedimientos que retrasarían o ayudarían en mejorar la gestión del transporte urbano de pasajeros. Como parte de esas medidas las autoridades deben tomar en cuenta lo siguiente:

La falta de gestión que las instituciones gubernamentales tienen para llevar a cabo los cambios integrales del servicio de transporte urbano de pasajeros. Es notorio la baja capacidad de las autoridades locales y de gobierno central para la aplicar leyes importantes sobre infraestructura, educación, salud entre otros sectores. Además, no se llega a una adecuada coordinación entre entidades de gobierno en diferentes niveles en particular sobre las responsabilidad y alcances que tienen cada uno todo ello impacta negativamente en tratar de implementar mejoras al servicio de transporte público. (20)

El aumento sin planificación de la población urbana en la periferia crea serios problemas alrededor de las principales ciudades como se muestra en (Figura1). Las diferentes alternativas de solución para el servicio de transporte urbano de pasajeros en la ciudad de Lima deben ser planteamos como solución al problema actual, pero pensando también en una planificación futura urbana. El futuro aumento de la población debe ser ordenado de tal forma que pueda organizar un servicio masivo adecuado, ordenado e integrado. (20)

Figura 1
Estadística Población Urbana vs Rural



FUENTE: I.N.E.I. Estimaciones y proyecciones de población en base al Censo 2017
ELABORACION: DEPARTAMENTO DE ESTADISTICA - C.PI.

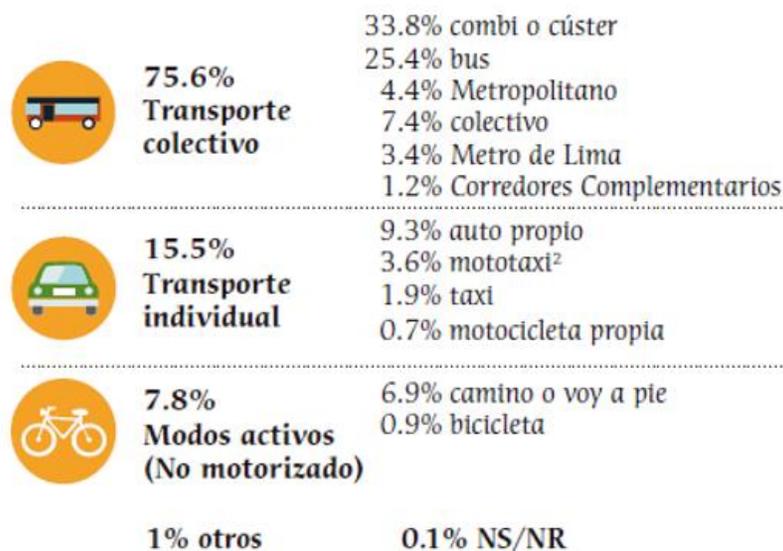
Fuente: (19)

El sistema de transporte metropolitano de Lima cuenta con una serie de inconvenientes que hacen más complicado el alcance de sus objetivos:

- Pocos vehículos de gran capacidad en el parque de transporte público.
- Falta de visión empresarial por parte de las empresas de transporte.
- Falta de visión innovadora para reducir el uso del transporte público en Lima. (21)

En Lima existe un observatorio denominado “Lima Como Vamos” el cual indica que el 75.6% de los limeños usa el transporte público, colectivo o cualquier medio complementario a estos para desplazarse de este grupo, el 25% viaja en buses y el 34% en combis. Por otro lado, solo un 15.5% lo hace en transporte individual: 1.9% viaja en taxi, 9.3% en auto propio. Además, un 7.8% usa transporte no motorizado y dentro de este grupo, el 0.9% lo hace en bicicleta y 6.9% va a pie. (22) (Figura 2)

Figura 2
Medios de transporte público en Lima



Fuente: Lima Cómo Vamos (22) Base: Total de entrevistados que estudia y/o trabaja fuera de casa

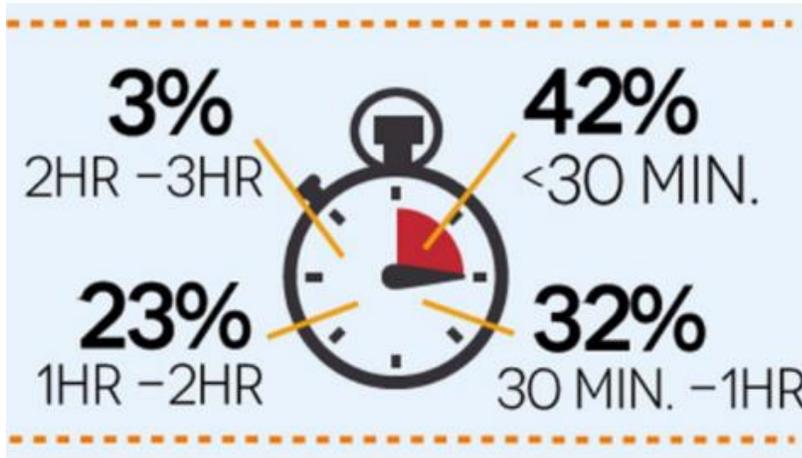
Según el tiempo y destino de viaje, en comparación con 2014, el 51.8% de limeños manifestó que su trayecto demora el mismo tiempo que en 2015. Un 37.6% señaló que le toma más tiempo, y solo el 9.8% redujo su tiempo de viaje. Además, al consultar sobre el tiempo que demoran desde que salen de su casa hasta llegar a su centro de trabajo o estudios, el 25% de limeños señaló que dedica más de dos horas al día en movilizarse. (22)

Los distritos a dónde más se trasladan los limeños a trabajar y/o estudiar es el Cercado de Lima (14.5%), seguido por San Juan de Lurigancho (8.1%) y Santiago de Surco (6.4%). San Isidro y Miraflores, distritos que alojan un importante número de oficinas y centro de estudios, son el destino del 4.7% y 5.9% de personas, respectivamente.

La necesidad del pasajero para llegar a su destino (Figura 3) es desconocida por la empresa de transporte y por quienes operan estas unidades, tampoco no hay ninguna obligación de estas empresas de transporte en cumplir con tiempos de recorrido. Por ejemplo, una persona que sube a su bus en un punto A, el tiempo que le tomara para llegar a un punto B es variable, estará dependiendo de la hora en la que se requiere el servicio, del día si es laborable o no, si existe alguna restricción por obras realizadas por la municipalidad, etc. Todo ello es visto como serios problemas en el servicio de transporte y por lo cual existen diversos motivos como no tener un

adecuado manejo en el aumento de los buses y vehículos además no se cuenta con leyes o procedimientos que permitan regular el crecimiento de vehículos de transporte informal y a esto se añade la poca educación o conocimiento respecto de los actores como pasajero y conductores respecto al reglamento de tránsito (21).

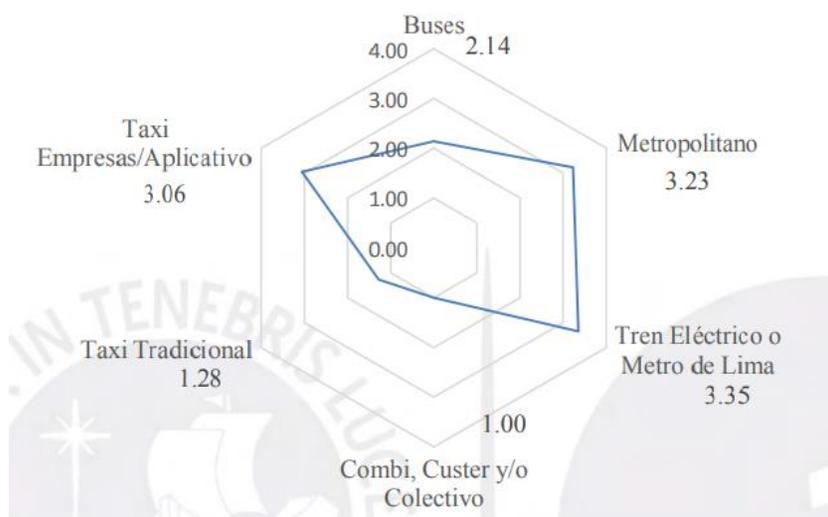
Figura 3
Tiempos estimados por el Ciudadano en Llegar a Destino



Fuente: (4)

Los buses no pueden satisfacer la demanda del servicio de transporte esta deficiencia es cubierta por otros medios de transporte como las combis, Coasters, y colectivos los cuales son mayormente informales, ofrecen un servicio inseguro, nada confortable, contaminando el medio ambiente y que a menudo ocasionan accidentes de tránsito y generan el desorden del sistema de transporte. (2). El siguiente gráfico (Figura 4) muestra las preferencias de los ciudadanos:

Figura 4
Estadística sobre las Preferencias de los Ciudadanos



Fuente: (2)

En el trabajo (21) señala que las reparaciones de las pistas y vías por donde circulan los vehículos son necesarias debido a que con ello se busca beneficiar a la ciudadanía. Pero, las actividades de remodelación incrementarán el tiempo de viaje a los usuarios por lo que las instituciones a cargo de estas tareas deberían comunicar con la debida anticipación del tiempo extra que tardarán los usuarios de esa manera los usuarios frecuentes podrán planificar rutas alternas o buscar la forma de completar su viaje diario.

Mantener información unificada será de gran utilidad debido a que permitirá direccionar las decisiones considerando lo que los involucrados en el servicio requieren realmente. De lo contrario estamos ayudando a generar malestar en todas las personas, usuarios directos o no del servicio, personas que simplemente transitan o vehículos particulares que solamente circulan por la misma vía que los buses todos ellos estarán afectados por la falta de toma de decisiones adecuadas y oportunas. (21)

- Administración de las salidas de los autobuses.
- Administración de los buses registrados por ruta, y organizar su partida.
- Administración de los horarios para cada ruta. Incluye horarios de paso de cada medio de transporte en cada paradero, desde la partida hasta la llegada.
- Control del tráfico.

- Control del estado de los tramos en la ruta. (21)

El servicio de transporte urbano tiene problemas comunes a todas las personas que hacen uso de este servicio debido a ello la solución que se desarrolla para mitigar y registrar estos inconvenientes es una propuesta a través de un prototipo y desarrollo de un aplicativo que brinde información sobre el desempeño de las líneas de transporte público, del cumplimiento de requisitos, comportamiento del personal registrado por la empresa de transporte que son insumos básicos para la determinación de la renovación de licencias en Lima Metropolitana. De ello se desprende que esta propuesta contribuye con obtener información relevante de los mismos usuarios sobre el servicio brindado por los transportistas.

2.1.1 Otorgamiento de concesiones

Según la ordenanza municipal 1599 (23) la Municipalidad Metropolitana de Lima otorga concesiones para la prestación del servicio de transporte público regular de personas de conformidad con lo establecido en la presente Ordenanza. Las reglas de los procesos de licitación describen los criterios para acceder y poder mantenerse en la ruta licitada y finalmente en los documentos de concesión, y, lo que las demás ordenanzas indiquen de manera explícita. El responsable de conducir este proceso es el alcalde y emite un informe en reunión de consejo metropolitano.

El permiso para operar la ruta se formaliza a través de la firma del contrato de concesión la misma que suscribirán el alcalde de Lima y los gerentes o representantes de la empresa que gana el contrato. Es posible registrar este contrato en registros públicos si la empresa de transporte lo solicite y siempre que asuma los gastos de este trámite.

2.1.2 Proceso de licitación de ruta

Según Borjas (21) el concurso para asignar rutas de circulación a los buses contribuye directamente a que las empresas de transporte de pasajeros puedan presentarse y obtener una de las rutas con el fin de poder atender la demanda del servicio por un período de tiempo determinado. Todo ello se lleva a cabo con presencia de personal notarial que garantizan la transparencia. Para iniciar este proceso se siguen los siguientes pasos:

Establecer una nueva ruta. Se consigue una vez que se logre realizar el estudio específico, especializado y coordinado entre los órganos de gerencia de transporte. Se toma en cuenta:

Aumentar todo lo que sea posible el acceso al servicio donde haya demanda con mejoras en la atención y el servicio.

Realizar cambios al servicio actual buscando incrementar la calidad, esto se logra teniendo claridad en la satisfacción de la demanda y usando los recursos viales existentes que ofrece el Área Metropolitana de Lima.

Identificar las variantes de recorridos tales como vías rápidas, intersecciones, rutas de inicio y de final todo ello en función a la demanda requerida y la capacidad de satisfacerla.

La necesidad de fomentar la estrecha comunicación y que involucra los buses y los paraderos de tal forma que se garantice al pasajero un servicio idóneo.

Atender todas las coberturas que existen identificando las particularidades de cada área y tipo de servicio que requiere mejorar.

Dimensionar una malla de rutas alimentadoras que incrementen la accesibilidad de los usuarios a los corredores masivos y troncales.

Proponer vías alternas para cada ruta en los que los pasajeros puedan elegir entre rutas directas y rápidas o realizando cambio de bus entre rutas diferentes.

Rentabilizar las posibles vías de uso de buses analizando las variables de coste para operarlas teniendo como fin satisfacer la demanda de los usuarios.

Identificar las capacidades de los buses de transporte con el estado de las vías logrando crear servicio acorde a la ruta y al pasajero.

Racionalizar los costos operativos de los buses alienado a la ruta asignada y a los requerimientos de recorrido de los pasajeros.

2.2.2 Proceso de autorización de servicio

Es el proceso para la obtención de la autorización de servicio y tiene como objetivo permitir a las empresas de transportes obtener la autorización para brindar el servicio de transporte público en la ciudad de Lima, estos deben cumplir con los requisitos y las obligaciones a las que conlleva tener la autorización (9). Los pasos para completar este proceso son los siguientes:

Las empresas que soliciten la autorización deberán enviar lo siguiente:

- Hoja resumen de la Empresa.
- Copia simple del estatuto social y de la partida registral de la empresa donde se establezca como principal actividad de la sociedad la de prestación del servicio de transporte.
- Declaraciones Juradas suscritas por los representantes de la Empresa.
- Declaración Jurada de contar con el patrimonio mínimo mayor o igual a 100 Unidades Impositivas Tributarias en rutas urbanas y 50 en rutas periféricas.
- Declaración jurada de contar como mínimo con un área de administración y operaciones.
- Relación de conductores y de cobradores (registrados en la GTU).
- Relación de vehículos (deben de cumplir con los requerimientos técnicos establecidos en la Ordenanza N° 1599).
- Copias simples de las Tarjetas de Identificación vehicular.
- Copias simples de los certificados SOAT o CAT, de las pólizas de seguro y de los certificados de inspección técnica complementaria.
- Declaración Jurada de no haber recibido sanción firme de cancelación o inhabilitación respecto del servicio que se solicita.
- Presentar estudio técnico para la asignación de la ruta (impreso y digital) sobre los planes funcionales y de operación.
- Declaración jurada de contar con una caja centralizadora para la administración.

La Municipalidad de Lima a través del área encargada de recibir la documentación durante un plazo de treinta días revisa toda la documentación. Si existen documentos observados son

devueltos a la empresa para que los subsane en cinco días luego de ese plazo de no haber subsanado se procede a emitir un documento declarando improcedente. En caso todos los documentos estén conforme a lo solicitados y a las normas establecidas se continuará con el proceso.

El área de Subgerencia de Regulación del Transporte envía a la Subgerencia de Estudios de Tránsito y Transporte. La Subgerencia de Estudios de Tránsito y Transporte realiza informe y lo remite a la Subgerencia de Regulación del Transporte. La Subgerencia de Regulación del Transporte revisa informe. La Subgerencia de Regulación del Transporte solicita a la empresa de Transporte subsanar las deficiencias presentadas en el expediente. Esta vez solo se tiene 10 días calendarios.

Si todos los documentos presentados son correctos se comunica a la empresa que se le entrega la concesión de la ruta para servicio de transporte. En quince días como máximo durante dos días la empresa anunciara la ruta asignada en el diario "El Peruano". En caso no se cumpla con este procedimiento se anulará la concesión.

Durante el período de publicación es posible que se presente objeciones por lo que el área encargada de la Municipalidad de Lima resolverá estas observaciones en coordinación con la empresa de transporte en un plazo de treinta días. De no lograrse la resolución se anulará la concesión.

En caso todos los puntos se cumplieron satisfactoriamente, la Subgerencia de Regulación del Transporte emite la resolución de la autorización en un plazo no mayor a 15 días. La autorización de servicio es intransferible e indivisible.

Luego de avisar la resolución la GTU evidencia los resultados en la página web institucional durante treinta días.

Pasado los treinta días de comunicar el resultado y de no haber objeciones se inicia el servicio a cargo de la empresa ganadora.

2.2.3 Plazo de vigencia

Es el período de tiempo por el cual una determinada empresa tiene derecho a transitar por una ruta asignada en cumplimiento del contrato de concesión otorgado por la Municipalidad de Lima.

2.2.4 Renovación de autorización

La GTU (Gerencia de Transporte Urbano) de manera previa al proceso de renovación de autorización, deberá declarar apta a la ruta en la que se presta el servicio que solicita ser renovado, de acuerdo con los estudios técnicos que desarrolle, con lo establecido en el Plan Regulador y el Sistema de Rutas.

Para seguir usando la ruta concesionada de prestación del servicio en la vía la empresa declarada como apta deberá presentar su solicitud de renovación de Autorización de Servicio dentro de los sesenta (60) días anteriores al vencimiento de esta y acreditar el cumplimiento de los requisitos establecidos en el artículo 14º de la presente Ordenanza. En ningún caso, la autorización cuyo plazo haya vencido genera un derecho adquirido para la empresa solicitante de la renovación.

La resolución que otorga la renovación de la Autorización de Servicio entrará en vigor desde el día siguiente de vencido el plazo de la autorización anterior.

No procederá la renovación, si se ha aplicado a la persona jurídica la sanción de cancelación o inhabilitación definitiva respecto de dicha ruta o servicio, según el caso.

2.2.5 Condiciones de prestación del servicio de transporte público

Según la ordenanza (23) las empresas autorizadas deben cumplir con las siguientes condiciones del servicio:

Del vehículo

Mantener las características técnicas generales y específicas de los vehículos, así como las demás condiciones que le permitieron acceder a la autorización para la prestación del servicio de transporte.

Realizar y garantizar el mantenimiento y limpieza adecuada para operación de los vehículos que comprenden su flota habilitada.

Comunicar a la GTU, en un plazo no mayor de cinco (5) días, la transferencia de los vehículos que integran su flota o cualquier variación que se haya producido en la información presentada para el otorgamiento de la habilitación.

Disponer que los vehículos habilitados porten elementos de emergencia, entendiéndose por tales:

- Extintores de fuego, en óptimo funcionamiento. El número de extintores y la clase de estos se regulan por lo previsto en la NTP 833.032, la que asume carácter obligatorio en el servicio de transporte.
- Conos o triángulos de seguridad,
- Botiquín para brindar primeros auxilios.

Atendiendo a las características de regularidad, continuidad y generalidad del servicio de transporte regular de pasajeros, las empresas autorizadas deberán contratar el servicio de grúas de remolque para la atención de sus vehículos durante la prestación del servicio.

Verificar antes de prestar el servicio que todos los neumáticos del vehículo habilitado cumplen con lo dispuesto por el RNV.

Verificar que el vehículo cuente con las láminas retro reflectivas y demás disposiciones relacionadas al tránsito, de acuerdo con la normatividad vigente.

Verificar que el limitador de velocidad y el dispositivo registrador, o el que lo sustituya, se encuentren en perfecto estado de funcionamiento.

Garantizar que el vehículo cuenta con cinturones de seguridad en el asiento del conductor y en los que se encuentren en la primera fila del vehículo.

Verificar que el dispositivo instalado en el salón del vehículo refleja la velocidad que marca el velocímetro.

De los conductores

Contar con el número suficiente de conductores y cobradores para prestar el servicio en los términos en que éste se encuentre autorizado, considerando lo establecido en la ficha técnica.

Prestar el servicio con conductores y cobradores que hayan aprobado la capacitación sobre trato a los usuarios y otras conductas propias de sus funciones.

Cumplir con inscribir a los conductores y cobradores ante la GTU y verificar que se encuentren habilitados, antes que éstos presten el servicio de transporte.

Garantizar que los conductores que resulten seleccionados aleatoriamente por la GTU cumplan con realizar un examen médico de comprobación de aptitud psicofísica.

Verificar que, antes de iniciar la conducción y durante la prestación del servicio, los conductores porten su licencia de conducir vigente, de acuerdo con la clase y categoría requerida por las características del vehículo y el servicio de transporte regular de personas; no sobrepasen el límite de edad máximo, de conformidad con lo establecido en el RNAT; y, no hayan ingerido alcohol o sustancias que produzcan alteración de los sentidos o del sistema nervioso. En este último caso, de evidenciarse algún síntoma que haga presumir esta situación, la empresa autorizada no deberá autorizar la conducción si previamente no se realizan los exámenes de descarte que resulten necesarios.

Verificar que el conductor que haya participado en un accidente de tránsito con consecuencias de muerte o lesiones personales apruebe un nuevo examen psicosomático. Los resultados del examen deberán presentarse ante la GTU dos (2) días antes que el conductor vuelva a prestar

el servicio de transporte, dentro de un plazo máximo de quince (15) días calendario de ocurrido el accidente.

Verificar que el conductor no realice jornadas de conducción efectiva continua de más de cinco (5) horas. El tiempo de descanso entre cada jornada de conducción deberá ser no menor de una (1) hora. En caso de que la jornada de conducción efectiva sea menor de dos (2) horas, el tiempo de descanso será no menor de treinta (30) minutos. La duración acumulada de jornadas de conducción no deberá exceder de diez (10) horas en un período de veinticuatro (24) horas.

Verificar que los conductores cuenten con información sobre las obligaciones que deben ser cumplidas durante la prestación del servicio, sobre las sustancias que no pueden ser consumidas por generar alteraciones de la conciencia, somnolencia u otro efecto que afecte la conducción.

2.2.6 Prestación del servicio de transporte público

Durante la prestación del servicio las empresas autorizadas deberán cumplir con las siguientes condiciones operacionales:

Colocar en el interior del vehículo, en lugar visible para el usuario, la información del código e itinerario de la ruta y paraderos autorizados.

Las tarifas del servicio que presta deberán ser colocadas en la parte lateral derecha externa e interna del vehículo.

Colocar el o los teléfonos de la empresa autorizada y los que señale la GTU, en la parte interna y externa del vehículo, para atender quejas, reclamos o denuncias de los usuarios. Colocar en la parte frontal externa e interna del extremo superior del vehículo un letrero visible de día y de noche con la indicación del origen, destino y código de la ruta.

Entregar comprobante de pago o boleto a los usuarios por la prestación del servicio. Reservar y señalizar como mínimo los dos (2) asientos más cercanos a la puerta de acceso delantera del vehículo para uso preferente de las personas con discapacidad, adultos mayores, mujeres gestantes y mujeres con bebés en brazos.

Identificar y señalar las salidas de emergencia.

2.2.7 Proceso de suspensión a la autorización de servicio

La aplicación de la ordenanza 1599 a una empresa de transporte señala los lineamientos y criterios de suspensión de la concesión de la ruta y permite la aplicación de acciones de prevención o de corrección otorgando plazos y mecanismos que regulan este proceso.

Por otro lado, Para Bonifaz (20) durante los últimos años, se han implementado determinados desarrollos tecnológicos que han permitido simplificar las transacciones que deben realizar los ciudadanos para adquirir y utilizar los pasajes de los distintos tipos de transporte público. La reciente implementación del dinero electrónico a los sistemas de transporte ha permitido que los usuarios ya no requieran del uso de dinero físico para poderse manejar a lo largo de los medios de transporte público. Asimismo, las principales ciudades a nivel mundial manejan un único medio de pago (a través del uso de una tarjeta única, al estilo de una tarjeta de débito) para los distintos tipos de transporte público (metro, autobuses, etc.) y un sistema tarifario integrado, de tal manera que se faciliten las transacciones para los usuarios y se le acerque al sistema de transporte público de calidad propuesto por cada ciudad.

Por otro lado, según la norma UNE -EN13816 de la Asociación Española para la Calidad (AEC); la cual establece unos criterios de evaluación a partir de los atributos del modo agrupados en ocho categorías: servicio ofertado, accesibilidad, información, tiempo, atención al cliente, confort, seguridad e impacto ambiental; esto con el fin de aportar un esquema a aquellas entidades prestadoras del servicio de transporte público para que se comprometan en un servicio de calidad en función de la satisfacción de las expectativas de los usuarios.

Según la Fundación Europea para el Manejo de la Calidad – EFQM (2006), la calidad se define como “todas las formas a través de las cuales la organización satisface las necesidades y expectativas de los clientes, su personal, las entidades implicadas financieramente y toda la sociedad en general”.

2.4 Definición de términos básicos

a) El servicio de transporte público urbano de personas

Vásquez (24) señala que es vital considerar el transporte de pasajeros como servicio básico y de acceso a todas las personas ya que es un servicio necesario su uso esta generalizado y que contribuye a satisfacer la necesidad básica de todos los usuarios y al mismo tiempo es una actividad que genera trabajo y ganancia económicas por lo que su operatividad se debe garantizar siempre.

b) Dirección general de circulación terrestre

En el Perú el Ministerio a cargo de la actividad de transporte cuenta en su organigrama un área dedicada a la gestión general de circulación terrestre y depende directamente del Viceministerio de Transportes. Tiene como fin normar, autorizar, supervisar, fiscalizar y regular el transporte y tránsito terrestre de personas y carga, actividad que realiza de acuerdo con lo dispuesto por el ordenamiento legal vigente.

c) Transporte público terrestre

Esta actividad está gestionada por el Ministerio de Transporte y comunicaciones y ejecutada a través de su Viceministerio la Dirección General de Circulación Terrestre (25). Entre sus responsabilidades están la de supervisar, normar, fiscalizar autorizar, y regular el transporte y tránsito terrestre de personas y carga. Además, las actuales leyes, como la 271811, le brindan también facultades a los Gobiernos Regionales y a las Municipalidades provinciales en el ámbito de su localidad.

d) Servicio de transporte de pasajeros terrestres en Lima Metropolitana

Servicio incorporado por el reglamento que regula el servicio de transporte de pasajeros a nivel local y nacional para Ómnibus y otras formas para Lima (RSTUPTU), aprobado mediante registro municipal No.104. De esa manera esta autoridad, que en este caso es la GTU otorga a

los particulares la facultad para prestar el “servicio público” de traslado local o inter- distrital de personas en una determinada ruta dentro del ámbito Urbano e Interurbano en una provincia. El título habilitante en este caso se denomina concesión, pero no necesariamente tiene la calidad discrecional que las caracteriza, no obstante, son otorgadas vía licitación pública. (25)

e) GTU-Gerencia de transporte urbano en Lima Metropolitana

En la Municipalidad de Lima se tiene un órgano especializado y dedicado a gestionar el transporte urbano entre sus responsabilidades está la de realizar la planificación y regulación del tránsito de vehículos de pasajeros entregando los permisos para que puedan prestar el servicio en sus variadas formas de prestación. Otra función fundamental que tiene este órgano es la de fomentar la planificación para el desarrollo de un servicio que brinde seguridad a los pasajeros, además que sea sostenible todo ello con el fin de mejorar la prestación del servicio contribuyendo en mejorar la calidad de vida de los que usan el servicio. También tiene como responsabilidad impulsar el orden e integración de todo el transporte urbano de pasajeros de la ciudad de Lima Metropolitana (26).

f) Empresa de transporte

Es una empresa dedica al transporte de pasajeros es la persona jurídica que se encuentra debidamente registrada e inscrita y, además, tiene permiso valido o tiene una concesión para realizar el servicio urbano de transportar personas por las vías autorizadas (21). En esta investigación denominaremos de esta forma a las compañías que otorgan este. Todas las empresas de esta actividad son gestionadas y evaluadas por la Municipalidad de Lima a través de la Gerencia de Transporte Urbano.

g) Administración de rutas en el transporte público

Capacidad de gestión de todo el sistema de transporte que incluye empresa de buses autorizadas por determinadas vías asignadas con los paraderos donde los pasajeros esperan usar el servicio.

h) Rutas

Son los caminos o vías de desplazamiento autorizados que permiten unificar lugares distintos y distantes. Estas vías contribuyen de manera ordenada a brindar el acceso a los usuarios del servicio de transporte público que requiere desplazarse entre dichos lugares. Durante el recorrido se pueden encontrar lugares autorizados para el recojo de pasajeros. La asignación de rutas esta administrada por la Gerencia de Transporte Urbano otorgando licencias a las empresas de transporte teniendo en cuenta la demanda, la cobertura y la frecuencia que los buses deben brindar el servicio.

Existen clasificación de rutas dependiendo de la intensidad de uso, velocidad, horarios y áreas en las cuales están ubicadas para su desplazamiento, rutas regionales para el desplazamiento de personas entre provincias o departamentos, rutas turísticas para el paseo y distracción y las rutas urbanas con paradas determinadas para dejar y recoger pasajeros.

i) Tráfico

Congestión de vehículos particulares, buses, personas, mercancías u otros medios de movimiento que usan las vías públicas.

j) Frecuencia

Se emplea para determinar las repeticiones de los buses con la misma ruta entre diferentes líneas o con fragmentos de rutas en común entre ellas.

k) Paradero

Lugar donde las personas esperan la llegada del bus. Este lugar es adecuado y seguro para las personas y los buses con ruta deben detenerse para que los pasajeros bajen o suban al bus. Los paraderos se encuentran ubicados estratégicamente en las rutas de los buses de tal forma que los usuarios puedan hacer uso del servicio (21).

l) Pasajero

Es el usuario que requiere hacer uso del servicio de transporte de pasajeros público para ello el pasajero realiza un pago todas las veces que usa el bus este pago le otorga el derecho de subir el bus y ser transportado de un determinado lugar a otro que es el destino. En las ciudades todas las personas son consideradas potenciales pasajeros (21).

m) Administración de horarios de transporte público

Es el modo de gestionar y regular las horas que se deben considerar para el servicio de transporte urbano de pasajeros tomando en cuenta la hora de inicio de recorrido del bus la llegada a cada paradero en su recorrida hasta la finalización total de su ruta.

n) Aplicativo móvil

Software diseñado y construido para de una determinada tarea específica y usando una plataforma diseñada para dispositivos móviles. Es posible obtenerla a través de una plataforma web e instalado en dispositivos móviles diversos como celulares, tablets o cualquier reproductor diseñado como móvil (27).

Las aplicaciones móviles, más conocidas comúnmente como “apps” son programas diseñados para ser ejecutados en teléfonos, tablets y otros dispositivos móviles, que permiten al usuario realizar prácticamente cualquier actividad, acceder a servicios, consultar o recibir noticias, comunicarse con otras personas, compartir sus experiencias, etc.

o) Tipos de aplicaciones web

Todas las Apps se ajustan a una clasificación específica dependiendo de su naturaleza:

App nativa: diseñada bajo un lenguaje de programación específico y está orientada a funcionar también bajo un sistema operativo determinado.

Web App: no se instalan en el dispositivo y se utilizan exclusivamente con el uso de internet. Suelen ser adaptaciones de páginas web al formato móvil.

Web App Nativa: también conocida como aplicación híbrida, se descarga de una app store y se instala en el dispositivo mostrando un icono. Dependiendo del tipo de aplicación, algunas requerirán de conexión a internet para funcionar y otras no.

Las aplicaciones móviles más utilizadas, son aquellas que se han ganado la preferencia del público, no solo por su apariencia sino por su usabilidad, buen funcionamiento, excelente diseño y la experiencia de usuario que son capaces de proveer, entre ellas tenemos: Facebook, WhatsApp, YouTube, Amazon, Google Maps, Instagram, Telegram y Gmail. (28)

Las compañías encargadas del desarrollo de aplicaciones han centrado sus esfuerzos en el diseño y en la programación de las aplicaciones móviles y son, además, intermediarios entre los verdaderos creadores y los consumidores. Los mayores fabricantes y distribuidores de aplicaciones Android (Google Play©) y Apple (iTunes Store©) en realizar las descargas de las apps desde sus respectivas tiendas. (29)

p) Usabilidad y funcionalidad

Son términos cualitativos, la usabilidad es entendida como la facilidad en el uso de un aplicativo o página web según la funcionalidad u objetivo esperado. También se puede referir a cualquier aparato o proceso realizado para mejorar o facilitar el cumplimiento de un objetivo. También se puede considerar como una medida de calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema. (30)

El instrumento de evaluación de usabilidad en dispositivos móviles está propuesto en base a una síntesis y adaptación entre las cuatro características elementales propuestas por la norma ISO 9241-11: Usuarios, tareas, equipo y entorno. En conjunto con las referencias bibliográficas más reconocidas para la evaluación de aplicaciones móviles de la actualidad como soporte para tomar en cuenta, y los principales aspectos que son tendencia en la evaluación de aplicaciones móviles y estudios realizados en diversas universidades alrededor del mundo para cuantificar el nivel de usabilidad en medida de un producto software al interactuar con el usuario. (31)

q) Desarrollo de software predictivo

Tradicionalmente todo proyecto se inicia con una planeación e identificación de requerimientos antes de entrar a una etapa detallada de su solicitud. Todas las necesidades son evidenciadas el inicio de esta manera se busca mantener la calidad final del producto enmarcada en un calendario aceptado y con poca posibilidad de modificación.

En este tipo de formas de desarrollar un software se tienen bien definidos los alcances y la secuencia que se aplicará para su desarrollo siempre en una única dirección evitando modificar cualquier elemento que vaya en contra de este precepto. Es por ello, que todas las necesidades se manifiestan al inicio y acordados en cumplirse durante el desarrollo y ejecución del proyecto esto forma disminuye la interacción con el cliente durante el proceso (32).

El desarrollo predictivo tiene marcadas diferencias con las otras metodologías y sirvió para realizar el comparativo con la metodología ágil descritas en el resumen de la (tabla 4).

r) Desarrollo de software ágil

Este tipo de proyectos dividen sus productos en entregables pequeños siguiendo una lista ordenada con requerimientos y funciones similares. Cada subproyecto se trabaja de manera independiente acortando el plazo de revisión y control de calidad, así como la entrega al cliente. Debido a esta forma la comunicación con el cliente o dueño del pequeño producto se realiza de manera interactiva y permanente generando retroalimentación permitiendo y aceptando los cambios durante todo el proceso de desarrollo. La entrega de estos pequeños productos se realiza en semanas y no es necesario esperar que todos los productos estén terminados se realizan entregas parciales generando satisfacción en el cliente (33).

s) Base de datos

Programa capaz de almacenar gran cantidad de información de manera relacionada y estructurada y permite consultarla.

t) **Diseño responsivo**

El diseño responsivo o adaptable sirve para que un usuario pueda entrar a una página web desde un ordenador o un teléfono móvil sin que su capacidad para navegar por ella se vea perjudicada. (34) busca conseguir siempre la mejor experiencia de usuario, independientemente del aparato que use para entrar a una página web, y no duda en modificar todo cuanto sea necesario para conseguirlo. (Figura 5)

Figura 5
Características del Diseño Responsivo



Fuente: (34)

El diseño responsivo permite reducir el tiempo de desarrollo, evita los contenidos duplicados, y aumenta la visualización de los contenidos ya que permite compartirlos de una forma mucho más rápida y natural.

u) **Scrum**

Es una forma o método para construir un aplicativo de manera ágil, es decir, rápidamente. Priorizando el desarrollo con iteraciones y entregas de partes del producto final. (Figura 6) SCRUM es considerado un marco de trabajo que se emplea como metodología ágil para el desarrollo de proyectos aplicando buenas prácticas y definiendo roles, con ello se puede tener un mejor manejo del proyecto con entregables semanales y tareas categorizadas por prioridad basados en ciclos (35). La metodología SCRUM gestiona estas iteraciones a través de reuniones cortas y diarias con el fin de evidenciar los problemas y mitigarlos. Por la flexibilidad que brinda SCRUM se procedió a la elaboración de este proyecto con dicha metodología.

Figura 6
Proceso Scrum Completo



Fuente: (36)

v) Product backlog:

Consiste en identificar y describir las actividades que son necesarias para completar adecuadamente un proyecto, estas actividades son escritas en un tablero denominado backlog que es visualizado por todos los interesados y ejecutores del proyecto.

w) Sprint backlog:

Es un registro o anotación de actividades que son desarrollados por todo el equipo en la etapa de planificación (Tabla 4). En esta lista cada responsable tiene una o más tareas o actividades y un determinado plazo para concluir las. Es así como bajo esta metodología se subdivide el trabajo en pequeñas partes por lo que se tiene un mejor control de los retrasos y los motivos.

Tabla 4
Ejemplo de Sprint Backlog

| Requisito | Tarea | Quien | Estado (No iniciada / en progreso / completada) | Dia: | | | | | | | | | | | | | | | | |
|-------------|----------|-------|--|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | | | | | |
| | | | | Horas pendientes | | | | | | | | | | | | | | | | |
| Requisito A | Tarea 1 | Joao | Completada | | 16 | 8 | | | | | | | | | | | | | | |
| Requisito A | Tarea 4 | Laura | Completada | | 4 | | | | | | | | | | | | | | | |
| Requisito A | Tarea 5 | Laura | Completada | | 4 | | | | | | | | | | | | | | | |
| Requisito A | Tarea 3 | Gabri | Completada | | 8 | | | | | | | | | | | | | | | |
| Requisito A | Tarea 2 | Laura | Completada | | 16 | 8 | 4 | | | | | | | | | | | | | |
| Requisito A | Tarea 6 | Gabri | Completada | | 8 | 8 | 8 | | | | | | | | | | | | | |
| Requisito A | Tarea 7 | Joao | Completada | | 16 | 16 | 16 | 8 | | | | | | | | | | | | |
| Requisito A | Tarea 8 | Laura | Completada | | 8 | 8 | 8 | | | | | | | | | | | | | |
| Requisito A | Tarea 9 | Laura | Completada | | 8 | 8 | 8 | 8 | 8 | | | | | | | | | | | |
| Requisito A | Tarea 10 | Laura | Completada | | 8 | 8 | 8 | 8 | 8 | 8 | 4 | | | | | | | | | |
| Requisito A | Tarea 11 | Joao | Completada | | 16 | 16 | 16 | 16 | 16 | 16 | 8 | | | | | | | | | |
| Requisito B | Tarea 12 | Gabri | Completada | | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 8 | | | | | | |
| Requisito B | Tarea 13 | Laura | Completada | | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 8 | | | | | | |
| Requisito B | Tarea 14 | Joao | En progreso | | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 4 | | | | | |
| Requisito B | Tarea 15 | Gabri | En progreso | | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | | | | |
| Requisito B | Tarea 16 | Laura | En progreso | | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | | | | |
| Requisito C | Tarea 17 | Joao | No iniciada | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| Requisito C | Tarea 18 | Gabri | No iniciada | | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Requisito C | Tarea 19 | Laura | No iniciada | | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| Requisito C | Tarea 20 | Joao | No iniciada | | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |

Fuente: (36)

x) Framework

Grupo de código denominado librerías desarrolladas con el fin que se puedan utilizar de manera general en otros códigos de desarrollo cuando sean necesarias.

y) Experiencia de usuario - UX

La Experiencia de usuario es todo aquel proceso que el usuario efectúa cuando interactúa con un producto y/o servicio. De cara a la compañía, se trata del procedimiento con el que se analizan las acciones que lleva a cabo este consumidor para valorar qué es lo más y menos atractivo para este, o si logra usarlo adecuadamente, si se encuentran complicaciones y un largo etcétera.

Se trata de un término nacido bajo el ala del marketing digital y que guarda una estrechísima relación con la experiencia de marca. Siendo esta última la intención de la compañía de estrechar lazos con sus consumidores para conseguir una relación larga, duradera y fructífera; es fácil deducir que la experiencia positiva funciona a la hora de conseguir ese objetivo tan deseado y, además, para conseguir que la imagen de la firma sea mucho mejor.

z) Efectividad, eficiencia y satisfacción

Por efectividad se entenderá la precisión y la plenitud con las que los usuarios alcanzan los objetivos especificados. A esta idea van asociadas la facilidad de aprendizaje (en la medida en que este sea lo más amplio y profundo posible), la tasa de errores del sistema y la facilidad del sistema para ser recordado (que no se olviden las funcionalidades ni sus procedimientos).

Por eficiencia se entenderán los recursos empleados en relación con la precisión y plenitud con que los usuarios alcanzan los objetivos especificados. A esta idea van asociadas la facilidad de aprendizaje (en tanto que supone un coste en tiempo; igualmente, si se requiere un acceso continuo a los mecanismos de ayuda del sistema), la tasa de errores del sistema y la facilidad del sistema para ser recordado (una asimilación inapropiada puede traducirse en errores de usuario).

Por satisfacción se entenderá la ausencia de incomodidad y la actitud positiva en el uso del producto. Se trata, pues, de un factor subjetivo.

CAPÍTULO III: METODOLOGÍA

3.1 Metodología para la gestión del proyecto

Para la gestión del proyecto se usó el estándar para la Dirección de Proyectos PMI que está avalada por el Instituto Nacional de Normalización de los Estados Unidos (ANSI) y que fue desarrollado utilizando un proceso basado en los conceptos de consenso, apertura, debido proceso y equilibrio.

Según la Guía de los Fundamentos para la Dirección de Proyectos (PMBOK - 2017) (37) “La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos de este. Se logra mediante la aplicación e integración adecuadas de los procesos de la dirección de proyectos, agrupados lógicamente, que conforman los grupos de procesos básicos.”

La guía del PMBOK comprende dos grandes secciones:

- Áreas de conocimiento específico para la gestión de un proyecto.
- Grupos de Procesos básicos.

El PMBOK (37) reconoce 5 grupos de procesos básicos y 9 áreas de conocimiento comunes a casi todos los proyectos.

3.1.1 Grupos de procesos básicos

- *Iniciación*: Define y autoriza el proyecto o una fase de este. Está formado por dos procesos: Objetivos y Alcance del proyecto.
- *Planificación*: Define, refina los objetivos y planifica el curso de acción requerido para lograr los objetivos y el alcance pretendido del proyecto.

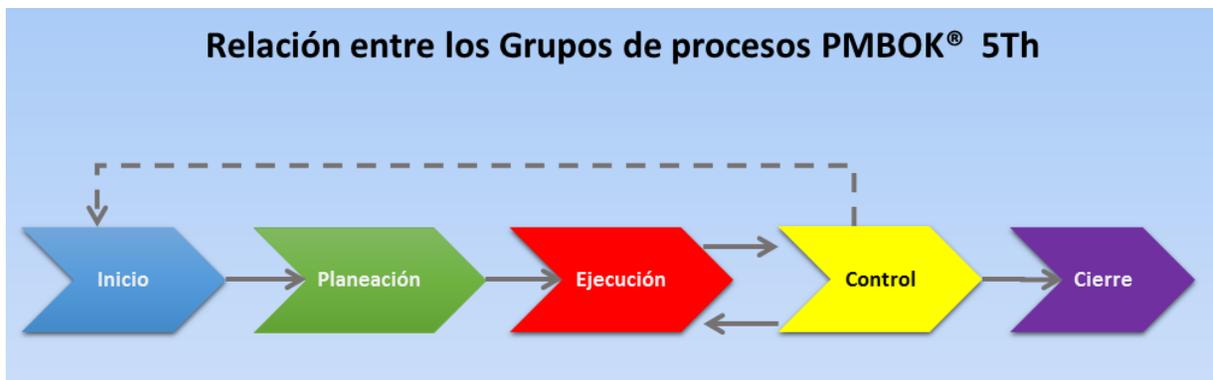
- *Ejecución:* Compuesto por aquellos procesos realizados para completar el trabajo definido en el plan a fin de cumplir con las especificaciones de este. Implica coordinar personas y recursos, así como integrar y realizar actividades del proyecto en conformidad con el plan para la dirección del proyecto.
- *Seguimiento y Control:* Mide, supervisa y regula el progreso y desempeño del proyecto, para identificar áreas en las que el plan requiera cambios.
- *Cierre:* Formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto o una fase de este.

3.1.2 Áreas de conocimiento del proyecto

1. *Gestión del Alcance del Proyecto:* describe los procesos necesarios para asegurarse que el proyecto incluya todo el trabajo requerido para completarse satisfactoriamente.
2. *Gestión del Tiempo del Proyecto:* describe los procesos relativos a la puntualidad en la conclusión del proyecto.
3. *Gestión de los Costes del Proyecto:* describe los procesos involucrados en la planificación, estimación, presupuesto y control de costes de forma que el proyecto se complete dentro del presupuesto aprobado.
4. *Gestión de la Calidad del Proyecto:* describe los procesos necesarios para asegurarse de que el proyecto cumpla con los objetivos para los cuales ha sido emprendido.
5. *Gestión de los Recursos Humanos del Proyecto:* describe los procesos para organizar y dirigir los Recursos Humanos del proyecto.
6. *Gestión de las Comunicaciones del Proyecto:* describe los procesos relacionados con la generación, distribución almacenamiento y destino final de la información del proyecto en tiempo y forma.

7. **Gestión de la Integración del Proyecto:**
describe los procesos y actividades que forman parte de los diversos elementos de la Dirección de Proyectos, que se identifican, definen, combinan, unen y coordinan dentro de los Grupos de Procesos de Dirección de Proyectos.
8. **Gestión de los Riesgos del Proyecto:** describe los procesos relacionados con el desarrollo de la gestión de riesgos de un proyecto.
9. **Gestión de las Adquisiciones del Proyecto:** describe los procesos para comprar o adquirir productos, servicios o resultados, así como para contratar procesos de dirección.

Figura 7
Ciclo de vida del grupo de Procesos según el PMBOK



Fuente: (37)

3.1.3 Grupo de proceso de inicio

En esta parte del proceso se definió el proyecto, el alcance, los principales entregables, limitaciones y los criterios de éxito del proyecto.

Para esta tesis se desarrolló el documento referido al acta de constitución en el cual se describieron los aspectos formales del proyecto, así como sus principales condiciones de aceptación. Además, se definieron los requerimientos de alto nivel, las fases que contempla el desarrollo y se identificó a los interesados.

3.1.4 Grupo de proceso de planificación

En esta segunda etapa se definieron los documentos referidos a las tareas, actividades, recursos y tiempo tales como el Plan de actividades y desarrollo del cronograma y la estructura desglosable del trabajo (EDT). El cronograma como el EDT se desarrollan acorde a la metodología de desarrollo.

3.2 Metodología aplicada para el desarrollo de software (Scrum)

El marco de trabajo que se usó para estructurar, planificar y controlar el proceso de desarrollo de esta tesis es Scrum. Para ello, se contó con un programador especialista en desarrollo de aplicaciones móviles y un personal de control de calidad y revisión del aplicativo.

Una metodología es un conjunto de procedimientos que se estructuran y realizan en función de conseguir un objetivo. Existen varios tipos de metodologías orientadas al desarrollo de software entre las cuales tenemos las predictivas que utilizan un diseño de procesos secuencial y las llamadas ágiles que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta, dentro de las cuales podemos mencionar Scrum y programación extrema (XP).

En la siguiente (Tabla 6) exponemos de manera resumida las principales diferencias entre la metodología predictiva y la ágil:

Tabla 5
Matriz de Comparación de Metodologías

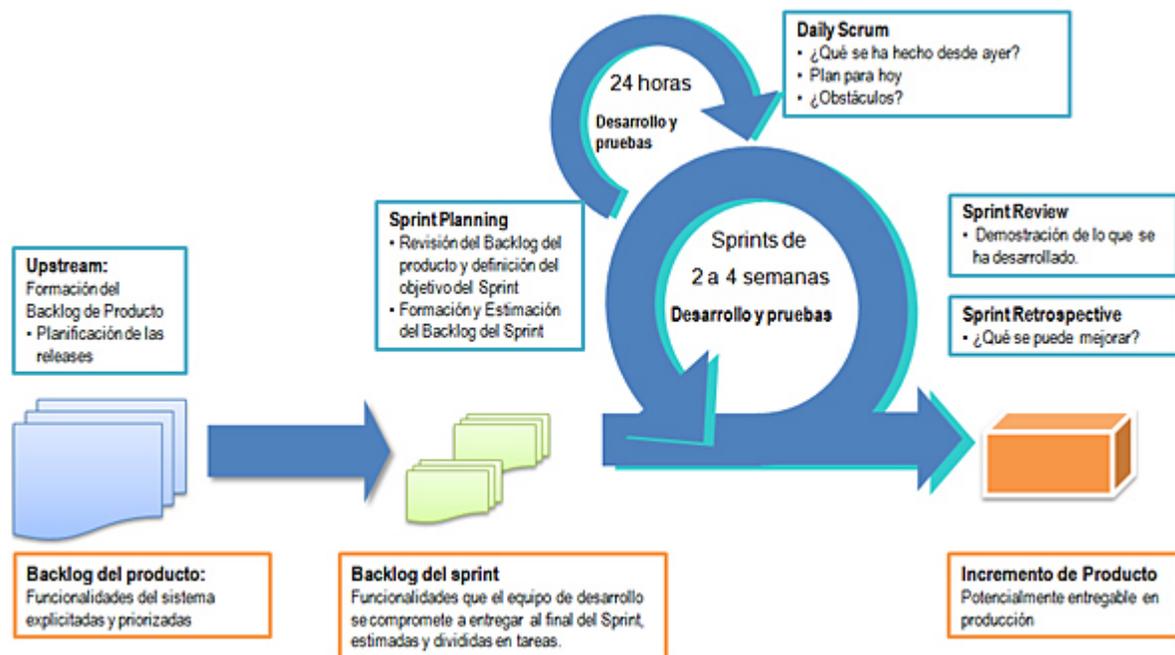
| Metodología predictiva | Metodologías ágiles |
|-------------------------------|--|
| Predictivos | Adaptativos |
| Orientados a procesos | Orientados a personas |
| Actividades rigurosas | Actividades y tareas flexibles |
| Es desarrollado como proyecto | Se divide el proyecto y partes pequeñas. |

| | |
|---|---|
| Poca comunicación con el cliente | Permanente comunicación con los involucrados. |
| El producto se entrega cuando se concluye el desarrollo | Entregas constantes de software |
| Documentación extensa | Poca documentación |

Fuente: (33)

Dentro del marco metodológico que fue aplicado es importante diferencia entre el Ciclo de procesos de gestión de proyectos (Figura 7) y el Ciclo de vida del desarrollo de software (Figura 8); el primero puede iniciar o finalizar de manera independiente al comienzo de la gestión, el segundo puede concluir antes de tener el producto terminado.

Figura 8
Ciclo de vida del desarrollo de Software – Metodología Scrum



Fuente: (36)

3.2.1 Componente Scrum

Para Metodología Scrum, 2019 (36) los componentes de Scrum son los siguientes:

3.2.1.1 **Las reuniones:**

Se definirá un documento en el que se reflejaran los requisitos del sistema por prioridades. En esta fase se determinará también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración.

Se obtendrá además en esa reunión un Sprint Backlog, registro de actividades o tareas y que son el fin más importante del sprint.

Las reuniones diarias como elementos son importantes ya que se organizan diversas tareas con cierto grado de prioridad de acuerdo con las necesidades de los usuarios y se discuten todos los inconvenientes que se hayan tenido hasta el momento

3.2.1.2 **Los roles**

Dueño del producto: Es el responsable directo y quien toma decisiones y asume la responsabilidad. Conoce la necesidad del cliente y del negocio por lo que se encarga de describir correctamente la necesidad real.

Maestro Scrum: Tiene como función verificar que modelo y prototipo funcione de acuerdo con el modelo y metodología establecida. También debe retirar cualquier inconveniente que se presente con el objetivo que el proceso continúe.

Equipo De Desarrollo: suele ser un equipo pequeño de unas 5-9 personas y tienen capacidad y decisión para planificar decidir acciones para conseguir la meta trazada. Está involucrado en la planificación de la actividades y tareas del Backlog.

Usuarios: Es el destinatario final del producto.

Stakeholders: Son todas las personas que serán beneficiados con la culminación e implementación del producto final. Tienen participación y capacidad de decisión en cualquier momento durante el desarrollo del producto.

Managers: Toma las decisiones finales participando en la selección de los objetivos y de los requisitos.

3.1.4.1 Elementos scrum

Para el autor (36) los elementos scrum son los siguientes:

- **Product Backlog:** lista de necesidades del cliente.
- **Sprint Backlog:** lista de tareas que se realizan en un Sprint.
- **Incremento:** parte añadida o desarrollada en un Sprint, es una parte terminada y totalmente operativa.

Las iteraciones son uno de los principales elementos de la metodología scrum ya que permite una mayor satisfacción al usuario final respecto a sus necesidades.

3.1.4.2 Product backlog

Para el autor del libro (36) el Product Backlog es el inventario en el que se almacenan todas las funcionalidades o requisitos en forma de lista priorizada (Tabla 4). Estos requisitos serán los que tendrá el producto o los que irá adquiriendo en sucesivas iteraciones. La lista será gestionada y creada por el cliente con la ayuda del SCRUM Máster, quien indicará el coste estimado para completar un requisito, y además contendrá todo lo que aporte un valor final al producto. Las tres características principales de esta lista de objetivos serán los siguientes:

1. Contendrá los objetivos del producto, se suele usar para expresarlos las historias de usuario.
2. En cada objetivo, se indicará el valor que le da el cliente y el coste estimado; de esta manera, se realiza la lista, priorizando por valor y coste, se basará en el ROI.
3. En la lista se tendrán que indicar las posibles iteraciones y las versiones de prueba que se han indicado al cliente.
4. La lista ha de incluir los posibles riesgos e incluir las tareas necesarias para solventarlos.

Es necesario que antes de empezar el primer Sprint se definan cuáles van a ser los objetivos del producto y tener la lista de los requisitos ya definida. No es necesario que sea muy detallada, simplemente deberá contener los requisitos principales para que el equipo pueda trabajar. Realizar este orden de tareas tiene como beneficios:

El proyecto no se paraliza simplemente por no tener claro los requisitos menos relevantes, y el cliente podrá ver resultados de forma más rápida.

Los requisitos secundarios aparecerán a medida que se va desarrollando el proyecto, por lo tanto, no se pierde tanto tiempo en analizarlos al principio y el cliente será más consciente de sus necesidades.

Los requisitos secundarios puede que no se lleguen a necesitar porque se han sustituido o porque no reportan un retorno ROI interesante. Una vez definidos los requisitos se tendrá que acordar cuándo se tiene que entender un objetivo como terminado o completado. Se entiende que un producto está completado si:

- Asegura que se puede realizar un entregable para realizar una demostración de los requisitos y ver qué se han cumplido.
- Incluirá todo lo necesario para indicar que se está realizando el producto que el cliente desea.

Tabla 6
Ejemplo de Product Backlog

| Id | Prioridad | Descripción | Est. | Por. |
|-----------|------------------|---|-------------|-------------|
| 1 | Muy Alta | Plataforma tecnológica | 30 | AR |
| 2 | Muy Alta | Interfaz de Usuario | 40 | LR |
| 3 | Muy Alta | Un usuario se registra al sistema | 40 | LR |
| 4 | Alta | El operador define el flujo de texto de un expediente | 60 | AR |
| 5 | Alta | Etc.... | 999 | XX |

Fuente: (36)

3.1.5 Historia de usuarios:

Para Trigas (36) esto es la narración verbal de las personas que forman parte de la operatividad del proceso convirtiéndolas luego en funcionalidades del aplicativo que podrán ser modificadas posteriormente.

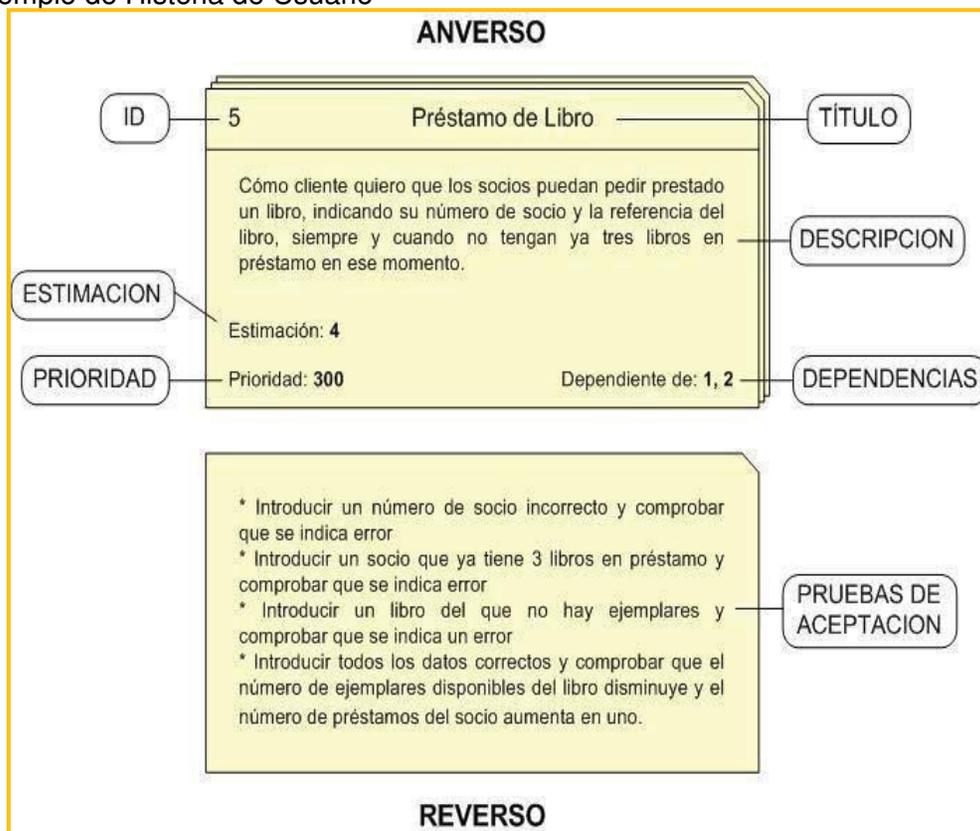
Las historias de usuario se componen de tres fases denominadas “Las 3 C”:

Card: Será una breve descripción escrita que servirá como recordatorio (Figura 9).

Conversation: Es una conversación que servirá para asegurarse de que se ha entendido bien todo, y concretar el objetivo.

Confirmation: Revisiones funcionales para fijar detalles que sean relevantes e indicar cuál va a ser el límite.

Figura 9
Ejemplo de Historia de Usuario



Fuente: (36)

3.3 Población y muestra

La población está conformada por todos los usuarios del servicio de transporte público de pasajeros en Lima Metropolitana. Como se ha mencionado anteriormente el 75% de los Limeños utilizan el servicio de transporte público en sus diferentes modalidades, según el INEI para el 2018 Lima tenía 9.320.000 habitantes por extrapolación esto representa un total de 6.990.000 pasajeros de transporte público de este grupo, el 25% viaja en buses aproximadamente 1.747.500 pasajeros para el año 2018.

Debido a que esta Tesis está enfocada en el desarrollo de un aplicativo móvil que evalúe el desempeño de cumplimiento de gestión del transporte urbano de pasajeros y que actualmente no se cuenta con nuevas cifras publicadas por el INE se seleccionó una muestra poblacional por juicio y opinión según la accesibilidad de la información.

CAPÍTULO IV: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

4.1 Identificación de requerimientos

4.1.1 Planificación del proyecto

Tabla 7: Acta del Proyecto

| Acta de Constitución del Proyecto | | | | | |
|---|--|--------|----|----|------|
| Proyecto: | Aplicativo Móvil para el Registro de Desempeño y Cumplimiento del Servicio de las Empresas de Transporte Público de Pasajeros | | | | |
| Patrocinador: | Bachiller en Ing. De Sistemas Eduardo Ydrogo Ramírez | | | | |
| Preparado por: | Bach. Eduardo Ydrogo Ramírez | Fecha: | 01 | 02 | 2019 |
| Revisado por: | Ing. Eduardo Daniel Ricse Medina | Fecha: | 02 | 03 | 2019 |
| Breve descripción del Producto o Servicio del Proyecto | Aplicativo Móvil que permita el registro de las incidencias durante el servicio de transporte urbano de pasajeros. | | | | |
| Alineamiento del Proyecto | | | | | |
| Objetivo del Proyecto: | Desarrollar un aplicativo móvil que permita registrar y mostrar información del cumplimiento y desempeño del servicio brindado por las empresas de transporte público de pasajeros en la ciudad de Lima. | | | | |

| | |
|--|---|
| Alcance: | Desarrollar los prototipos, pantallas y funcionalidades indicadas en las historias de usuarios. |
| Limitaciones: | No se implementará ni se difundirá el aplicativo desarrollado. |
| Entregables: | Acta de constitución. Cronograma del Proyecto Estructura Desglosable del trabajo Prototipo y Desarrollo de Aplicativo. |
| Factores Críticos: | El Proyecto debe desarrollar en cuatro meses. |
| Requerimientos de Alto nivel: | No exceder el tiempo acordado. |
| Requisitos relacionados al objetivo del proyecto: | Permite grabar información sobre el estado de limpieza del bus. Grabar información sobre el servicio prestado por el conductor o cobrador. Permitir grabar las incidencias ocurridas durante la prestación del servicio referida a las rutas y paraderos. Grabar información sobre información de ocurrencias tales como se pasó la luz roja. Mostrar información a los usuarios sobre las incidencias ocurridas del bus. |
| Extensión y Alcance del Proyecto | |
| Fases del Proyecto: | Identificación de los requisitos Diseño de los Prototipos Desarrollo de los componentes de registro de incidencias y de reportes del prototipo Testear el prototipo y los componentes. |
| Interesados: | Patrocinador |
| Involucrados: | Patrocinador Jefe del proyecto |
| Riesgos: | No concluir a tiempo el desarrollo del aplicativo móvil. |
| Fecha inicio: | 04-03-2019 |
| Fecha fin: | 26-07-2019 |

| | |
|----------------------------------|---------------|
| Presupuesto: | S/1,200 soles |
| Responsable del Proyecto: | Patrocinador |

Fuente: Elaboración propia

4.1.2 Cronograma del proyecto

Se preparó el cronograma del proyecto para un máximo de ciento cinco días.

Tabla 8
Plan de Actividades del Proyecto

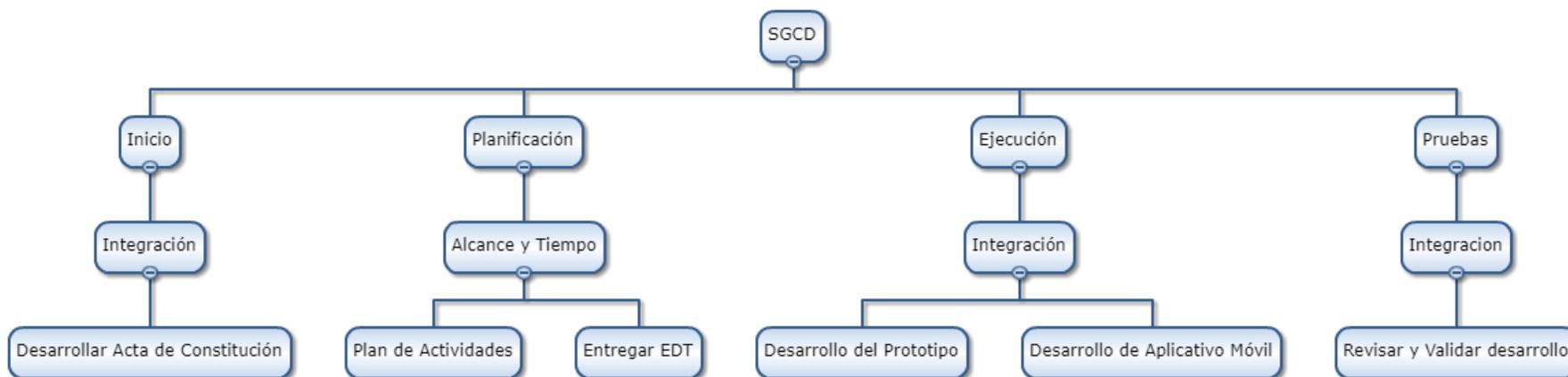
| Tarea | Duración | Comienzo | Fin | Entregable |
|----------------------------------|-----------------|-----------------|--------------|--------------------------------------|
| SGCD | 105 días | lun 04/03/19 | vie 26/07/19 | |
| Inicio | 30 días | lun 04/03/19 | vie 12/04/19 | |
| Desarrollar Acta de Constitución | 30 días | lun 04/03/19 | vie 12/04/19 | Acta de Constitución |
| Planificación | 14 días | lun 15/04/19 | jue 02/05/19 | |
| Alcance y Tiempo | 14 días | lun 15/04/19 | jue 02/05/19 | |
| Preparación del EDT | 4 días | lun 29/04/19 | jue 02/05/19 | EDT |
| Ejecución | 50 días | lun 29/04/19 | vie 05/07/19 | |
| Desarrollo del Prototipo | 20 días | lun 29/04/19 | vie 24/05/19 | Prototipos diseñados |
| Desarrollo del Aplicativo Móvil | 30 días | lun 27/05/19 | vie 05/07/19 | Código de Programación desarrollados |
| Pruebas y Entrega | 15 días | lun 8/07/19 | vie 26/07/19 | |
| Revisar y validar el desarrollo | 15 días | lun 8/07/19 | vie 26/07/19 | Resultados |

Fuente: Elaboración Propia

4.1.3 Estructura desglosable del trabajo

Esta es una herramienta imprescindible que consiste en la descomposición jerárquica del trabajo a ser ejecutado por el equipo de proyecto, para cumplir con los objetivos de éste y crear los entregables requeridos, donde cada nivel descendente de la EDT representa una definición con un detalle incrementado del trabajo del proyecto.

Figura 10
Estructura Desglosable del Trabajo



Fuente: Elaboración Propia.

4.2 Alcance de la solución

Se desarrolló un aplicativo móvil que permita a los usuarios registrar las incidencias ocurridas durante el uso del servicio de transporte público de pasajero en apoyo a la gestión de cumplimiento y desempeño de las empresas de transporte público.

La solución tecnológica se denominó “SGCDTP – Sistema de Gestión de Cumplimiento y Desempeño del Transporte Público”. Es un aplicativo móvil para los pasajeros que hacen uso del servicio de transporte público en la ciudad de Lima Metropolitana y que les permitió registrar información sobre las incidencias ocurridas durante la prestación del servicio.

La solución se construyó en base a los componentes de un aplicativo móvil que permitió el registro de los pasajeros y luego se les brindó el acceso mediante su usuario y clave para que puedan registrar la información relevante sobre las incidencias ocurridas durante el uso del servicio de transporte público de pasajeros. Además, la solución permitió visualizar en un dispositivo móvil una gráfica de incidencias reportadas según el bus de transporte con el fin que el usuario pueda tener información y tomar decisión sobre el particular.

4.3 Etapas en el proceso de construcción

Aplicando los conceptos anteriormente expuestos, se procedió a definir las fases mediante las cuales se desarrolló el aplicativo según la Metodología Scrum (Tabla 9).

Tabla 9
Fases de la Metodología Ágil

| Fases | |
|---------------------------------|-----------------------------|
| Project Charter Planificación | Elaboración del documento |
| | Entrega del documento |
| Plan de Cronograma del Proyecto | Elaboración del Cronograma |
| | Verificación del Cronograma |
| | Entrega del EDT |
| | Aprobación del EDT |

| | |
|-------------------------------|---|
| | Elaboración de prioridades |
| Lista de Productos | Identificación de Historias de usuarios |
| | Descripción de Historias de Usuarios |
| | Priorización de Historias de Usuarios |
| | Verificación de Historia de Usuario |
| Prototipo Producto- Sprint | Diseño de Mockups |
| Sprint 1 | HU-001 |
| | HU-002 |
| | HU-003 |
| | HU-004 |
| | HU-005 |
| | HU-006 |
| | HU-007 |
| | HU-008 |
| | HU-009 |
| | HU-010 |
| | HU-011 |

Fuente: Elaboración Propia.

4.4 Requerimientos no funcionales

En la siguiente (Tabla 10) se describen las características generales y restricciones del aplicativo móvil, con la finalidad de especificar criterios que pueden usarse para juzgar la operatividad de este.

Tabla 10
Requerimientos No Funcionales

| Número | Atributo | Solicitudes |
|-----------|---------------------------|---|
| RQ-NF-001 | Seguridad | El aplicativo debe ser confiable para todos los usuarios y administradores. Debe garantizar el acceso a la información sea autorizada y segura. |
| RQ-NF-002 | Usabilidad | La interfaz del aplicativo móvil debe ser intuitiva y sencilla. |
| RQ-NF-003 | Consistencia y Coherencia | El desarrollo debe garantizar coherencia y consistencia de la información tanto al momento de ser almacenados como al momento ser modificados por cualquier usuario y en cualquier momento. Además, el aplicativo debe permitir registro y modificación simultánea. |

Fuente: Elaboración Propia

4.5 Historias de usuarios

A continuación (Tabla 11), se describen los diferentes escenarios que conformaron al aplicativo, identificando a sus Actores, nivel de prioridad y riesgo, asignando a un responsable de la ejecución y detallando la historia y las condiciones empleadas según la metodología Scrum.

Tabla 11
Historias de Usuario para el aplicativo móvil

| | |
|---|--------------------------------|
| Historia: 01 | Actor: Usuario/Pasajero |
| Título: Registrar Usuario al Aplicativo | |
| Prioridad: Alta | |
| Riesgo: Bajo | |
| Responsable: Bach. Eduardo Ydrogo | |
| Historia: Se pedirá los datos de Nombre, Apellido Paterno, Apellido Materno, DNI, número de celular, Usuario, y clave de acceso. | |
| Condiciones: La clave deberá tener una longitud de entre 6 y 12 caracteres. | |

Historia: 02 **Actor:** Usuario/Pasajero

Título: Usuario recupera sus datos de acceso

Prioridad: Alta

Riesgo: Medio

Responsable: Bach. Eduardo Ydrogo

Historia: Debe existir una opción para recuperar los datos de acceso al aplicativo ingresando el número de celular con que se registró.

Condición: Usuario y Clave serán enviados al celular a través de un mensaje de texto.

Historia: 03 **Actor:** Usuario/Pasajero

Título: Usuario Ingresa al Aplicativo

Prioridad: Alta

Riesgo: Medio

Responsable: Bach. Eduardo Ydrogo

Historia: Se puede acceder al aplicativo ingresando los datos de usuario y clave

Condición: Debe existir la opción de “¿Olvido sus datos?” y la opción de “Registrarse” en caso de que el usuario no recuerde sus datos de acceso o no se encuentre registrado.

Historia: 04 **Actor:** Usuario/Pasajero

Título: Menú de opciones del usuario o pasajero

Prioridad: Alta

Riesgo: Alto

Responsable: Bach. Mario Zúñiga

Historia: El aplicativo debe mostrar opciones al usuario para registrar una incidencia y para ver la información de calificación del bus.

Condición: La Pantalla debe tener un botón de “cerrar” para volver a la pantalla anterior.

Historia: 05 **Actor:** Usuario/Pasajero

Título: Registro de Incidencia

Prioridad: Alta

Riesgo: Alto

Responsable: Bach. Eduardo Ydrogo

Historia: El aplicativo debe permitir registrar la incidencia ocurrida durante el viaje en el bus. Las incidencias que se registrarán serán las siguientes:

Se pasó la luz roja, no respeto los paraderos, el vehículo estaba sucio y mal comportamiento del conductor o cobrador. El usuario puede escoger más de una incidencia. El usuario debe registrar las placa, fecha y hora de la incidencia.

Se debe permitir que el usuario registre otra incidencia diferente a las predeterminadas describiéndolas en un máximo de tres líneas o agregar comentarios.

Condición: Las incidencias indicadas en esta historia como predeterminadas son las visibles, percentibles y de mayor ocurrencia durante la prestación del servicio de transporte público. Con el fin de reducir el tiempo de la interacción entre el usuario y el aplicativo las incidencias podrán ser seleccionadas haciendo "Check".

Historia: 06

Actor: Usuario/Pasajero

Título: Usuario puede visualizar información del bus

Prioridad: Alta

Riesgo: Alto

Responsable: Bach. Eduardo Ydrogo

Historia: El aplicativo debe permitir visualizar información referente a un bus de una línea de transporte. El aplicativo debe mostrar el número de faltas registradas.

Condición: Incluir íconos que ayuden a la visualización de la información junto con el número de incidencias reportadas.

Historia: 07

Actor: Jefe de Proyecto

Título: Institución Reguladora Visualiza Información de Incidencias.

Prioridad: Media

Riesgo: Bajo

Responsable: Bach. Eduardo Ydrogo

Historia: Se ingresa el RUC de la empresa para generar un reporte informativo sobre las incidencias ocurridas durante la prestación del servicio de transporte público de pasajeros.

Condición: El diseño deberá contener la incidencia y el totalizado por tipo de incidencias. El reporte debe emitirse por empresa.

Historia: 08

Actor: Jefe de Proyecto

Título: Registro de Usuarios Fiscalizadores.

Prioridad: Alta

Riesgo: Bajo

Responsable: Bach. Eduardo Ydrogo

Historia: Se pedirá los datos de Nombre, Apellido Paterno, Apellido Materno, DNI, número de celular, Usuario, y clave de acceso.

Condiciones: La clave deberá tener una longitud de entre 6 y 12 caracteres.

Historia: 09

Actor: Jefe de Proyecto

Título: Registro de Empresas de Transporte.

Prioridad: Alta

Riesgo: Bajo

Responsable: Bach. Eduardo Ydrogo

Historia: Se pedirá los datos de RUC, Descripción y Dirección

Condición: La Pantalla debe tener un botón de "Grabar" para guardar en la BD los datos ingresados y otro botón de "cerrar" para volver a la pantalla anterior.

Historia: 10

Actor: Jefe de Proyecto

Título: Registro de Buses.

Prioridad: Alta

Riesgo: Bajo

Responsable: Bach. Eduardo Ydrogo

Historia: se pedirá el campo clave: RUC de la empresa de transporte, adicionalmente se pedirá los datos de identificación del Bus tales como: Placa, Número de Asientos, Conductor, Cobrador y Ruta.

Condición: se le dará la opción de Buscar y automáticamente se extraerá da la base de datos el nombre de la empresa correspondiente al RUC ingresado.

Historia: 11

Actor: Jefe de Proyecto

Título: Registro de Problemas.

Prioridad: Alta

Riesgo: Bajo

Responsable: Bach. Eduardo Ydrogo

Historia: Se pedirá el único dato de Descripción.

Condición: El id del problema será asignado y almacenado de forma automática en la BD.

Fuente: Elaboración propia

4.6 Planificación de Sprints

Tabla 12
Sprint Planning

| Historia Usuario | Enunciado | Tareas | Responsable | Salida | Estado | Horas | Prioridad |
|------------------|---------------------------------|------------------------------|----------------------------|-----------------------------------|-----------|-------|-----------|
| HU-01 | Registrar Usuario al Aplicativo | Crear Pantalla | Programador | Pantalla de Registro de Usuario | Terminado | 12 | Alta |
| | | Probar la Grabación de datos | Administrador del Proyecto | | | | |
| HU-02 | Recuperar Clave de Usuario | Crear Pantalla | Programador | Pantalla de Recuperación de Clave | Terminado | 6 | Alta |
| HU-03 | Usuario Ingresa al Aplicativo | Crear Pantalla | Programador | Pantalla inicial | Terminado | 4 | Alta |
| HU-04 | Visualizar Menú de Opciones | Crear Pantalla | Administrador del Proyecto | Pantalla muestra la información | Terminado | 2 | Media |

| Historia Usuario | Enunciado | Tareas | Responsable | Salida | Estado | Horas | Prioridad |
|-------------------------|---|---------------------------|----------------------------|---|---------------|--------------|------------------|
| | para el Usuario | | | | | | |
| HU-05 | Usuario Registrar Incidencia Ocurrida Durante el Viaje | Crear Pantalla | Programador | Pantalla de Registro de Incidencia | Terminado | 4 | Alta |
| | | Probar y Validar registro | Administrador del Proyecto | | | | |
| HU-06 | Usuario Visualiza Información del Bus | Diseñar el reporte | Programador | Pantalla de reporte | Terminado | 12 | Alta |
| HU-07 | Institución Reguladora Visualiza Información de Incidencias | Diseñar el reporte | Programador | Pantalla de reporte | Terminado | 12 | Alta |
| HU-08 | Registrar Usuario Fiscalizador | Crear Pantalla | Programador | Pantalla de Registro de Usuario Fiscalizador | Terminado | 6 | Alta |
| | | Probar y Validar registro | Administrador del Proyecto | | | | |
| HU-09 | Registrar Empresa de Transporte | Crear Pantalla | Programador | Pantalla de Registro de Empresa de Transporte | Terminado | 6 | Alta |
| | | Probar y Validar registro | Administrador del Proyecto | | | | |

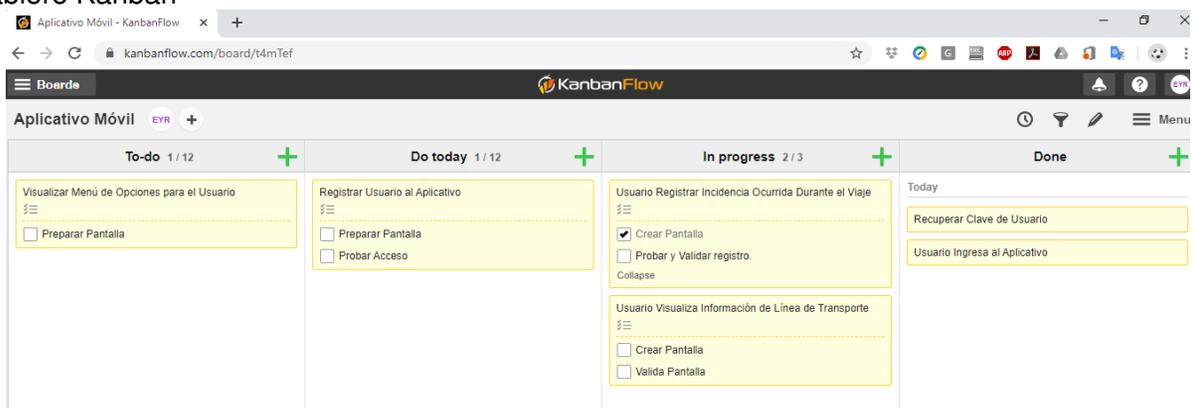
| Historia Usuario | Enunciado | Tareas | Responsable | Salida | Estado | Horas | Prioridad |
|------------------|---------------------|---------------------------|----------------------------|----------------------------------|-----------|-------|-----------|
| HU-10 | Registrar Buses | Crear Pantalla | Programador | Pantalla de Registro de Buses | Terminado | 6 | Alta |
| | | Probar y Validar registro | Administrador del Proyecto | | | | |
| HU-11 | Registrar Problemas | Crear Pantalla | Programador | Pantalla de Registro de Problema | Terminado | 4 | Alta |
| | | Probar y Validar registro | Administrador del Proyecto | | | | |

Fuente: Elaboración Propia.

4.7 Tablero de seguimiento

Se utilizó el tablero web Kanban Flow para hacer seguimiento y control a las tareas programadas. Dicho tablero nos permitió planificar, organizar y gestionar las tareas de desarrollo del aplicativo móvil, esta herramienta se encuentra disponible de forma gratuita en la web.

Figura 11
Tablero Kanban



Fuente: Captura de pantalla del aplicativo móvil – Kanban Flow.

4.8 Análisis de la solución

En esta etapa del proceso se diseñaron las pantallas correspondientes a las diferentes interfaces de usuario que conformaron al aplicativo (Tablas 14 hasta 24)

Tabla 13
Diseño de Pantalla - Interface Registrar Usuario al Aplicativo

| Código | Interfaz | Criterios de Aceptación |
|--|--|--|
| <p>HU-01: Registrar Nuevo Usuario al Aplicativo.</p> |  | <p>Validar que todos los campos estén llenos.</p> <p>Validar que el campo celular tenga los dígitos completos.</p> <p>Validar que el campo celular sea numérico.</p> <p>Validar que el campo Cédula sea sólo numérico.</p> <p>Grabar la información en la base de datos.</p> |

Fuente: Elaboración propia.

Tabla 14
Diseño de Pantalla - Interface Recuperar Datos de usuario

| Código | Interfaz | Criterios de Aceptación |
|-----------------------------------|---|---|
| HU-02: Recuperar Datos de Usuario |  | <p>Validar que el campo celular esté lleno y sea sólo numérico.</p> <p>Realizar la búsqueda de los datos en la BD.</p> <p>Enviar un mensaje de Texto con los datos de registro (Usuario y Clave).</p> |

Fuente: Elaboración propia.

Tabla 15
Diseño de Pantalla - Interface de Ingreso al Aplicativo

| Código | Interfaz | Criterios de Aceptación |
|---|---|--|
| HU-03: El Usuario Ingresa correctamente al Aplicativo |  | <p>Validar que los campos estén llenos.</p> <p>Si los datos de entrada son correctos deberá ingresar al menú de opciones del aplicativo.</p> |

Fuente: Elaboración propia.

Tabla 16
 Diseño de Pantalla - Interface para Visualizar Opciones para el Usuario.

| Código | Interfaz | Criterios de Aceptación |
|--|---|---|
| HU-04: El Usuario Visualiza el Menú de Opciones del aplicativo |  | El usuario deberá visualizar las opciones de “Registrar Incidencia” y “Ver información de unidad de transporte” |

Fuente: Elaboración propia.

Tabla 17
 Diseño de Pantalla - Interface para Registrar Incidencia

| Código | Interfaz | Criterios de Aceptación |
|---|---|--|
| HU-05: El Usuario Registra Incidencia Ocurrida Durante el Viaje |  | <p>La fecha y hora se muestran de manera automática, pero pueden ser cambiadas por el usuario.</p> <p>Validar que al menos una de las 4 opciones esté seleccionada.</p> <p>Validar que el campo Comentario no exceda los 200 caracteres.</p> <p>Grabar la información en la base de datos.</p> |

Fuente: Elaboración propia.

Tabla 18
 Diseño de Pantalla - Interface para Visualizar Información de Empresas.

| Código | Interfaz | Criterios de Aceptación |
|---|---|---|
| HU-06: El Usuario Visualiza Información de Unidad de Transporte |  | Validar y buscar el número de Placa. Deberá mostrar la información de la base de datos con los símbolos de "Check" (si el valor es 0) o "X" (si el valor es mayor que 0) para cada tipo de incidencia. Las imágenes deben ser de buena calidad. |

Fuente: Elaboración propia.

Tabla 19
 Diseño de Pantalla - Interface para que los Fiscalizadores Visualicen Información

| Código | Interfaz | Criterios de Aceptación |
|--|---|--|
| HU-07: Institución Reguladora Visualiza Información de Incidencias |  | La fecha "Hasta" debe ser igual o mayor a la fecha "Desde" |

Fuente: Elaboración propia.

Tabla 20
 Diseño de Pantalla - Interface para Registrar Usuario Fiscalizador

| Código | Interfaz | Criterios de Aceptación |
|---|---|--|
| <p>HU-08: El jefe del proyecto hace el Registro de Usuario Fiscalizador</p> |  | <p>Validar que todos los campos estén llenos.</p> <p>Validar que el campo DNI sea sólo numérico.</p> <p>Validar que el campo celular sea numérico y tenga los dígitos completos.</p> <p>Grabar la información en la base de datos.</p> |

Fuente: Elaboración propia.

Tabla 21
 Diseño de Pantalla - Interface para Registrar Empresa de Transporte

| Código | Interfaz | Criterios de Aceptación |
|---|---|--|
| <p>HU-09: El jefe del proyecto hace el Registro de la empresa de transporte</p> |  | <p>Validar que todos los campos estén llenos.</p> <p>Validar que el campo RUC sea sólo numérico.</p> <p>Grabar la información en la base de datos.</p> |

Fuente: Elaboración propia.

Tabla 22
Diseño de Pantalla - Interface para Registrar Buses

| Código | Interfaz | Criterios de Aceptación |
|---|---|--|
| <p>HU-10: El jefe del proyecto hace el Registro del Bus</p> |  | <p>Validar el campo RUC sea sólo numérico, realizar búsqueda y mostrar información del Nombre de la empresa correspondiente.</p> <p>Validar que todos los campos estén llenos.</p> <p>Validar que el campo Asientos sea sólo numérico.</p> <p>Grabar la información en la base de datos.</p> |

Fuente: Elaboración propia.

Tabla 23
Diseño de Pantalla - Interface para Registrar la descripción de problemas

| Código | Interfaz | Criterios de Aceptación |
|--|---|---|
| <p>HU-11: El jefe del proyecto hace el Registro de la descripción del problema</p> |  | <p>Validar que todos los campos estén llenos.</p> <p>Grabar la información en la base de datos.</p> |

Fuente: Elaboración propia.

4.9 Análisis tecnológico

Tabla 24
Especificación de Hardware

| Hardware | Especificación |
|---|---|
| Servidor de Almacenamiento de Base de Datos | Microsoft Azure |
| Móviles | Celulares y Tablet con Sistema Operativo Android 7.0+ |

Fuente: Elaboración Propia.

Tabla 25
Especificación de Software

| Software | Especificación |
|----------------------------|----------------------------|
| Software de desarrollo | Android Studio |
| Base de datos | Microsoft SQL Server Azure |
| Herramienta de seguimiento | Kanban Flow |
| Herramienta de prototipo | Balsamiq Mockups |

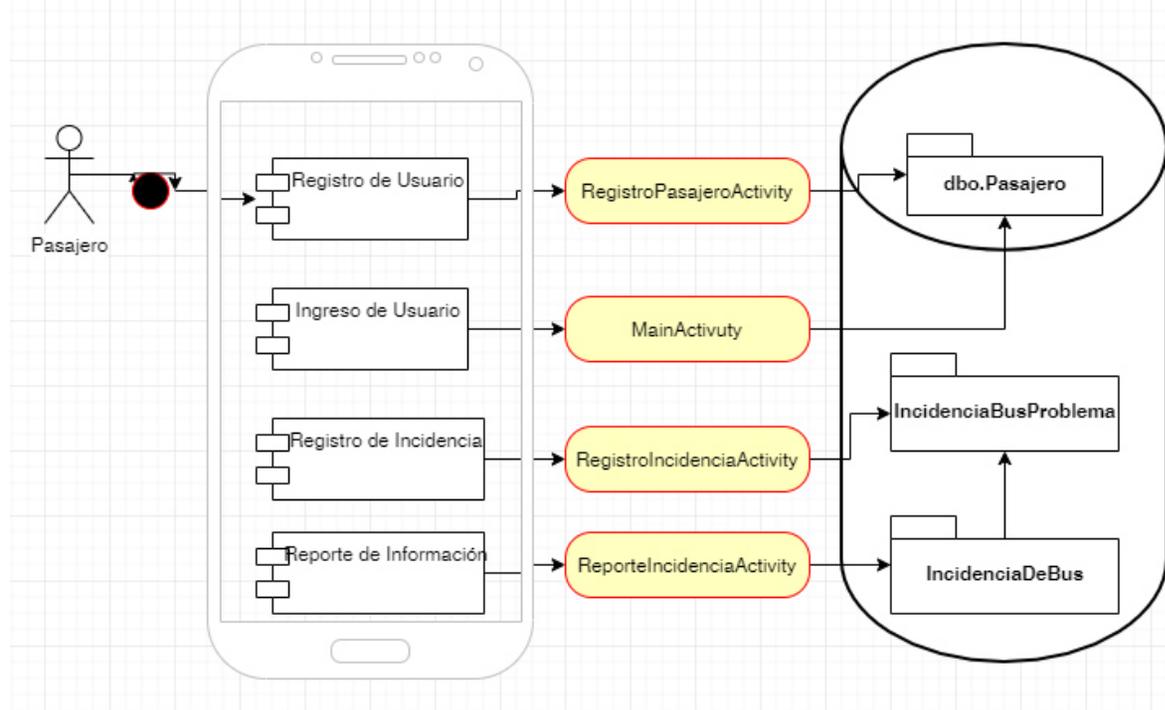
Fuente: Elaboración Propia.

4.10 Diseño de la solución

4.10.1 Arquitectura de la Solución

La solución se desarrolló en un enfoque de cliente-servidor con tres capas definidas, la capa de usuario que contiene las interfaces donde el usuario ingresa y visualiza la información, la capa de aplicación que recibe las peticiones de los usuarios, valida las reglas que se deben cumplir y se comunica con la base de datos para almacenar o solicitar los datos y la capa de datos donde almacena la información y se recupera los datos que se visualizan.

Figura 12
Componentes de la Solución



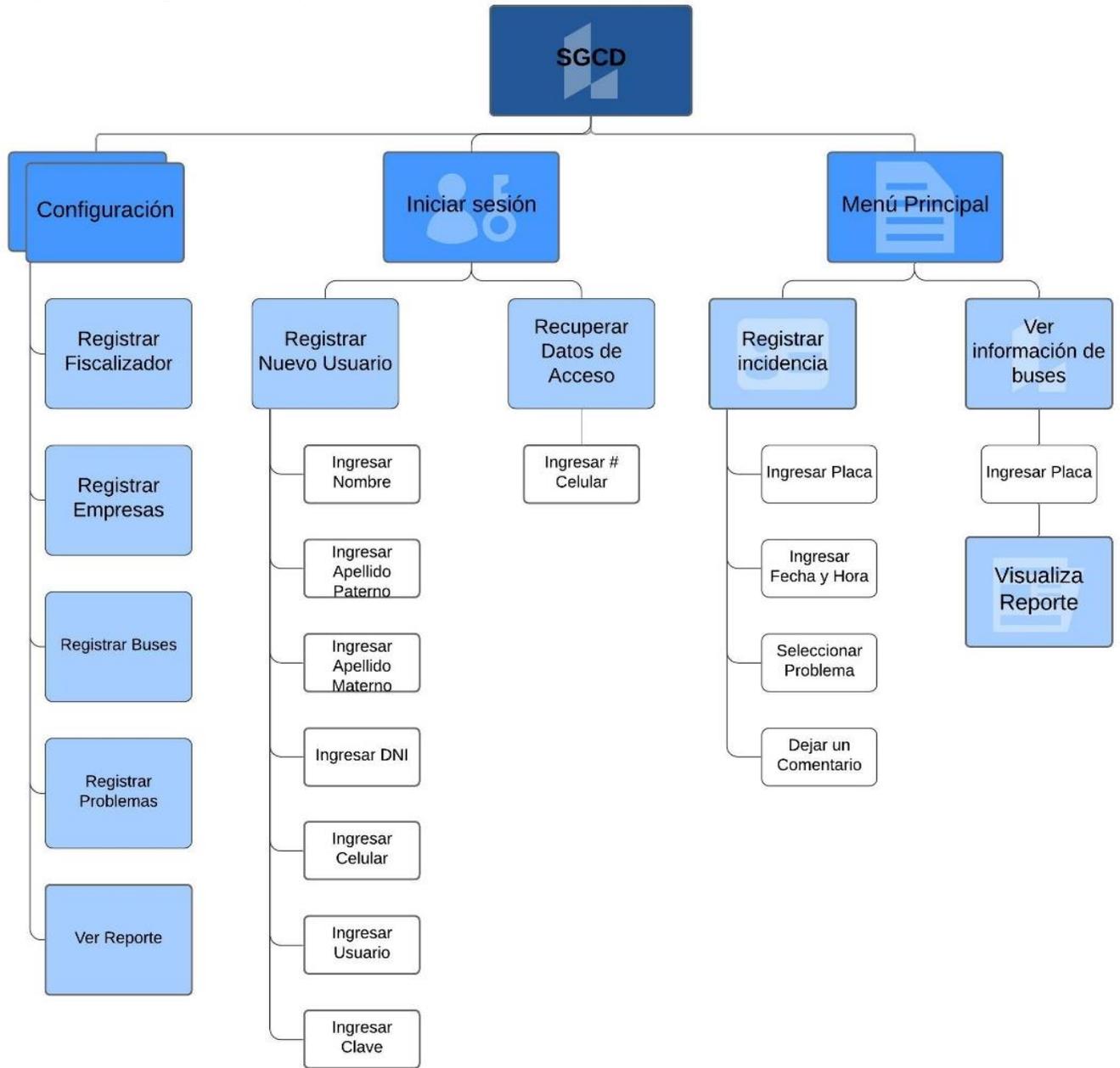
Fuente: Elaboración Propia

La interacción de los componentes anteriormente ilustrados se evidencia en el momento en que se logran almacenar los datos correctamente en las diferentes tablas de la base de datos y al recuperarlos de forma satisfactoria en las visualizaciones. Las interfaces de usuario envían los datos introducidos por el usuario a la base de datos por medio de un canal de comunicación entre la capa de aplicativo y la base de datos realizado usando "jdbc.driver".

4.11 Mapa de navegación del aplicativo móvil

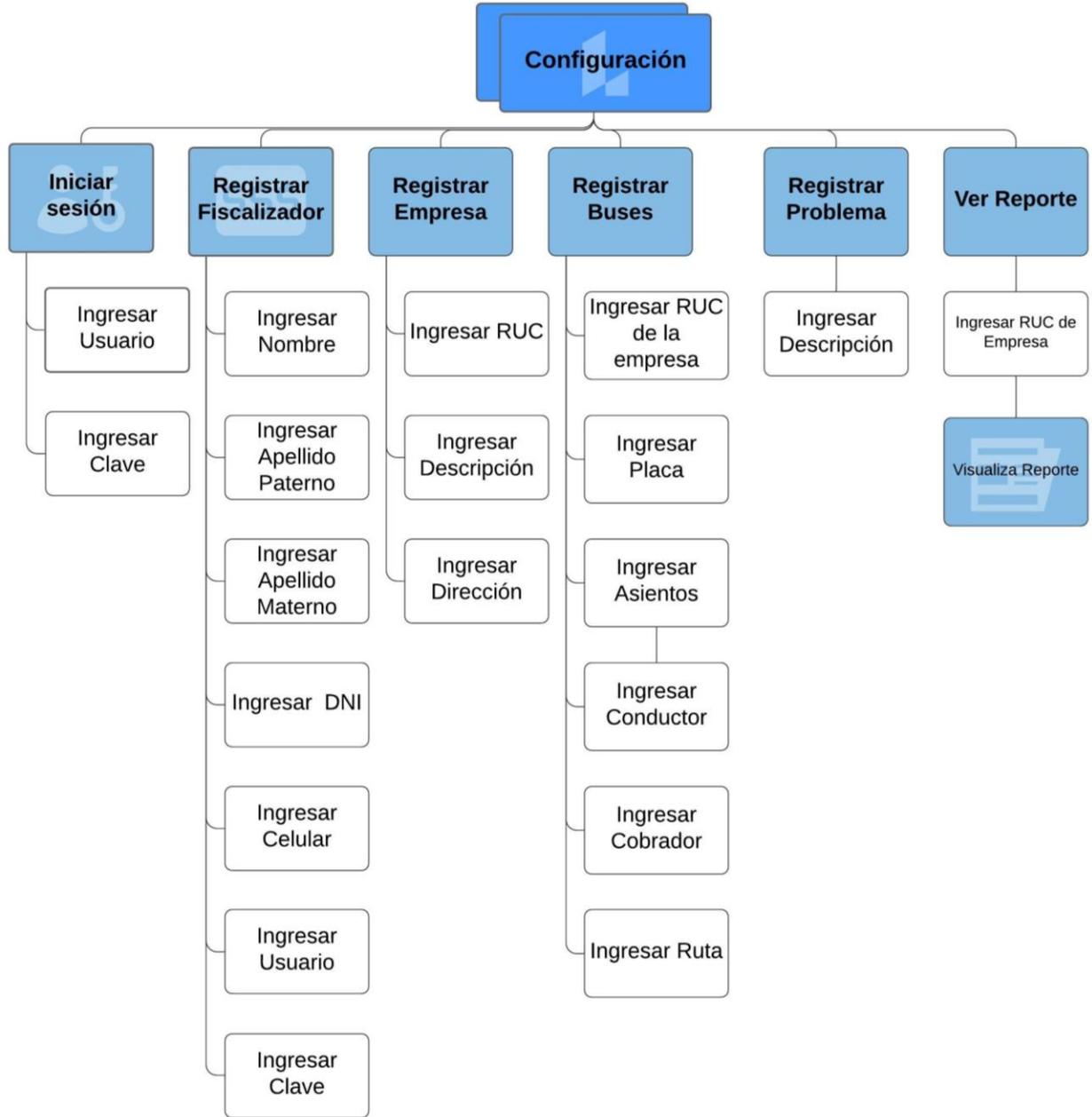
Se entiende por mapa de navegación de aplicaciones a la representación gráfica de la organización de la información de una estructura web. (Figuras 13 y 14) Esta muestra todas las relaciones de jerarquía y secuencia permitiendo elaborar escenarios de comportamiento de los usuarios.

Figura 13
 Mapa de Navegación del Aplicativo Móvil



Fuente: Elaboración Propia.

Figura 14
 Mapa de Navegación del Aplicativo Móvil - Sección Configuración



Fuente: Elaboración Propia.

4.12 Plataforma de base de datos

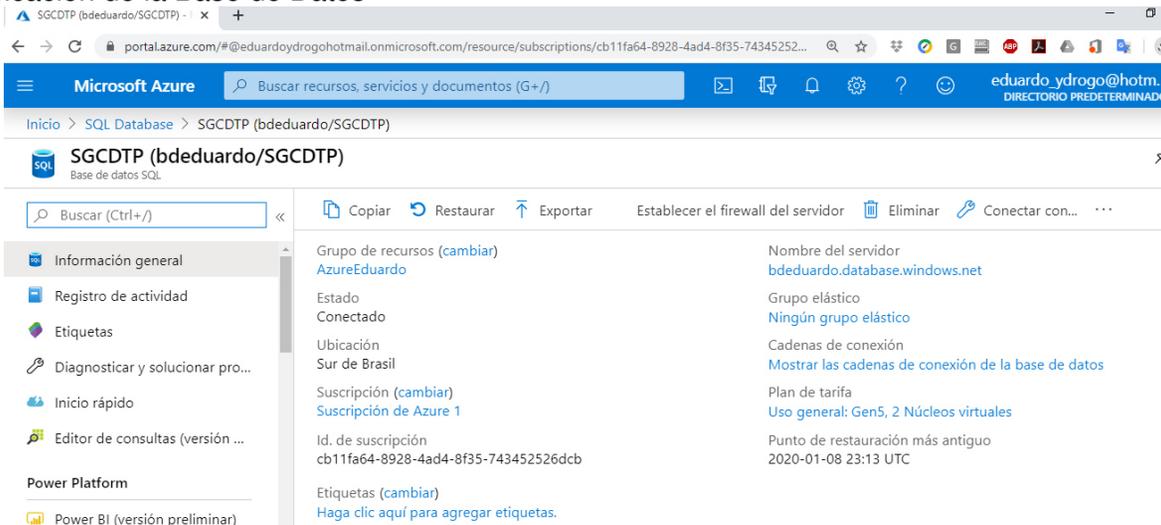
El aplicativo posee una arquitectura cliente-servidor donde el cliente realiza las peticiones a un servidor de base de datos, estas peticiones pueden provenir de diversos dispositivos móviles inteligentes tales como tabletas o celulares con sistema operativo Android.

La capa de datos está basada en la programación de procedimientos almacenados en el marco de Transact-SQL la misma que usa el estándar ANSI del lenguaje SQL.

La estructura de seguridad de la plataforma en la cual se encuentra la base de datos permite configurar los accesos a través de firewall virtuales, de esa manera es posible tener un nivel de seguridad apropiado. La misma estructura permite crear redes virtuales o privadas. Adicionalmente, la plataforma permite la copia de seguridad, restauración y exportación de la información de la base de datos.

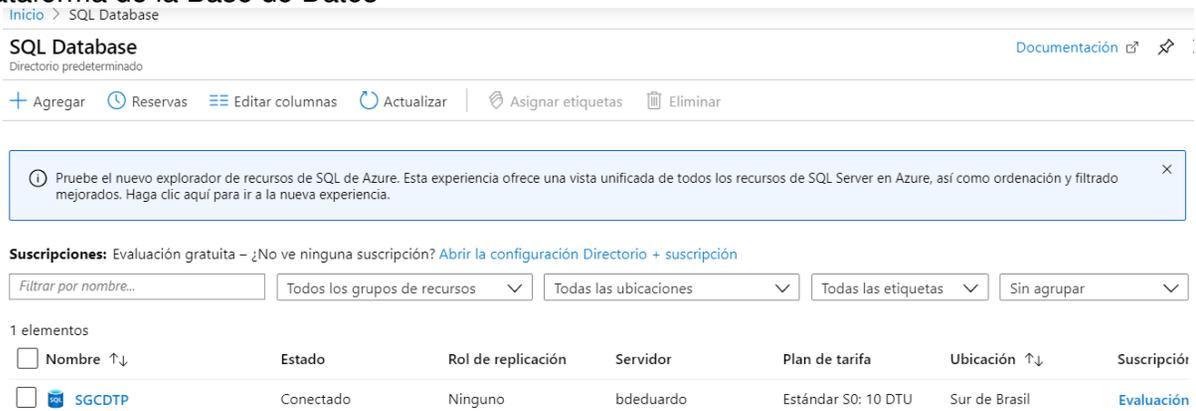
Para el almacenamiento de la información a registrar se empleó la plataforma Microsoft Azure esto permite crear, modificar y diseñar la base de datos en Microsoft SQL Server Azure.

Figura 15
Ubicación de la Base de Datos



Fuente: Impresión de pantalla plataforma Microsoft Azure.

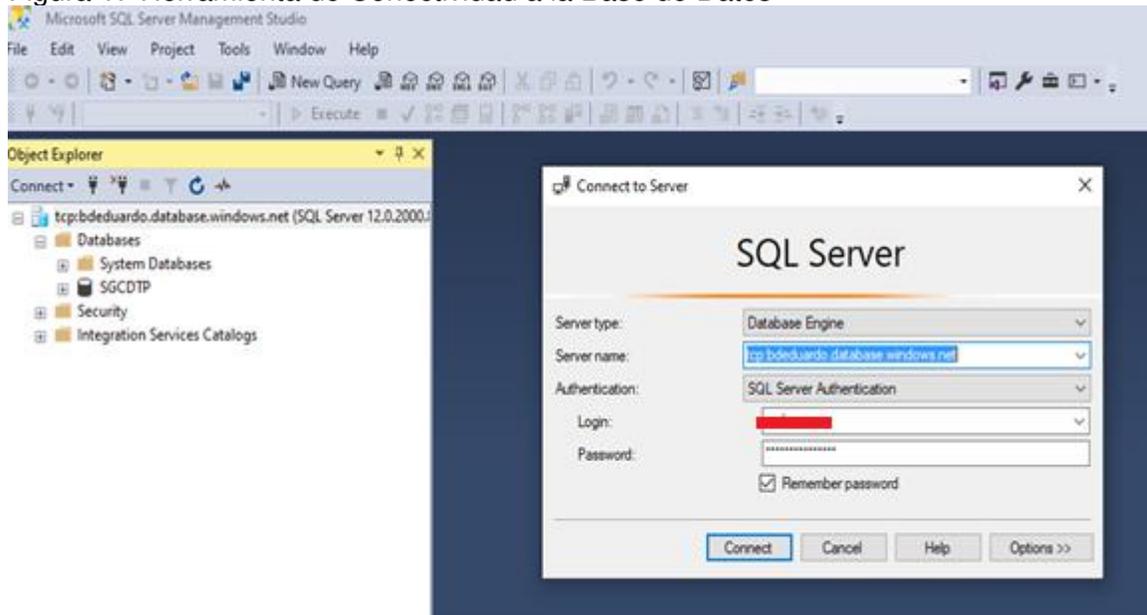
Figura 16
Plataforma de la Base de Datos



Fuente: Impresión de pantalla plataforma Microsoft Azure.

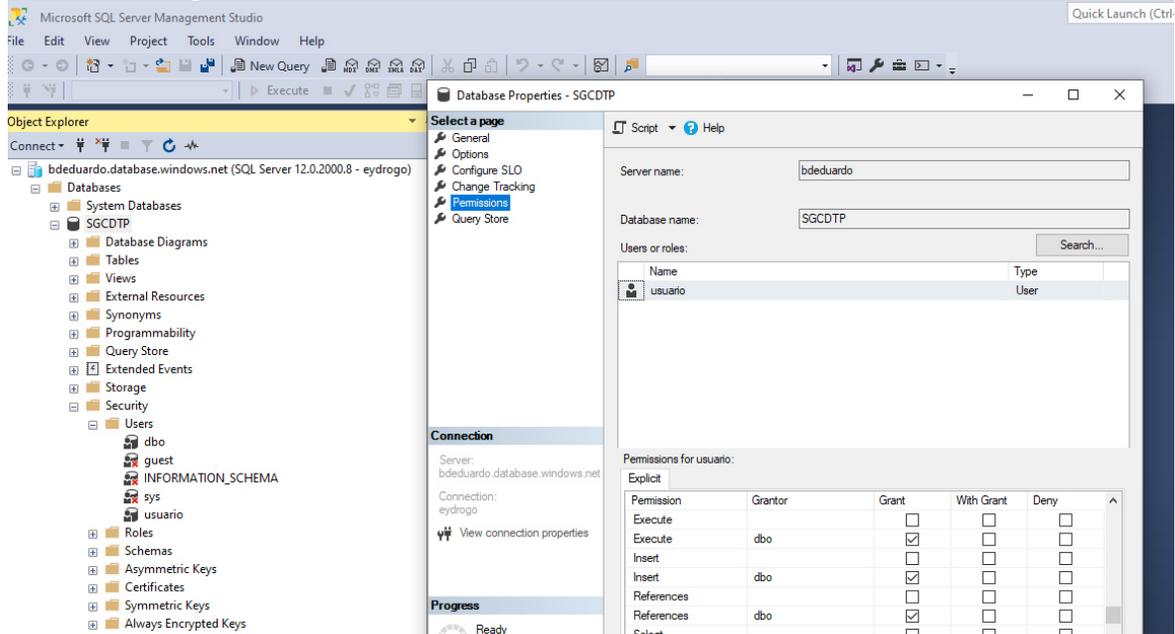
La conectividad hasta la plataforma de base de datos se realiza usando la herramienta Microsoft SQL Server Management Studio.

Figura 17 Herramienta de Conectividad a la Base de Datos



Fuente: Impresión de pantalla plataforma SQL Server.

Figura 18
Estructura de Seguridad de Acceso a la Base de Datos



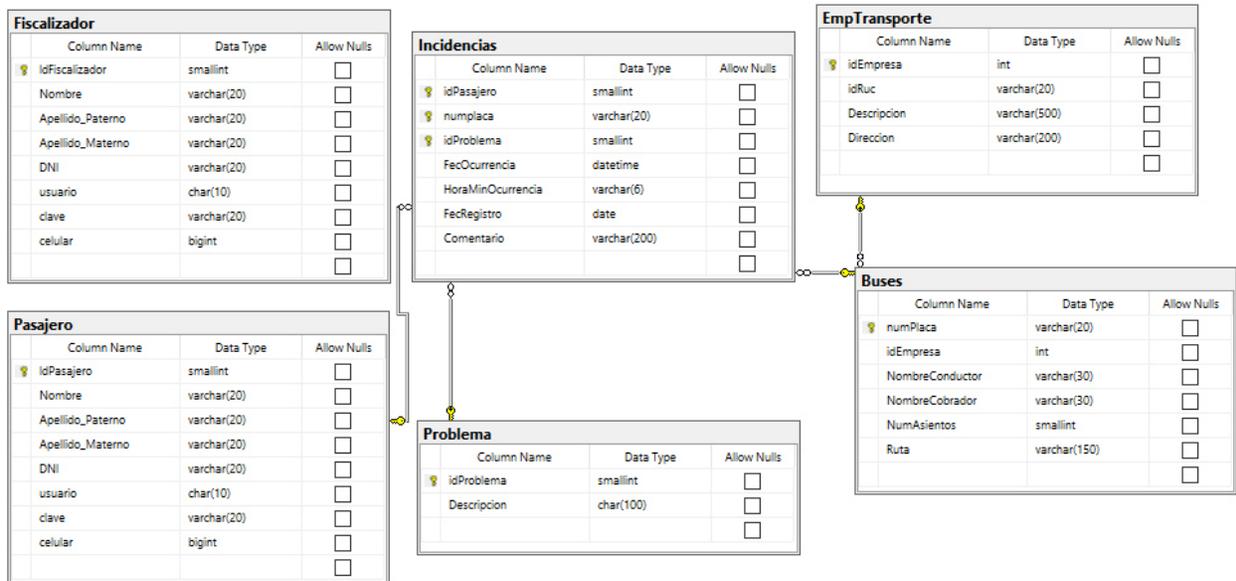
Fuente: Impresión de pantalla plataforma SQL Server.

4.13 Diseño de la de base de datos

Se realizó un modelo relacional de la base de datos del aplicativo móvil realizando las relaciones a nivel de campo garantizando de esa manera la consistencia de la información que se almacena.

Cabe mencionar que el campo FecRegistro de la tabla o entidad Incidencias almacena la fecha y hora en que se realizó el registro de la incidencia, es llenado por el mismo aplicativo a través del uso del comando getdate() y no debe confundirse con el campo FecOcurrencia el cual es llenado por el usuario y corresponde al momento en que sucedió la Incidencia.

Figura 19
Modelo Relacional de la Base de Datos



Fuente: Elaboración Propia.

4.14 Diseño de las tablas de almacenamiento de base de datos

El diseño de las tablas en las bases de datos fue sumamente importante para el desarrollo de la aplicación y para gestión de los datos obteniendo funcionalidad, fiabilidad y buen rendimiento. La siguiente tabla describe las entidades tablas y los procedimientos almacenados asociados.

Tabla 26
Diseño de Tablas de Base de Datos.

| Entidad / Tabla | Descripción | Procedimientos Almacenados |
|-----------------|--|--------------------------------------|
| EmpTransporte | Esta tabla almacena la información que se registra sobre las empresas que brindan el servicio de transporte urbano de pasajeros. | SP_SGCDTP_EmpTransporte (ANEXO Q) |
| Buses | Se almacena información de los | SP_SGCDTP_Buses (ANEXO R) |

| Entidad / Tabla | Descripción | Procedimientos Almacenados |
|-----------------|---|------------------------------------|
| | buses que tienen asignada las empresas de transporte de pasajeros. | |
| Incidencias | En esta entidad se almacena la información relacionada con las incidencias que registran los pasajeros o usuarios del servicio. El campo FecRegistro almacena la fecha y hora en que se realizó el registro y es llenado por el aplicativo a través del uso del comando getdate() y el campo FecOcurrencia es llenado por el usuario y corresponde al momento en que sucedió la Incidencia. | SP_SGCDTP_Incidencias (ANEXO P) |
| Pasajero | Se almacena la información básica relacionada al pasajero o usuarios del servicio. | SP_SGCDTP_Pasajero (ANEXO M) |
| Problema | Esta entidad almacena la información de los problemas que el usuario podría reportar | SP_SGCDTP_Problemas (ANEXO S) |

| Entidad / Tabla | Descripción | Procedimientos Almacenados |
|------------------------|---|-----------------------------------|
| | sobre el servicio recibido en el transporte público de pasajeros. | |

Fuente: Elaboración Propia.

4.15 Diccionario de datos

Tabla 27
Diccionario de Datos

| Tabla | Columna | Tipo de dato | Longitud | Permite Nulo | Auto numérico | Clave Foránea | Nombre Tabla Referenciada | Nombre Columna Referenciada |
|---------------|-----------------|--------------|----------|--------------|---------------|---------------------------------------|---------------------------|-----------------------------|
| Buses | numPlaca | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Buses | idEmpresa | int | 4 | NO | NO | FK_EmpTransporteBus _EmpTransporte | EmpTransporte | idEmpresa |
| Buses | NombreConductor | varchar | 30 | NO | NO | NULL | NULL | NULL |
| Buses | NombreCobrador | varchar | 30 | NO | NO | NULL | NULL | NULL |
| Buses | NumAsientos | smallint | 2 | NO | NO | NULL | NULL | NULL |
| Buses | Ruta | varchar | 150 | NO | NO | NULL | NULL | NULL |
| EmpTransporte | idEmpresa | int | 4 | NO | NO | NULL | NULL | NULL |
| EmpTransporte | idRuc | varchar | 20 | NO | NO | NULL | NULL | NULL |
| EmpTransporte | Descripcion | varchar | 500 | NO | NO | NULL | NULL | NULL |

| Tabla | Columna | Tipo de dato | Longitud | Permite Nulo | Auto numérico | Clave Foránea | Nombre Tabla Referenciada | Nombre Columna Referenciada |
|---------------|------------------|---------------------|-----------------|---------------------|----------------------|-----------------------------------|----------------------------------|------------------------------------|
| EmpTransporte | Direccion | varchar | 200 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | IdFiscalizador | smallint | 2 | NO | SI | NULL | NULL | NULL |
| Fiscalizador | Nombre | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | Apellido_Paterno | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | Apellido_Materno | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | DNI | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | usuario | char | 10 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | clave | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Fiscalizador | celular | bigint | 8 | NO | NO | NULL | NULL | NULL |
| Incidencias | idPasajero | smallint | 2 | NO | NO | FK_Incidencias_Pasajero | Pasajero | IdPasajero |
| Incidencias | numplaca | varchar | 20 | NO | NO | FK_Incidencias_Buses | Buses | numPlaca |
| Incidencias | idProblema | smallint | 2 | NO | NO | FK_IncidenciaBusProblema_Problema | Problema | idProblema |
| Incidencias | FecOcurrencia | datetime | 8 | NO | NO | NULL | NULL | NULL |

| Tabla | Columna | Tipo de dato | Longitud | Permite Nulo | Auto numérico | Clave Foránea | Nombre Tabla Referenciada | Nombre Columna Referenciada |
|--------------|-------------------|---------------------|-----------------|---------------------|----------------------|----------------------|----------------------------------|------------------------------------|
| Incidencias | HoraMinOcurrencia | varchar | 6 | NO | NO | NULL | NULL | NULL |
| Incidencias | FecRegistro | date | 3 | NO | NO | NULL | NULL | NULL |
| Incidencias | Comentario | varchar | 200 | NO | NO | NULL | NULL | NULL |
| Pasajero | IdPasajero | smallint | 2 | NO | SI | NULL | NULL | NULL |
| Pasajero | Nombre | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Pasajero | Apellido_Paterno | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Pasajero | Apellido_Materno | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Pasajero | DNI | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Pasajero | usuario | char | 10 | NO | NO | NULL | NULL | NULL |
| Pasajero | clave | varchar | 20 | NO | NO | NULL | NULL | NULL |
| Pasajero | celular | bigint | 8 | NO | NO | NULL | NULL | NULL |

Fuente: Elaboración propia.

4.16 Desarrollo de los procedimientos almacenados de la base de datos

Para la Interacción entre la base de datos y la capa de programación se diseñan los procedimientos almacenados necesarios codificándolos para que cumplan con la garantía de consistencia y coherencia de la información.

En los procedimientos almacenados se implementan los niveles de seguridad transaccional a fin de que la información se grabe completamente o se retraiga la transacción garantizando la no perdida o alteración de la información. Además, se inserta código de tratamiento de errores con el objetivo de obtener el mensaje del error que se pudiera producir.

Figura 20
Procedimiento Almacenado de Incidencias con Inicio de Transacción

```
ALTER Procedure [dbo].[SP_SGCDTP_IncidenciaDeBus]
    @Type char(1) = 'I'
    ,@idIncidencia int = 0
    ,@idPasajero smallint = 0
    ,@idEmpresa int = 0
    ,@idProblema1 smallint = 0
    ,@idProblema2 smallint = 0
    ,@idProblema3 smallint = 0
    ,@idProblema4 smallint = 0
    ,@numPlaca varchar(20) = 0
    ,@FecOcurrencia varchar(10)
    ,@HoraOcurrencia char(10) = ''
    ,@ObsProblema varchar(1000) = ''
as
if upper (@Type)= 'I'
Begin
    DECLARE @idIncidencia2 smallint
    DECLARE @idBus smallint
    Begin Try
        Begin Transaction
        --select @numPlaca
        set @idBus = (select idBus from [dbo].[EmpTransporteBus] where numPlaca = @numPlaca)
        set @idEmpresa = (select idEmpresa from [dbo].[EmpTransporteBus] where numPlaca = @numPlaca)
```

Fuente: Elaboración Propia – Captura de pantalla.

Figura 21
 Procedimiento Almacenado de Incidencias con Finalización de Transacción

```

VALUES
    (@idIncidencia2
    ,@IdPasajero
    ,@idEmpresa
    ,@idProblema1
    ,@idProblema2
    ,@idProblema3
    ,@idProblema4
    ,@IdBus
    ,@FecOcurrencia
    ,@HoraOcurrencia
    ,@ObsProblema
    ,getdate())
Commit Transaction
end try
Begin Catch
ROLLBACK TRANSACTION
declare @ErrorMessage nvarchar(4000);
declare @ErrorSeverity INT;
declare @ErrorState INT;
select
    @ErrorMessage = ERROR_MESSAGE(),
    @ErrorSeverity = ERROR_SEVERITY(),
    @ErrorState = ERROR_STATE();
RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)

```

Fuente: Elaboración Propia – Captura de pantalla.

Tabla 28: Listado de Procedimientos Almacenados

| Procedimiento | Tabla | Descripción |
|--|---------------------|---|
| SP_SGCDTP_Buses | Buses | Realiza los ingresos y actualizaciones de datos a la tabla de buses. |
| SP_SGCDTP_EmpTransporte | EmpTransporte | Realiza los ingresos y actualizaciones de datos a la tabla de empresas de transporte. |
| SP_SGCDTP_Fiscalizador | Fiscalizador | Realiza los ingresos y actualizaciones de datos a la tabla de fiscalizadores. |
| SP_SGCDTP_Incidencias | Incidencias | Registra las incidencias reportadas en la tabla incidencias. |
| SP_SGCDTP_Pasajero | Pasajero | Realiza los ingresos y actualizaciones de datos de usuarios pasajeros. |
| SP_SGCDTP_Problemas | Problema | Realiza los ingresos y actualizaciones a la tabla de problemas. |
| SP_SGCDTP_Reporte_Fiscalizar_Incidencias | Incidencias, Buses, | Procesa, totaliza y muestra información de incidencias de uso del |

| Procedimiento | Tabla | Descripción |
|------------------------------|-------------------------|--|
| | EmpTransporte, Problema | usuario fiscalizador usando las tablas de incidencias, buses y problema |
| SP_SGCDTP_Reporte_Incidencia | Incidencias, Problema | Procesa, totaliza y muestra información de incidencias ocurridas durante la prestación de servicio registrado por el pasajero. |

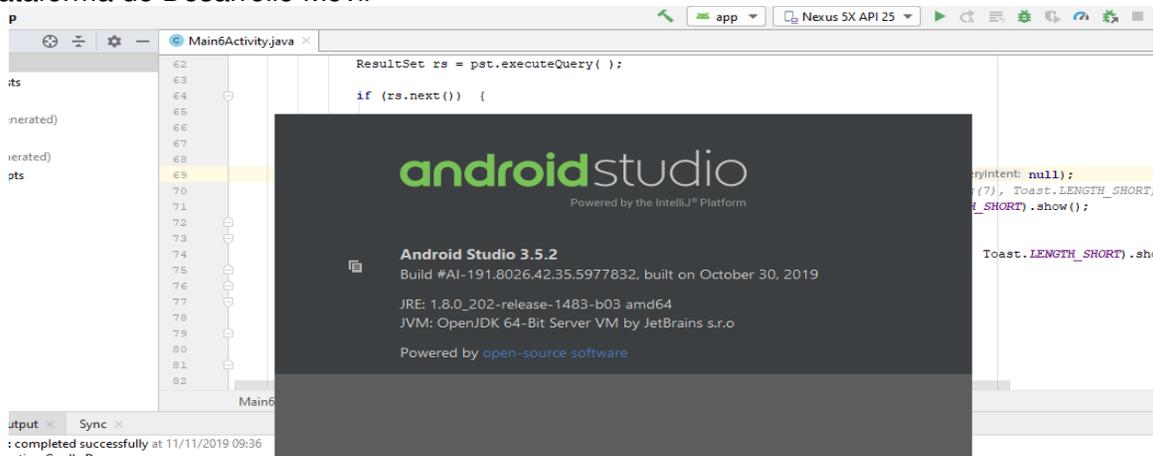
Fuente: Elaboración propia.

CAPÍTULO V: CONSTRUCCIÓN

5.1 Plataforma de desarrollo

Para el desarrollo del aplicativo se usó tecnología Android, para ello se empleó la herramienta de programación Android Studio 3.5.2.

Figura 22
Plataforma de Desarrollo Móvil



Fuente: Impresión de pantalla, plataforma Android Studio

5.1.1 Descripción del aplicativo móvil

El diseño del aplicativo móvil estuvo guiado bajo un enfoque de diseño visual responsivo, en donde el aplicativo posee la capacidad de mantener su misma lógica de navegación y reacomodar su estructura a un ancho adecuado para su visualización en dispositivos móviles.

En ese sentido, los usuarios que visualizaron el aplicativo desde diferentes dispositivos notan que ocupa el mismo tamaño de pantalla. Por lo tanto, siguiendo la lógica responsiva planteada en el modelo de navegación, hasta el tercer nivel ha de ser el mismo para todos los dispositivos desde donde se visualice el aplicativo móvil.

En la construcción del aplicativo se incorporó elementos de validaciones de datos previos a la solicitud de interacción con la base de datos para su almacenamiento. Estas validaciones

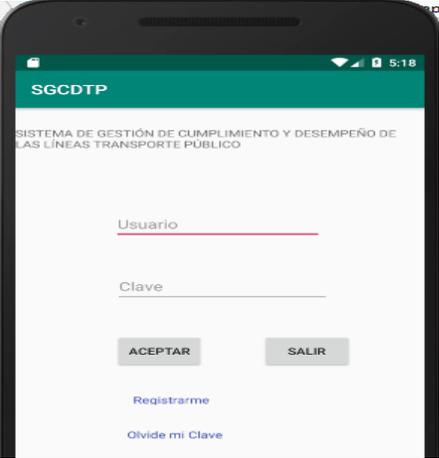
permiten realizar la verificación que los datos a enviar tengan valor, sean correctos y consistencia.

Para el registro de las incidencias y con el objetivo que al usuario le resulte sencillo, amigable y rápido, se implementó tres datos de ingresos y datos de selección rápida para determinar la calidad del servicio recibido en el transporte público de pasajero. Opcionalmente el usuario puede registrar información adicional que le permita complementar su incidencia.

5.1.2 Pantallas del aplicativo móvil

En esta sección se describen las diferentes interfaces de usuario y se detalla el funcionamiento de cada una.

Tabla 29
Pantallas del Aplicativo Móvil

| Descripción | Construcción |
|---|---|
| <p>Pantalla A:</p> <p>En esta pantalla se solicita al usuario ingresar un usuario y clave para el ingreso al aplicativo.</p> <p>El pasajero ingresa su usuario y clave si existe en la base de datos el sistema lo mostrará la pantalla de menú principal Pantalla D.</p> |  |

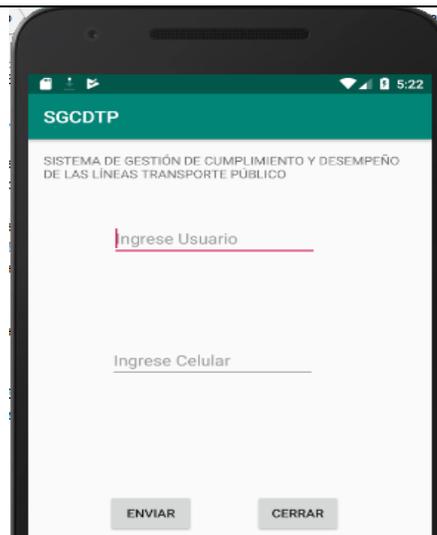
Pantalla B:

El usuario del servicio puede registrarse. Si el usuario quiere hacer uso del aplicativo debe ingresar los datos de identificación y acceso básicos los cuales quedarán registrados en la base de datos.



Pantalla C:

Pantalla para recuperar la clave en caso de olvido. En caso el usuario no recuerda su clave en esta pantalla puede solicitar que el sistema se la envíe por mensaje de texto. Para ello, debe ingresar su usuario y su número de celular con que se registró.



Pantalla D:

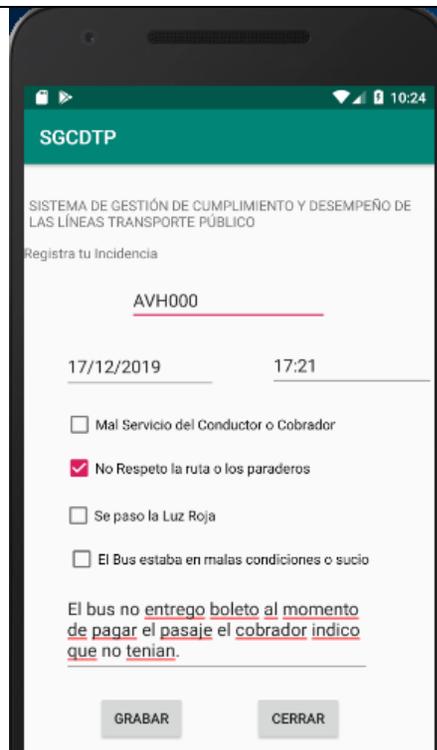
Menú de opciones principal del aplicativo móvil para que el usuario registre alguna incidencia o revise un dashboard de información de alguna línea de transporte.



Pantalla E:

Pantalla de registro de incidencias en la cual se solicita al usuario del servicio ingrese información referente al servicio recibido.

En esta pantalla el usuario debe ingresar como datos básicos la placa del bus, fecha y hora de ocurrencia del hecho y el sistema le muestra las opciones predeterminadas y una opción de descripción de tres líneas.



Pantalla F:

Pantalla de información gráfica sobre las incidencias cometidas por los buses o las empresas de transporte público. Esta pantalla es una visualización gráfica de las recurrentes incidencias que puede tener una empresa de transporte o un bus en particular.



Pantalla G:

Pantalla de información gráfica sobre el total de las incidencias cometidas por empresa de transporte público. Esta pantalla es una visualización gráfica que muestra un resumen tipo reporte a los usuarios fiscalizadores.



Pantalla H:

El usuario fiscalizador del servicio puede registrarse. Si quiere hacer uso del aplicativo como fiscalizador debe ingresar los datos de identificación y acceso básicos los cuales quedarán registrados en la base de datos.

The screenshot shows a mobile application interface for registration. At the top, there is a teal header with the text 'SGCDTP'. Below the header, the title of the system is displayed: 'SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y DESEMPEÑO DE LAS LÍNEAS TRANSPORTE PÚBLICO (DATOS DEL FISCALIZADOR)'. The form consists of several input fields: 'Nombre', 'Apellido Paterno', 'Apellido Materno', 'DNI', and 'Celular'. Below these fields are two more fields labeled 'usuario' and 'Clave'. At the bottom of the form, there are two buttons: 'GRABAR' and 'CERRAR'. The status bar at the top right shows the time as 4:45.

Pantalla I:

Pantalla que permite el registro de la empresa de transporte, usando el campo clave RUC.

The screenshot shows a mobile application interface for company registration. At the top, there is a teal header with the text 'SGCDTP'. Below the header, the title of the system is displayed: 'SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y DESEMPEÑO DE LAS LÍNEAS TRANSPORTE PÚBLICO - REGISTRAR EMPRESA'. The form consists of three input fields: 'Ruc', 'Descripcion', and 'Direccion'. At the bottom of the form, there are two buttons: 'GRABAR' and 'CERRAR'. The status bar at the top right shows the time as 4:46.

Pantalla J:

Pantalla de registro de Buses, realiza una búsqueda con el dato de entrada RUC de la empresa de transporte a la cual pertenece el Bus. Se ingresan los datos básicos que identifican al Bus.



Pantalla K:

Pantalla para registrar la descripción de los diferentes problemas que pueden ocasionarse durante la prestación del servicio. Estos problemas son los que se encuentran listados en la interface de Registro de Incidencias.



Fuente: Elaboración propia.

Tabla 30
Resumen de Pruebas por Historia de Usuario

| Historia Usuario | Estado de prototipado | de Estado de prueba | de Número de Pruebas realizadas (Iteraciones) |
|-------------------------|------------------------------|----------------------------|--|
| HU-01 | Construido | Probado | 10 |
| HU-02 | Construido | Probado | 10 |
| HU-03 | Construido | Probado | 10 |
| HU-04 | Construido | Probado | 10 |
| HU-05 | Construido | Probado | 10 |
| HU-06 | Construido | Probado | 10 |
| HU-07 | Construido | Probado | 10 |
| HU-08 | Construido | Probado | 10 |
| HU-09 | Construido | Probado | 10 |
| HU-10 | Construido | Probado | 10 |
| HU-11 | Construido | Probado | 10 |

Fuente: Elaboración propia.

5.2 Componentes usados para el desarrollo

El lenguaje de la programación empleado para el desarrollo de este aplicativo móvil es Java a través de la plataforma de Android Studio.

El entorno de desarrollo utilizado está basado en Intelli JDEA y se encuentra integrado a la interface de desarrollo de Android Studio permitiendo una compilación basada en Gradle la cual es una herramienta que permite la automatización de compilación de código abierto y se centra en la flexibilidad y el rendimiento.

El canal de comunicación o la capa de aplicación que recibe las peticiones de los usuarios, valida las reglas que deben cumplirse y se comunica con la base de datos fue creado con jdbc.driver.

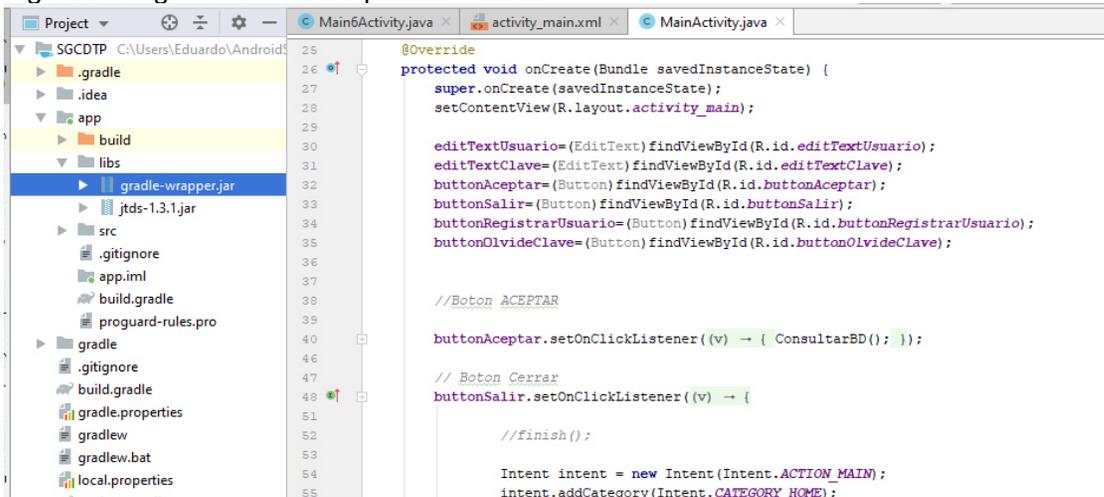
Para identificar los problemas de rendimiento que se pudieran presentar, errores de programación o código sospechoso, se usó la herramienta Lint. Cuando Lint encuentra algún error, o bien lo repara de forma automática o bien nos advierte de ello a través de distintos mensajes, basándose para esto en un conjunto de reglas definidas previamente y que son altamente configurables.

ICU4J API es el framework que expone Android para el desarrollo de aplicaciones a través del paquete android.icu. ICU4J es un conjunto de bibliotecas de Java de código abierto que nos brindó la compatibilidad con Unicode y globalización para el aplicativo. A partir de Android 7.0 (nivel de API 24), Android expone un subconjunto de ICU4J API para que los desarrolladores de apps las usen en el paquete android.icu.

5.3 Desarrollo del aplicativo móvil

El aplicativo se diseñó usando el lenguaje de programación Java para Android. Para las validaciones en la etapa de desarrollo se usó el emulador Nexus 5x API 25 de la plataforma Android. Para los gráficos se usó la librería MPAndroidChart que es completamente compatible con las versiones actuales de los móviles y tablets.

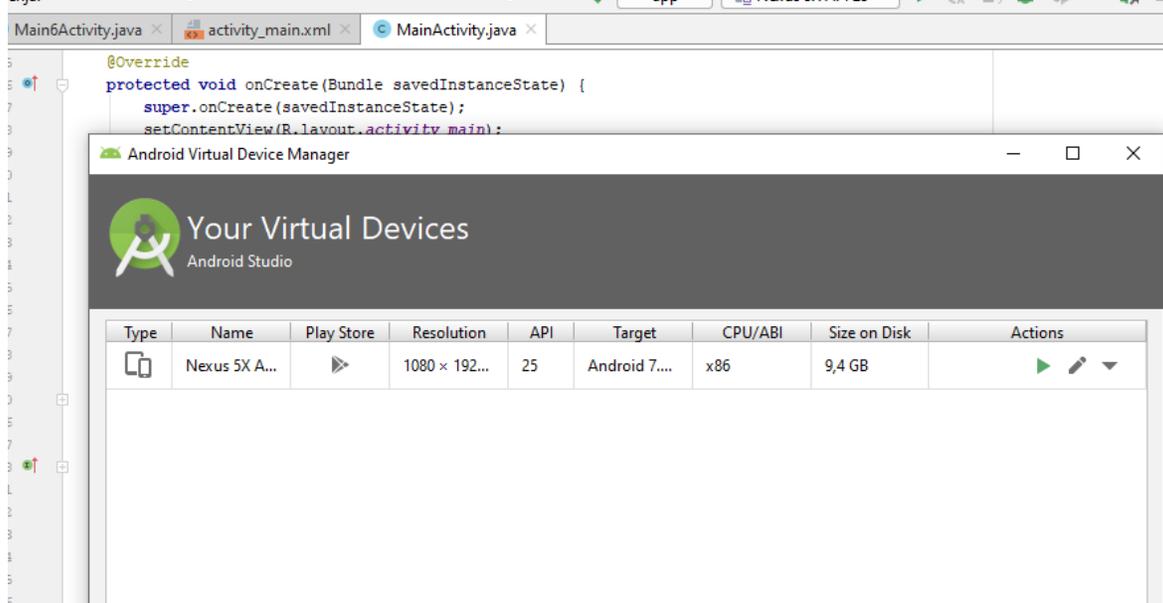
Figura 23
Código de Programación del Aplicativo Móvil



```
25 @Override
26 protected void onCreate(Bundle savedInstanceState) {
27     super.onCreate(savedInstanceState);
28     setContentView(R.layout.activity_main);
29
30     editTextUsuario=(EditText) findViewById(R.id.editTextUsuario);
31     editTextClave=(EditText) findViewById(R.id.editTextClave);
32     botonAceptar=(Button) findViewById(R.id.botonAceptar);
33     botonSalir=(Button) findViewById(R.id.botonSalir);
34     botonRegistrarUsuario=(Button) findViewById(R.id.botonRegistrarUsuario);
35     botonOlvideClave=(Button) findViewById(R.id.botonOlvideClave);
36
37     //Boton ACEPTAR
38     botonAceptar.setOnClickListener((v) -> { ConsultarBD(); });
39
40     // Boton Cerrar
41     botonSalir.setOnClickListener((v) -> {
42
43         //finish();
44
45         Intent intent = new Intent(Intent.ACTION_MAIN);
46         intent.addCategory(Intent.CATEGORY_HOME);
47     });
48
49 }
```

Fuente: Captura de pantalla.

Figura 24
Emulador de Dispositivos Móviles Usado para el Desarrollo de la Solución



Fuente: Captura de pantalla.

5.4 Visualización del funcionamiento del aplicativo

Se realizaron los videos que permiten visualizar el uso y funcionamiento del aplicativo para los usuarios:

<https://www.youtube.com/watch?v=MQmlK9X03IE>

https://www.youtube.com/watch?v=lp-a3OM_tq4&t=24s

5.5 Pruebas y resultados

Para las pruebas se consideraron los elementos de funcionalidad, usabilidad y utilidad, adicionalmente se realizaron pruebas tecnológicas a la base de datos y al funcionamiento del aplicativo a través de la carga masiva de registros. Así mismo, se realizaron pruebas a los servidores y a la plataforma tecnológica sobre la cual está desarrollado el aplicativo.

Las pruebas tienen por objetivo controlar la calidad del software verificando la correcta ejecución, visualización y accesibilidad del aplicativo desarrollado de esa manera se podrán realizar los ajustes técnicos o de diseño según corresponda.

5.5.1 Pruebas de funcionalidad, usabilidad y utilidad.

Las pruebas de funcionalidad y usabilidad consistieron en el uso del aplicativo por parte de personas que son usuarios del servicio de transporte público. Para estas pruebas se permitió el uso del aplicativo en un entorno real, controlado y se determinó la apropiada amigabilidad y usabilidad en sus interfaces, texto y forma de ingreso de la información. Luego, se consultó con los usuarios sobre la experiencia con el aplicativo y la utilidad de la información que se mostró obteniendo un alto grado de satisfacción.

Para esta prueba y en base a una población infinita que hace uso del servicio del transporte público y utilizando un nivel de confianza de 95% ($z=1.96$), la desviación estándar (s) de 93 y error máximo permitido (d) de 18, usando la fórmula estadística muestral ($n=((z*s)/d)^2$), por juicio y opinión, se otorgó acceso a los componentes desarrollados a 103 personas mayores de edad que hacen uso frecuente del servicio de transporte público para trabajar o estudiar en la ciudad de Lima en el distrito de San Borja y Surquillo para que registren la información de las incidencias ocurridas durante el uso del servicio de transporte urbano de pasajeros en el período febrero y marzo del presente año y de esa manera validen la funcionalidad, usabilidad y utilidad del aplicativo. Para obtener su experiencia y opinión se preparó una encuesta virtual. (Tabla 31)

Fórmula estadística muestral empleada:

z = nivel de confianza de 95%

s = desviación estándar de 93

d =error máximo permitido de 18

$$n = ((Z * s) / d)^2$$

Las variables empleadas en esta encuesta son de tipo cualitativo en sus dos versiones poliatómicas y dicotómicas.

Tabla 31
Encuesta aplicada a los usuarios del servicio de transporte Público

| | |
|---|--|
| 1.- Género | <input type="checkbox"/> Masculino |
| | <input type="checkbox"/> Femenino |
| 2.- Tipo de usuario | <input type="checkbox"/> Pasajero |
| | <input type="checkbox"/> Fiscalizador |
| 3.- ¿Con cuanta frecuencia utiliza Ud.? el servicio de transporte público (Buses)? | <input type="checkbox"/> A diario |
| | <input type="checkbox"/> Frecuentemente (3 a 5 veces por semana) |
| | <input type="checkbox"/> Ocasionalmente (3 a 5 veces al mes) |
| 4.- ¿Está Ud. de acuerdo con utilizar una App para registrar incidencias ocurridas durante la prestación del servicio de transporte público? | <input type="checkbox"/> Totalmente de acuerdo |
| | <input type="checkbox"/> Parcialmente de acuerdo |
| | <input type="checkbox"/> En desacuerdo |
| 5.- ¿Logró registrarse correctamente en la App? | <input type="checkbox"/> Si |
| | <input type="checkbox"/> No |
| 6.- ¿Pudo registrar las incidencias ocurridas durante su viaje en transporte público (Bus) usando la App? | <input type="checkbox"/> Si |
| | <input type="checkbox"/> No |
| 7.- ¿Consiguió visualizar la información de incidencias presentadas en una unidad de transporte (Bus)? | <input type="checkbox"/> Si |
| | <input type="checkbox"/> No |
| 8.- ¿Qué nivel de utilidad considera que tiene la información que visualizó a través del aplicativo móvil? | <input type="checkbox"/> Alto |
| | <input type="checkbox"/> Medio |
| | <input type="checkbox"/> Bajo |
| 9.- ¿Considera Ud. que la Municipalidad de Lima debe tomar como insumo lo registrado en el aplicativo móvil al momento de evaluar la renovación de licencias de las líneas de transporte? | <input type="checkbox"/> Si estoy de acuerdo |
| | <input type="checkbox"/> No lo sé |
| | <input type="checkbox"/> No estoy de acuerdo |

10.- ¿Considera Ud. que la Municipalidad de Lima debe tomar como insumo lo registrado en el aplicativo móvil al momento de evaluar la renovación de licencias de conducir los conductores de las líneas de transporte?

- Si
- No

11.- ¿Cómo calificaría Ud. el funcionamiento del aplicativo móvil para el registro de incidencias en el servicio de transporte público?

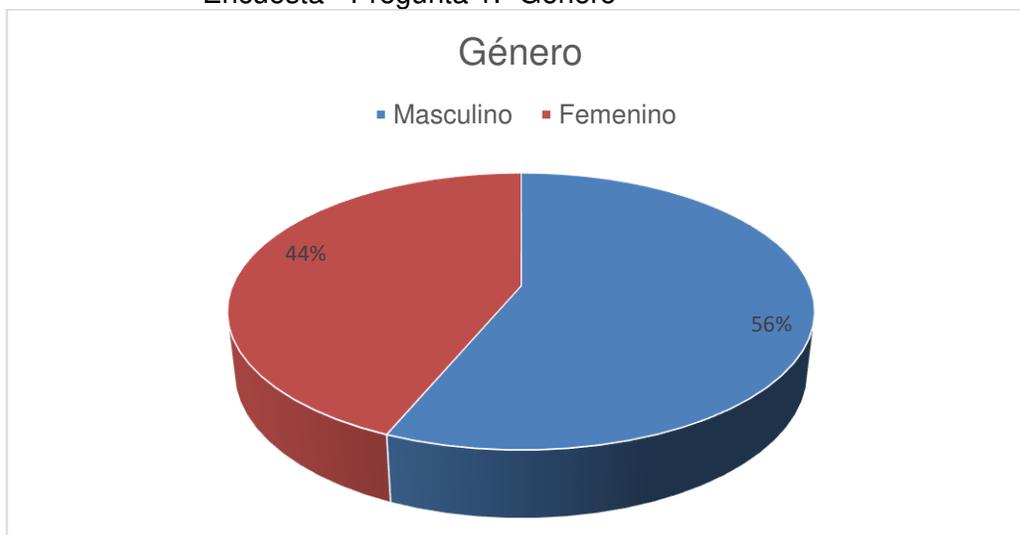
- Excelente
- Bueno
- Regular
- Deficiente

Fuente: Elaboración propia.

5.5.2 Resultado de pruebas de funcionalidad, usabilidad y utilidad.

Una vez analizadas las respuestas obtenidas por parte de los usuarios se procedió a graficar los resultados:

Figura 25
Encuesta - Pregunta 1.- Género



Fuente: Elaboración propia.

Leyenda: 58 personas que llenaron la encuesta son varones esto representa un 56,3% de la población y 45 personas el 44% son mujeres.

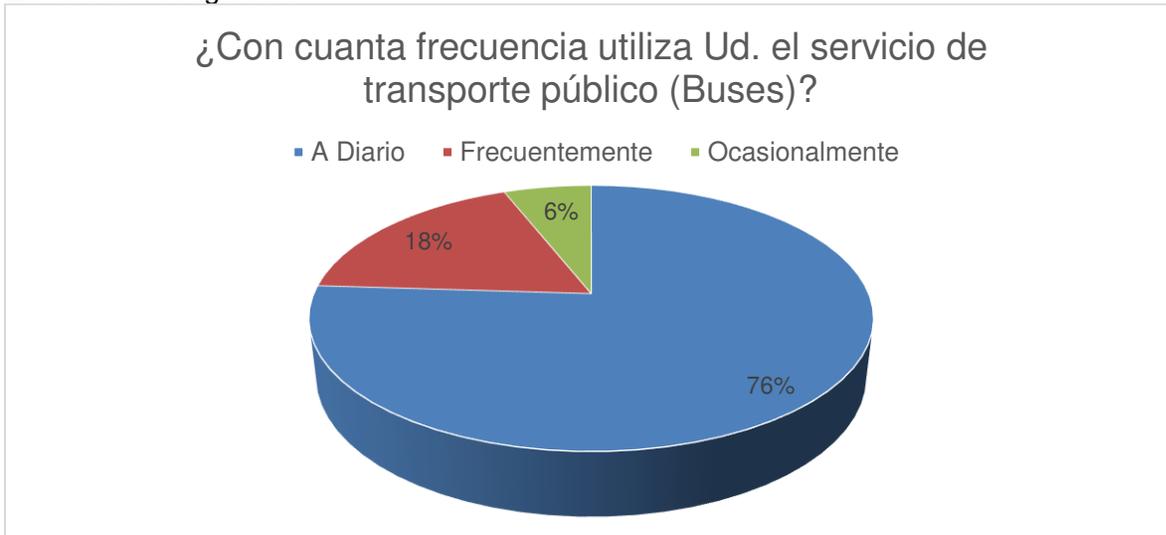
Figura 26
Encuesta - Pregunta 2: Tipo de Usuario



Fuente: Elaboración propia.

Leyenda: 96 personas que llenaron la encuesta (93%) son Pasajeros y 7 personas son Fiscalizadores (7%).

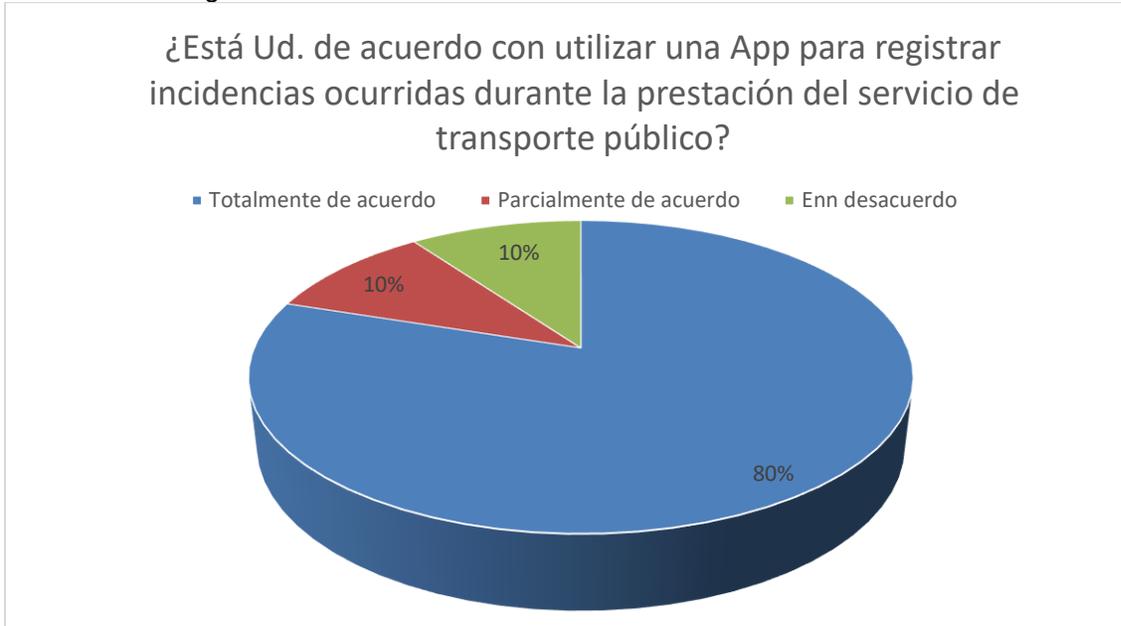
Figura 27
Encuesta – Pregunta 3



Fuente: Elaboración propia.

Leyenda: 78 personas encuestadas (76%) respondieron que usan Buses a diario, 19 personas (18%) los usan Frecuentemente y 6 personas (6%) Ocasionalmente.

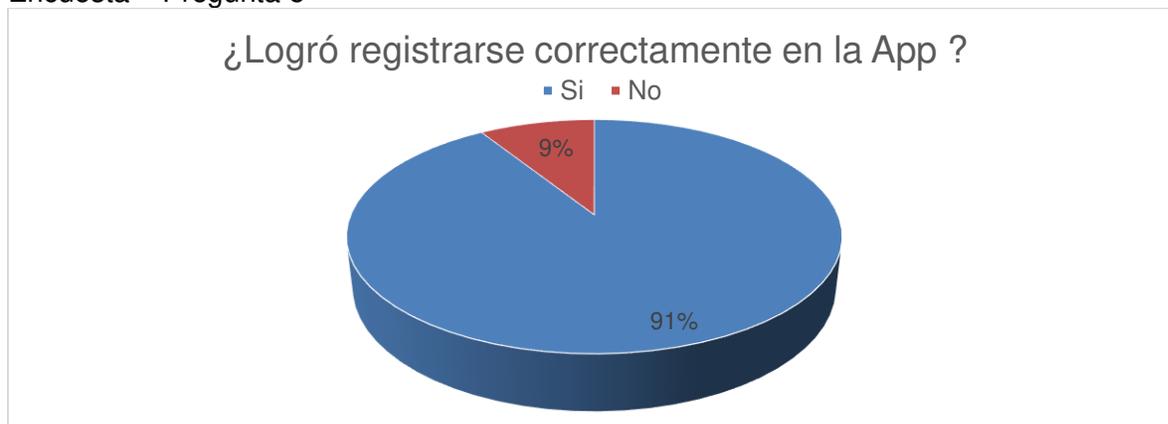
Figura 28
Encuesta – Pregunta 4



Fuente: Elaboración propia.

Leyenda: El 80% de los encuestados (82 personas) respondieron estar completamente de acuerdo con registrar las incidencias que ocurren durante la prestación del servicio de transporte público, un 10% (11 personas) estuvo Parcialmente de acuerdo y el 10% restante (10 personas) no estuvo de acuerdo.

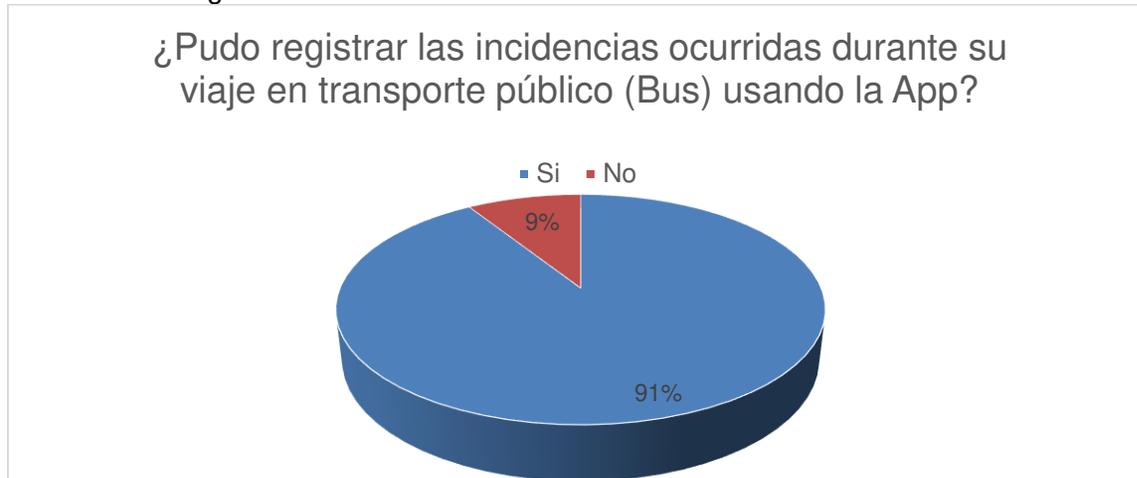
Figura 29
Encuesta – Pregunta 5



Fuente: Elaboración propia.

Leyenda: El 91% de los encuestados (94 personas) respondieron que, Si lograron registrarse correctamente en el aplicativo, mientras un 9% de los encuestados (9 personas) respondieron que No.

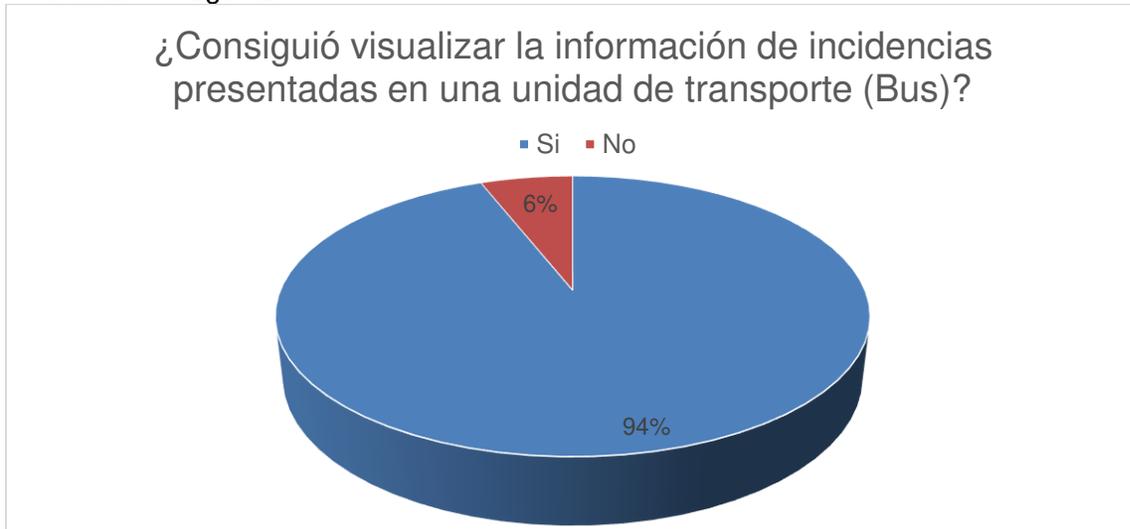
Figura 30
Encuesta – Pregunta 6



Fuente: Elaboración propia.

Leyenda: El 91% de los encuestados (94 personas) respondieron que sí pudo registrar las incidencias ocurridas durante su viaje en el transporte público de pasajeros, el 9% restante (9 personas) respondió que no.

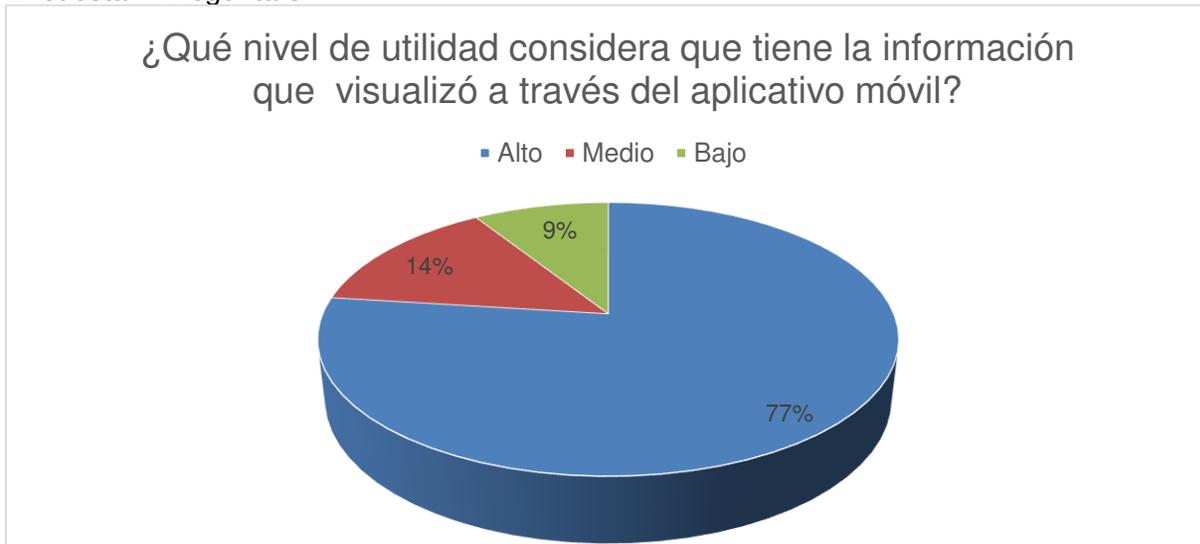
Figura 31
Encuesta – Pregunta 7



Fuente: Elaboración propia

Leyenda: El 94% de los encuestados (97 personas) respondieron que Si consiguió visualizar la información de incidencias presentadas en una unidad de transporte (Bus), el otro 6% de encuestados (6 personas) respondió que No lo consiguió.

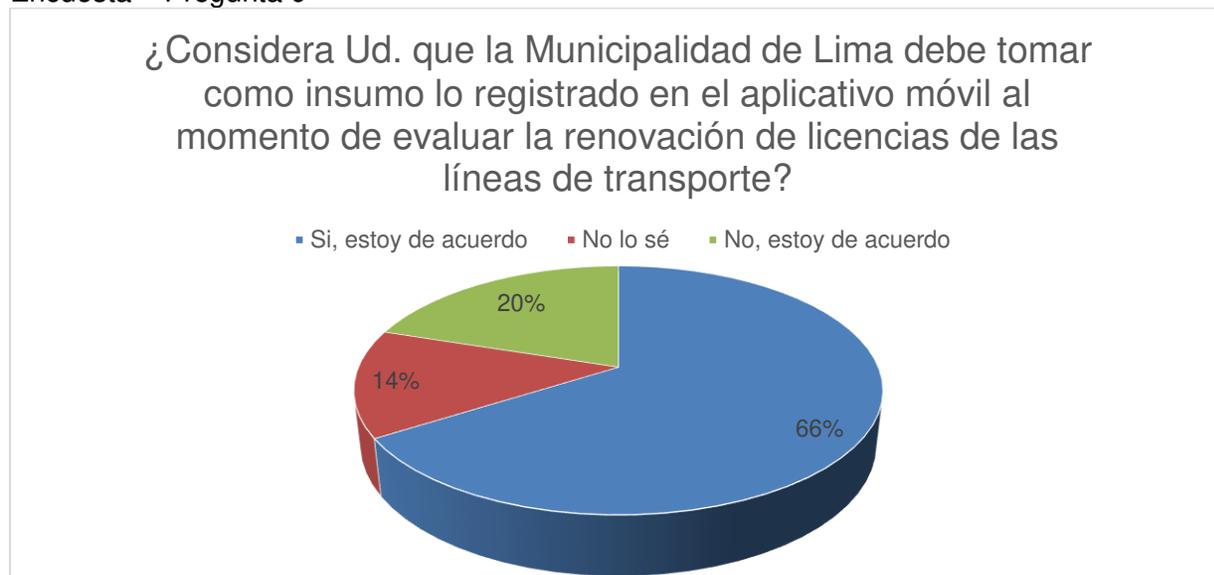
Figura 32
Encuesta – Pregunta 8



Fuente: Elaboración propia

Leyenda: El 77% de los encuestados (79 personas) respondieron que el nivel de utilidad de la información que visualizó en el aplicativo es Alto, el 14% de los encuestados (15 personas) respondieron que su nivel es Medio y el 9% restante (9 personas) respondieron que el nivel es Bajo.

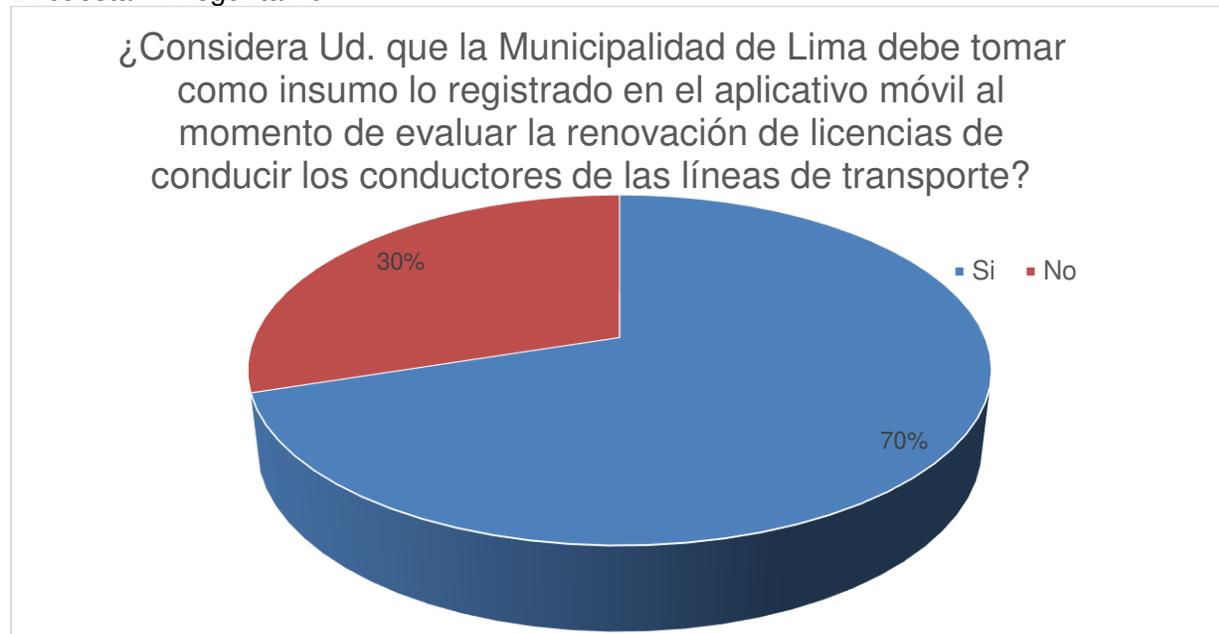
Figura 33
Encuesta – Pregunta 9



Fuente: Elaboración propia

Leyenda: 66% de los encuestados (68 personas) considera que Si, está de acuerdo en que la municipalidad de Lima debe considerar la información registrada en el aplicativo al momento de evaluar la renovación de licencia de ruta de las líneas de transporte, el 14% de ellas (15 personas) No lo sabe y el 20% restante (20 personas) que No, está de acuerdo.

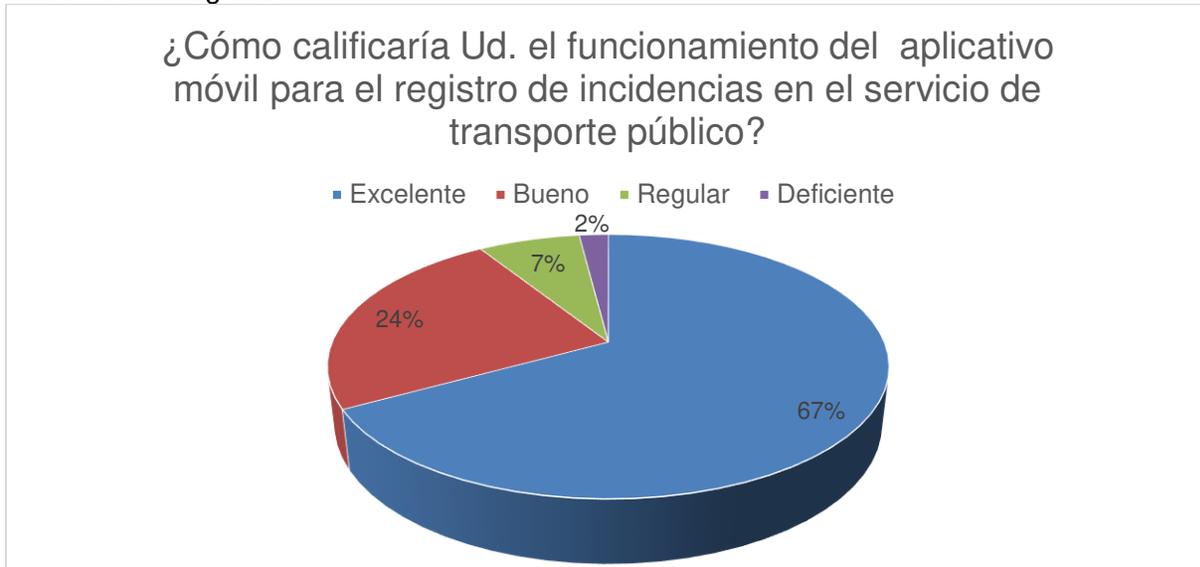
Figura 34
Encuesta – Pregunta 10



Fuente: Elaboración propia

Leyenda: el 70% de los encuestados (72 personas) considera que la municipalidad de Lima Si debe considerar la información registrada en el aplicativo al momento de evaluar la renovación de licencia de los conductores de las líneas de transporte público de pasajeros y el 30% restante (31 personas) que No.

Figura 35
Encuesta – Pregunta 11



Fuente: Elaboración propia

Leyenda: el 67% de los encuestados (65 personas) considera excelente el funcionamiento del aplicativo móvil. El 24% de los encuestados califica al aplicativo como bueno, un 7% lo califica como regular un 2% como deficiente.

5.5.3 Prueba de rendimiento del aplicativo

Esta prueba consistió en determinar la forma como el aplicativo móvil responde cuando su código fuente se ejecute en los dispositivos móviles. A través de esta prueba pudimos validar si el desarrollo final tendría una correcta funcionalidad en los tamaños de pantallas comunes para móviles y Tablet. Para ello, se configuró un simulador para SO Android Versión 7 o superior con una resolución de pantalla de 1080 x 1920 píxeles (HD) de espacio de 10 Gb y un CPU x86.

Figura 36
Configuración del Simulador de Pantalla Móvil

| Type | Name | Play Store | Resolution | API | Target | CPU/ABI | Size on Disk | Actions |
|------|-----------------|------------|---------------|-----|--------------|---------|--------------|---------|
| | Nexus 5X API... | | 1080 x 192... | 25 | Android 7... | x86 | 10 GB | |

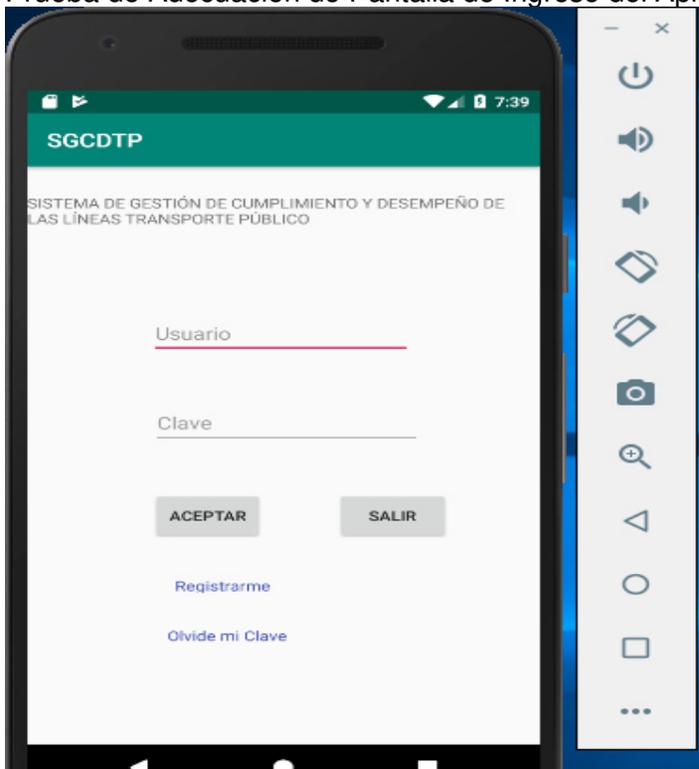
Fuente: Elaboración propia.

5.5.4 Resultado de la prueba de rendimiento del aplicativo

Al ejecutar el aplicativo en el simulador (Figura35) se evidenció que sí se adecuaba a las dimensiones de un equipo móvil estándar mostrando con claridad todos los campos de entrada de datos y mensajes de texto.

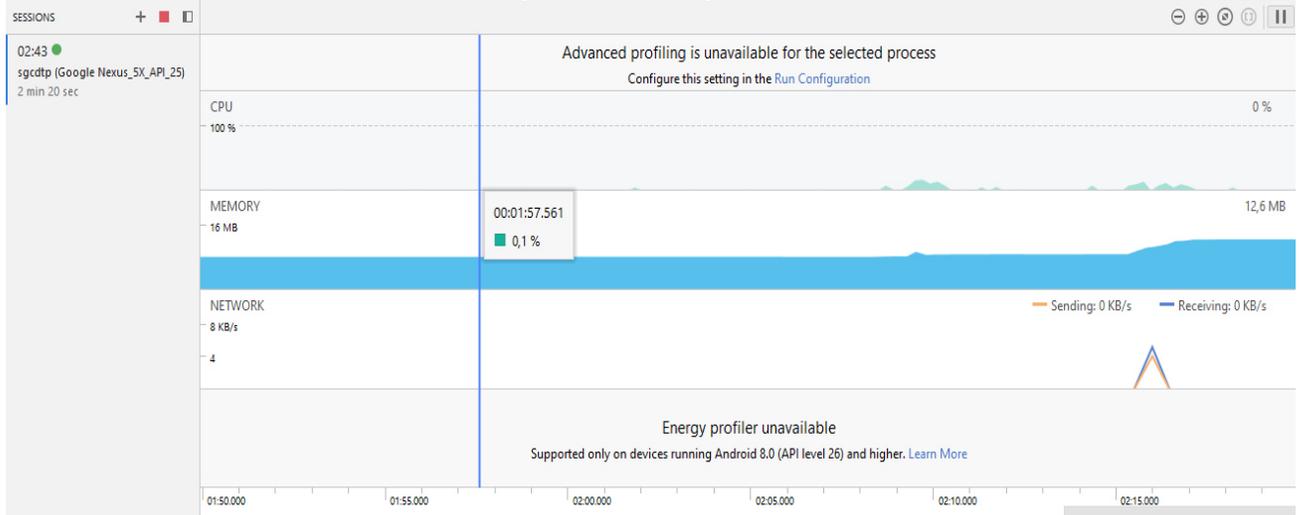
Para la prueba de rendimiento del aplicativo (Figura 36) se usó la opción de “profile” de la herramienta de desarrollo de Android Studio. La memoria se usó en promedio 0.1% del total, el CPU tuvo un uso poco significativo y el envío y recepción de los bytes estuvieron aceptables. Lo que demuestra que el aplicativo no tendrá problemas en su ejecución, además, demuestra que el aplicativo tendrá accesibilidad en una configuración de pantalla estándar sin afectar el rendimiento del equipo.

Figura 37
Prueba de Adecuación de Pantalla de Ingreso del Aplicativo Móvil



Fuente: Elaboración propia.

Figura 38
Resultado de Prueba de Rendimiento de Ejecución del Aplicativo

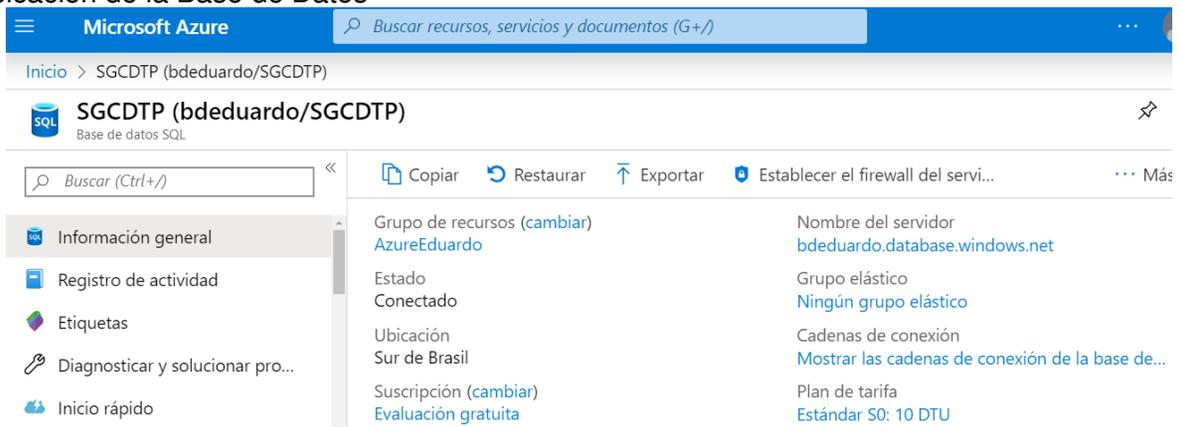


Fuente: Captura de Pantalla.

5.5.5 Prueba de base de datos

La base de datos se encuentra almacenada en los servidores de Microsoft Windows Azure con ubicación física en Brasil.

Figura 39
Ubicación de la Base de Datos

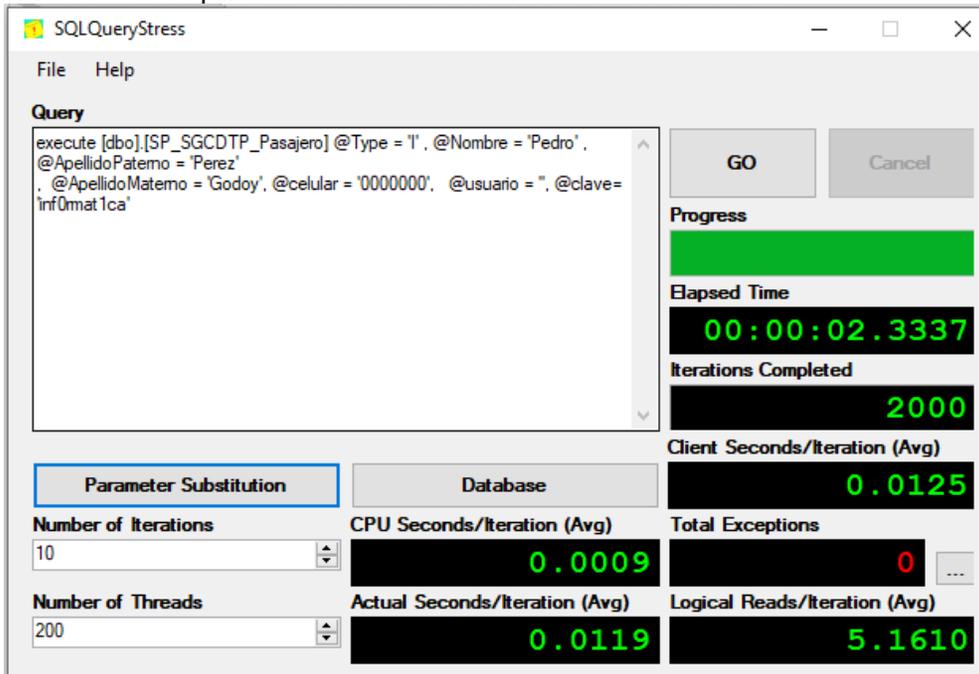


Fuente: Captura de Pantalla.

La prueba consistió en someter a la base de datos en técnicas de carga masiva o estrés a fin de determinar que la información se guarde y se mantenga coherente y consistente.

Para realizar esta prueba se utilizó la herramienta tecnológica SQLQueryStress v0.9.7.0 y se configuró para 200 usuarios con diez concurrencias por cada uno. Se realizó la prueba por cada Procedimiento Almacenado involucrado en la transacción a la base de datos.

Figura 40
Prueba de Estrés para Procedimiento de Creación de Usuarios



Fuente: Captura de Pantalla.

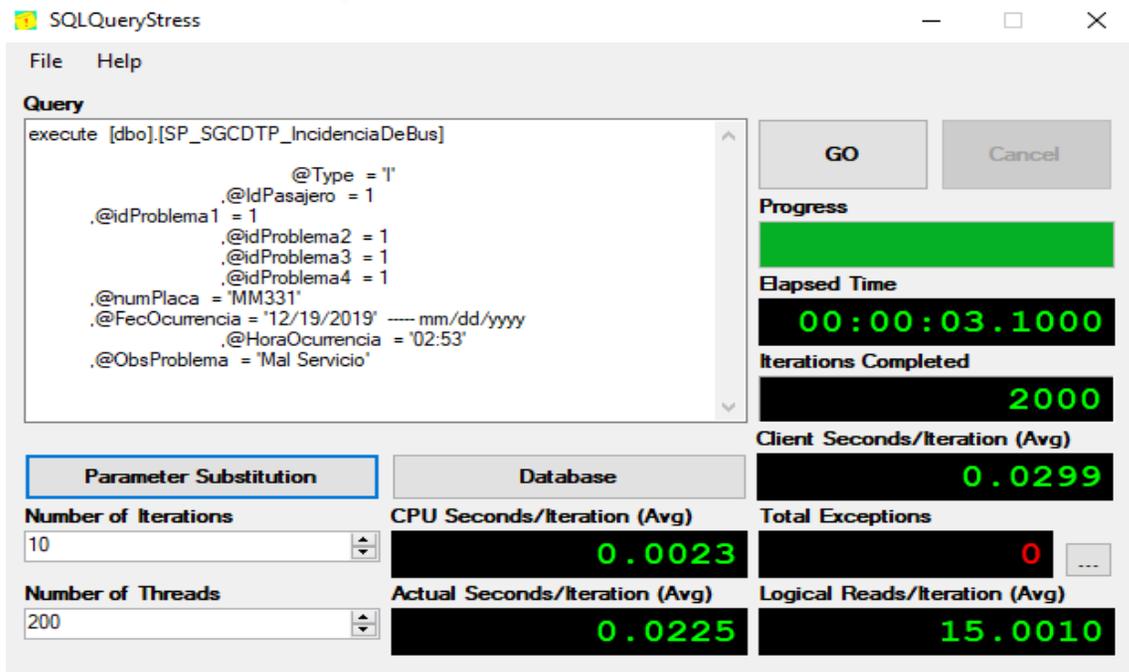
Figura 41
Prueba de Estrés para Procedimiento de Consulta de Usuarios

The screenshot shows the SQLQueryStress application window. The main query text is: `execute [dbo].[SP_S]{GCDTP_Pasajero} @Type = 'S' , @celular = '997326987'`. The interface includes a 'GO' button and a 'Cancel' button. A progress bar is shown at 100%. The elapsed time is 00:00:07.0071. The number of iterations completed is 2000. The client seconds per iteration (avg) is 0.2489. The total exceptions are 0. The number of iterations is set to 10, and the number of threads is set to 200. The CPU seconds per iteration (avg) is 0.0134, and the actual seconds per iteration (avg) is 0.0684. The logical reads per iteration (avg) is 59.0000.

| Parameter Substitution | Database | Client Seconds/Iteration (Avg) |
|--------------------------|--|--|
| Number of Iterations: 10 | CPU Seconds/Iteration (Avg): 0.0134 | Total Exceptions: 0 |
| Number of Threads: 200 | Actual Seconds/Iteration (Avg): 0.0684 | Logical Reads/Iteration (Avg): 59.0000 |

Fuente: Captura de Pantalla.

Figura 42
Prueba de Estrés para el Registro de Incidencias

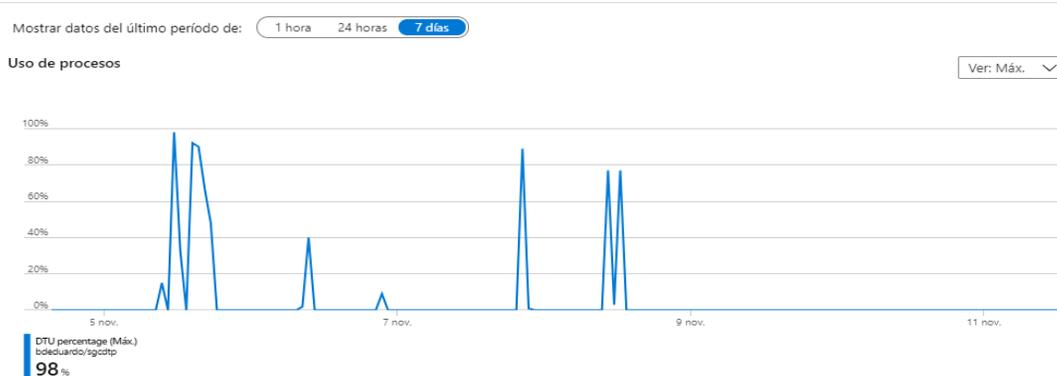


Fuente: Captura de Pantalla.

5.5.6 Resultado de prueba de la base de datos

El servidor de base de datos trabajo a un 98% del total de su capacidad sin presentar deterioro en el rendimiento o perdida de información lo que demuestra que la configuración, arquitectura y diseño están preparados para una carga masiva de información.

Figura 43
Resultado Estadístico de Uso de Base de Datos en el Servidor



Fuente: Captura de Pantalla.

Figura 44
Resultado de las Pruebas por Transacción Parte 1

SQLStress V3.2-001 © by David Kubelka

| SPID | Status | Login | HostName | BlkBy | DBName | Command | CPUTime | DiskIO | LastBatch | ProgramName | SPID1 | REQUESTID |
|------|----------|-------|----------|-------|--------|-------------------|---------|--------|---------------|-------------|-------|-----------|
| 1 | BACKGROU | sa | . | . | (null) | XTP_CKPT_AGENT | 0 | 0 | 12/13 10:03:4 | | 1 | 0 |
| 2 | BACKGROU | sa | . | . | (null) | LOG WRITER | 468 | 0 | 12/13 10:04:1 | | 2 | 0 |
| 3 | BACKGROU | sa | . | . | (null) | RECOVERY WRITER | 0 | 0 | 12/13 10:04:1 | | 3 | 0 |
| 4 | BACKGROU | sa | . | . | (null) | LAZY WRITER | 21093 | 0 | 12/13 10:04:1 | | 4 | 0 |
| 5 | BACKGROU | sa | . | . | master | SIGNAL HANDLER | 0 | 0 | 12/13 10:04:1 | | 5 | 0 |
| 6 | BACKGROU | sa | . | . | (null) | LOCK MONITOR | 46 | 0 | 12/13 10:04:1 | | 6 | 0 |
| 7 | BACKGROU | sa | . | . | (null) | XIO_RETRY_WORKER | 156 | 0 | 12/13 10:04:1 | | 7 | 0 |
| 8 | BACKGROU | sa | . | . | (null) | XIO_LEASE_RENEW/A | 156 | 0 | 12/13 10:04:1 | | 8 | 0 |
| 9 | BACKGROU | sa | . | . | master | BRKR TASK | 0 | 0 | 12/13 10:04:4 | | 9 | 0 |
| 10 | BACKGROU | sa | . | . | (null) | XE TIMER | 93 | 0 | 12/13 10:04:1 | | 10 | 0 |
| 11 | BACKGROU | sa | . | . | (null) | RESOURCE MONITOR | 1679609 | 0 | 12/13 10:04:1 | | 11 | 0 |
| 13 | BACKGROU | sa | . | . | (null) | XE DISPATCHER | 234 | 0 | 12/13 10:04:1 | | 13 | 0 |
| 15 | BACKGROU | sa | . | . | master | TRACE QUEUE TASK | 15 | 0 | 12/13 10:04:3 | | 15 | 0 |
| 16 | BACKGROU | sa | . | . | (null) | RECEIVE | 0 | 2 | 12/13 10:04:3 | | 16 | 0 |
| 17 | BACKGROU | sa | . | . | master | LOG POOL MEMORY | 173421 | 0 | 12/13 10:04:3 | | 17 | 0 |
| 18 | BACKGROU | sa | . | . | master | CHECKPOINT | 0 | 23 | 12/13 10:04:3 | | 18 | 0 |
| 19 | BACKGROU | sa | . | . | master | BRKR EVENT HNDLR | 0 | 45 | 12/13 10:04:4 | | 19 | 0 |
| 20 | BACKGROU | sa | . | . | master | BRKR TASK | 0 | 0 | 12/13 10:04:4 | | 20 | 0 |
| 21 | BACKGROU | sa | . | . | master | BRKR TASK | 109 | 0 | 12/13 10:04:4 | | 21 | 0 |
| 22 | BACKGROU | sa | . | . | master | BRKR TASK | 203 | 0 | 12/13 10:04:4 | | 22 | 0 |
| 55 | RUNNABLE | sa | EYDROGO | . | SGCDTP | SELECT INTO | 1110 | 81 | 12/17 12:03:2 | SQLStress | 55 | 0 |
| 12 | sleeping | sa | . | . | master | TASK MANAGER | 0 | 0 | 12/17 11:56:1 | | 12 | 0 |
| 14 | sleeping | sa | . | . | master | TASK MANAGER | 0 | 0 | 12/13 10:04:2 | | 14 | 0 |
| 23 | sleeping | sa | . | . | master | TASK MANAGER | 0 | 0 | 12/17 12:01:2 | | 23 | 0 |
| 24 | sleeping | sa | . | . | master | TASK MANAGER | 0 | 0 | 12/17 12:01:2 | | 24 | 0 |
| 25 | sleeping | sa | . | . | master | TASK MANAGER | 0 | 0 | 12/17 12:01:2 | | 25 | 0 |

Fuente: Captura de Pantalla.

Figura 45
Resultado de las Pruebas por Transacción Parte 2

SQLStress V3.2-001 © by David Kubelka

| SPID | Status | Login | HostName | BlkBy | DBName | Command | CPUTime | DiskIO | LastBatch | ProgramName | SPID1 | REQUESTID |
|------|----------|-------|----------|-------|--------|------------------|---------|--------|----------------|----------------|-------|-----------|
| 226 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:38 | SQLStress/Work | 226 | 0 |
| 227 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 31 | 0 | 12/17 12:02:39 | SQLStress/Work | 227 | 0 |
| 228 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 16 | 0 | 12/17 12:02:39 | SQLStress/Work | 228 | 0 |
| 229 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:34 | SQLStress/Work | 229 | 0 |
| 230 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 16 | 0 | 12/17 12:02:40 | SQLStress/Work | 230 | 0 |
| 231 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 16 | 0 | 12/17 12:02:40 | SQLStress/Work | 231 | 0 |
| 232 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 15 | 0 | 12/17 12:02:41 | SQLStress/Work | 232 | 0 |
| 233 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 15 | 0 | 12/17 12:02:41 | SQLStress/Work | 233 | 0 |
| 234 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:36 | SQLStress/Work | 234 | 0 |
| 235 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:42 | SQLStress/Work | 235 | 0 |
| 236 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 31 | 0 | 12/17 12:02:43 | SQLStress/Work | 236 | 0 |
| 237 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:43 | SQLStress/Work | 237 | 0 |
| 238 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 15 | 0 | 12/17 12:02:42 | SQLStress/Work | 238 | 0 |
| 239 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 30 | 0 | 12/17 12:02:40 | SQLStress/Work | 239 | 0 |
| 240 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 15 | 2 | 12/17 12:02:44 | SQLStress/Work | 240 | 0 |
| 241 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:45 | SQLStress/Work | 241 | 0 |
| 242 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:44 | SQLStress/Work | 242 | 0 |
| 243 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:46 | SQLStress/Work | 243 | 0 |
| 244 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 15 | 1 | 12/17 12:02:48 | SQLStress/Work | 244 | 0 |
| 245 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:45 | SQLStress/Work | 245 | 0 |
| 246 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:48 | SQLStress/Work | 246 | 0 |
| 247 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:46 | SQLStress/Work | 247 | 0 |
| 248 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 16 | 0 | 12/17 12:02:49 | SQLStress/Work | 248 | 0 |
| 249 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 31 | 0 | 12/17 12:02:49 | SQLStress/Work | 249 | 0 |
| 250 | sleeping | sa | EYDROGO | . | SGCDTP | AWAITING COMMAND | 0 | 0 | 12/17 12:02:52 | SQLStress/Work | 250 | 0 |

Fuente: Captura de Pantalla.

Las pruebas de estrés a la base de datos demostraron que un escenario de doscientos usuarios concurrentes la base de datos estaría preparada para soportarla.

5.5.7 Pruebas de caja negra

Este tipo de pruebas funcionales se centran en los resultados, donde se manejan la entrada a un módulo y se da un valor esperado para la salida sin tener en cuenta el funcionamiento interno.

Aplicando esta definición al Aplicativo móvil desarrollado en la presente Tesis podemos describir los siguientes procesos de prueba aplicados directamente a las interfaces de usuario:

5.5.7.1 Procedimiento: Registrar nuevo usuario

El aplicativo móvil tuvo la capacidad de realizar la validación de los datos ingresados por el usuario y emitir un mensaje de error si alguno de ellos no es válido.

Tabla 32
Pruebas de caja negra para el procedimiento: Registrar Nuevo Usuario

| Campo: | Datos Válidos: | Resultado: | Datos No Válidos: | No Resultado: |
|------------------|---|--|---|---|
| Nombre | Alfabético de longitud máxima 20, No puede estar en blanco. | Usuario Registrado satisfactoriamente. | Contiene números o se encuentra en blanco. | Error: Alguno de los datos introducidos no es correcto. |
| Apellido_Paterno | Alfabético de longitud máxima 20, No puede estar en blanco. | Los datos fueron almacenados correctamente en la Tabla Pasajero de la Base de Datos. | Contiene números o caracteres especiales, se encuentra en blanco. | La base de Datos no sufrió ninguna modificación. |
| Apellido_Materno | Alfabético de longitud máxima 20, No | | Contiene números o caracteres | |

| Campo: | Datos Válidos: | Resultado: | Datos No Válidos: | No Resultado: |
|---------------|--|-------------------|---|----------------------|
| | puede estar en blanco. | | especiales, se encuentra en blanco. | |
| DNI | Numérico de longitud 8. No puede estar en blanco. | | Contiene letras, su longitud es menor de 8 o se encuentra en blanco. | |
| Celular | Numérico. No puede estar en blanco. | | Contiene letras, su longitud es menor de 10 o se encuentra en blanco. | |
| Usuario | Alfabético de longitud máxima 10, No puede estar en blanco. | | Contiene números o caracteres especiales, se encuentra en blanco. | |
| Clave | Alfanumérica longitud entre 6 y 12 caracteres. No puede estar en blanco. | | Su longitud es menor de 6 caracteres. Se encuentra en blanco. | |

Fuente: Elaboración propia.

5.5.7.2 Procedimiento: Recuperar datos de usuario

El aplicativo estuvo en la capacidad de realizar la validación del número introducido por el usuario, la búsqueda del dato introducido en la tabla de Pasajero, la recuperación de los datos de acceso y el envío de dichos datos a través de un mensaje de texto.

Tabla 33
Pruebas de caja negra para el procedimiento: Recuperar datos de usuario.

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No | Resultado: |
|---------------|-------------------------------------|---|---|-----------|--|
| Celular | Numérico. No puede estar en blanco. | Usuario recuperado satisfactoriamente. | Contiene letras, su longitud es menor de 10 o se encuentra en blanco. | | Error: El número introducido no se encuentra registrado. |
| | Número encontrado en la BD. | Los datos fueron enviados vía mensaje de texto. | Número encontrado en la BD. | no en | |

Fuente: Elaboración propia.

5.5.7.3 Procedimiento: Registrar incidencias

El aplicativo logró validar los datos de entrada suministrados por el usuario y registrar en la Base de Datos los datos correspondientes a la Incidencia ocurrida.

Tabla 34
Pruebas de caja negra para el procedimiento: Registrar Incidencias

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No | Resultado: |
|---------------|--|------------------------------|-------------------------------------|-----------|----------------------------|
| Placa | Alfanumérico de longitud máxima 20, No | Los datos fueron almacenados | Contiene caracteres especiales o se | | Error: Alguno de los datos |

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|--------------------|---|---------------------------------|---|--|
| | puede estar en blanco. | correctamente en la Tabla | encuentra en blanco. | introducidos no son correctos. |
| Fecha - Hora | Con formato dd/mm/aa hh:mm:ss | Incidenias de la Base de Datos. | Se encuentra en blanco. | La base de Datos no sufrió ninguna modificación. |
| Tipo de Incidencia | Selección de 1 de las opciones | | Contiene números o se encuentra en blanco. | |
| Comentario | Alfanumérico de longitud máxima de 200. No puede estar en blanco. | | Contiene letras, su longitud es menor de 10 o se encuentra en blanco. | |

Fuente: Elaboración propia.

5.5.7.4 Procedimiento: Visualizar información de bus

En esta interfaz el aplicativo logró que el usuario visualizara la información de las incidencias cometidas por la unidad de transporte correspondiente al número de placa ingresado.

Tabla 35
Pruebas de caja negra para el procedimiento: Visualizar Información de Bus.

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|---------------|---|--|--|--|
| Placa | Alfanumérico. No puede estar en blanco. | Los datos correspondientes fueron mostrados como reporte a | Contiene caracteres especiales o se encuentra en blanco. | Error: El Dato introducido está errado o no se encuentra registrado. |

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|---------------|---------------------------|-------------------------|------------------------------|----------------------|
| | Dato encontrado en la BD. | través de la interface. | Dato no encontrado en la BD. | |

Fuente: Elaboración propia.

5.5.7.5 Procedimiento: Visualizar Información de empresa de Transporte (Fiscalizador)

En esta sección el aplicativo logró que el usuario fiscalizador visualizara la información del total de incidencias cometidas por la empresa de transporte correspondiente al número de RUC ingresado.

Tabla 36
Pruebas de caja negra del procedimiento: Visualizar Información de Empresas

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|---------------|---|--|---|--|
| RUC | Alfanumérico. No puede estar en blanco. Dato encontrado en la BD. | Los datos correspondientes fueron mostrados como reporte a través de la interface. | Contiene caracteres especiales o se encuentra en blanco. Dato no encontrado en la BD. | Error: El Dato introducido está errado o no se encuentra registrado. |

Fuente: Elaboración propia.

5.5.7.6 Procedimiento: Registrar usuario fiscalizador

En esta interfaz el aplicativo capturó los datos de entrada correspondientes a la identificación del usuario fiscalizador, validar cada uno de ellos y almacenarlos en la Base de datos.

Tabla 37
Pruebas de caja negra para el procedimiento: Registrar Usuario Fiscalizador

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|---------------|--------------------------------------|--------------------|-----------------------|----------------------------|
| Nombre | Alfabético de longitud máxima 20, No | Usuario Registrado | Contiene números o se | Error: Alguno de los datos |

| Campo: | Datos Válidos: | Resultado: | Datos No Válidos: | Resultado: |
|------------------|---|--|---|--|
| | puede estar en blanco. | satisfactoriamente. | encuentra en blanco. | introducidos no es correcto. |
| Apellido_Paterno | Alfabético de longitud máxima 20, No puede estar en blanco. | Los datos fueron almacenados correctamente en la Tabla Pasajero de la Base de Datos. | Contiene números o caracteres especiales, se encuentra en blanco. | La base de Datos no sufrió ninguna modificación. |
| Apellido_Materno | Alfabético de longitud máxima 20, No puede estar en blanco. | | Contiene números o caracteres especiales, se encuentra en blanco. | |
| DNI | Numérico de longitud 8. No puede estar en blanco. | | Contiene letras, su longitud es menor de 8 o se encuentra en blanco. | |
| Celular | Numérico. No puede estar en blanco. | | Contiene letras, su longitud es menor de 10 o se encuentra en blanco. | |
| Usuario | Alfabético de longitud máxima 10, No puede estar en blanco. | | Contiene números o caracteres especiales, se encuentra en blanco. | |

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|---------------|---|-------------------|---|----------------------|
| Clave | Alfanumérico, longitud entre 6 y 12 caracteres, No puede estar en blanco. | | Su longitud es menor de 6 caracteres. Se encuentra en blanco. | |

Fuente: Elaboración propia.

5.5.7.7 Procedimiento: Registrar empresa de transporte

En esta fase el aplicativo tuvo la capacidad de capturar los datos de entrada asignados a la empresa de transporte, validarlos y almacenarlos en la Base de Datos.

Tabla 38
Pruebas de caja negra para el procedimiento: Registrar Empresa de Transporte

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No Resultado: |
|---------------|---|--|--|--|
| RUC | Alfanumérico. No puede estar en blanco. | Los datos fueron almacenados | Contiene caracteres especiales o se encuentra en blanco. | |
| Descripción | Alfabético. No puede estar en blanco. | correctamente en la Tabla EmpTransporte de la Base de Datos. | Contiene caracteres especiales o se encuentra en blanco. | Error: Los Datos Ingresados están errados. |
| Dirección | Alfabético. No puede estar en blanco. | | Contiene caracteres especiales o se | |

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No | Resultado: |
|---------------|-----------------------|-------------------|-----------------------|-----------|-------------------|
| | | | encuentra en | | |
| | | | blanco. | | |

Fuente: Elaboración propia.

5.5.7.8 Procedimiento: Registrar buses

El aplicativo estuvo en la capacidad de realizar la validación del número introducido por el usuario

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No | Resultado: |
|---------------|--|---|--|--|-------------------|
| RUC | Alfanumérico. No puede estar en blanco. | Los datos fueron almacenados correctamente en la Tabla EmpTransporte de la Base de Datos. | Contiene caracteres especiales o se encuentra en blanco. | | |
| | Alfabético. No puede estar en blanco. | | Contiene caracteres especiales o se encuentra en blanco. | Error: Los Datos Ingresados están errados. | |
| | Alfabético. No puede estar en blanco. | | Contiene caracteres especiales o se encuentra en blanco. | | |

5.5.7.9 Procedimiento: Registrar Problema

| Campo: | Datos Válidos: | Resultado: | Datos Válidos: | No | Resultado: |
|---------------|---------------------------------------|--|--|-----------|--|
| Descripción | Alfabético. No puede estar en blanco. | Los datos fueron almacenados correctamente en la Tabla Problema de la Base de Datos. | Contiene caracteres especiales o se encuentra en blanco. | | Error: Los Datos Ingresados están errados. |

5.6 Resultado de las pruebas al aplicativo

Los resultados muestran que se logró desarrollar un aplicativo móvil que permite registrar y mostrar información del cumplimiento y desempeño del servicio brindado por las empresas de transporte público de pasajeros en la ciudad de Lima.

El resultado de las pruebas muestra que las interfaces desarrolladas del aplicativo móvil permiten el registro de usuarios pasajeros y fiscalizadores al aplicativo móvil.

Se logró desarrollar la interface de usuario que permite registrar información sobre el servicio brindado por las empresas de transporte público de pasajeros en la ciudad de Lima. Se registró información de las incidencias que comprendían: incumplimiento de ruta y paraderos, mala atención del conductor o cobrador, irrespeto de la luz roja y el estado de limpieza de la unidad de transporte.

Las pruebas muestran que la base de datos registra y almacena adecuadamente la información que se obtiene a través del aplicativo móvil.

Como resultado se obtuvo que el 91% de los encuestados logro registrarse satisfactoriamente en el aplicativo móvil desarrollado.

Otro resultado es que el 91% de los encuestados registró información en el aplicativo móvil de las incidencias ocurridas durante la prestación de servicio de transporte público en la ciudad de Lima

El 77% de los encuestados respondieron que la información que visualizaron es de utilidad por lo que se logró desarrollar reportes en el aplicativo móvil para los usuarios pasajeros sobre el desempeño y cumplimiento del servicio brindado por los buses de transporte público en la ciudad de Lima.

El desarrollo del aplicativo cuenta con un diseño gráfico adecuado para visualizar la información en cualquier dispositivo móvil. El resultado evidencia que el aplicativo cumple los requerimientos tecnológicos de rendimiento, adaptabilidad y facilidad de uso.

Por lo tanto, se puede concluir que el desarrollo del aplicativo móvil fue aceptable, de utilidad y logró cumplir con los objetivos planteados en el presente documento, logrando así el registro del cumplimiento y desempeño del servicio de transporte público en Lima.

Tabla 39
Resumen de Pruebas Realizadas.

| Objetivo | Prueba Realizada | Resultado |
|---|---|---|
| Desarrollar un aplicativo móvil que permita registrar y mostrar información del cumplimiento y desempeño del servicio brindado por las empresas de transporte público en la ciudad de Lima. | Prueba de funcionalidad, usabilidad y utilidad. Preguntas 4 y 11 de la encuesta aplicada. | 80% de las personas que probaron el aplicativo indicaron que están de acuerdo con utilizar una App para registrar incidencias ocurridas durante la prestación del servicio de transporte público. Y a su vez un 67% de los encuestados calificó el funcionamiento del |

| Objetivo | Prueba Realizada | Resultado |
|---|---|---|
| | | aplicativo móvil como Excelente, un 24% como Bueno. |
| Diseñar las interfaces de usuario que permitan el registro de pasajeros y fiscalizadores al aplicativo móvil. | Prueba de usabilidad de aplicativo. Pregunta 5 de la encuesta realizada. Prueba de base de datos. | 91% de las personas que usaron el aplicativo pudieron registrarse como nuevo usuario y acceder a la siguiente interface. |
| Diseñar la interface de usuario que permita registrar la información en tiempo real sobre las incidencias ocurridas durante la prestación de servicio de transporte público en la ciudad de Lima. | Prueba de usabilidad de aplicativo. Pregunta 6 de la encuesta realizada. Prueba de base de datos. | 91% de las personas que usaron el aplicativo lograron registrar en tiempo real las incidencias ocurridas durante la prestación de servicio de transporte público. |
| Diseñar la Base de datos para almacenar la información sobre las incidencias ocurridas durante la prestación de servicio de transporte público en la ciudad de Lima. | Preguntas 7, 8 y 9 de la encuesta aplicada. Prueba de base de datos. | La mayoría de las personas que usaron el aplicativo pudieron registrar información en varias interfaces del aplicativo correctamente. |

| Objetivo | Prueba Realizada | Resultado |
|---|--|--|
| Crear un reporte en el aplicativo móvil que muestre al usuario pasajero información sobre el desempeño y cumplimiento del servicio brindado por los buses de transporte público en la ciudad de Lima. | Preguntas 7 y 8 de la encuesta realizada. Prueba de base de datos. | El 94% de las personas encuestadas pudo visualizar información sobre las incidencias presentadas en una unidad de transporte (Bus). Adicionalmente, el 77% aseguró que el nivel de utilidad de la información mostrada es Alto y un 14% Medio. |
| Crear un reporte en el aplicativo móvil que muestre al usuario fiscalizador información totalizada de las incidencias cometidas por las empresas de transporte en la ciudad de Lima. | Prueba de funcionalidad, usabilidad y utilidad. Preguntas 9 y 10 de la encuesta aplicada. Prueba de Base de Datos. | El 66% de las personas que uso el aplicativo considera que la Municipalidad de Lima debe tomar como insumo lo registrado en el aplicativo móvil al momento de evaluar la renovación de licencias de las líneas de transporte y un 70% que debe hacerlo al momento de evaluar la renovación de licencias de conducir de los conductores de Buses. |

Fuente: Elaboración Propia

Tabla 40
Resultado de las Pruebas No Funcionales

| Número | Atributo | Solicitudes | Resultado |
|---------------|-----------------|--|--|
| RQ-NF-001 | Seguridad | El aplicativo debe ser confiable para todos los usuarios y administradores. Debe garantizar el acceso a la | Las figuras 14, 15 y 16 muestran que los accesos para la modificación de la estructura de las tablas o la información está protegido mediante un usuario y |

| | | |
|-----------|---------------------------|--|
| | | información sea clave. El usuario debe tener clave autorizada y segura. para ingresar al aplicativo. La base de datos contiene una tabla donde se almacena los datos de los usuarios. |
| RQ-NF-002 | Usabilidad | <p>La interfaz del aplicativo móvil debe ser intuitiva y sencilla.</p> <p>En la Figura 28 se evidencia que al menos 91% de los encuestados pudieron registrar las incidencias en el aplicativo móvil.</p> |
| RQ-NF-003 | Consistencia y Coherencia | <p>La Figura 18 (Modelo Relacional) muestra la relación de las tablas a nivel de estructura lo que garantiza coherencia de información.</p> <p>El desarrollo debe garantizar coherencia y consistencia de la información tanto al momento de ser almacenados como al momento de ser modificados por cualquier usuario y en cualquier momento. Además, el aplicativo debe permitir registro y modificación simultánea.</p> <p>El anexo 15 y 16 evidencias que los procedimientos almacenados contienen sentencias transaccionales lo que garantiza consistencia al momento de grabar la información.</p> <p>Las Figuras 27 y 28 demuestra que el aplicativo y la base de datos permiten grabar información de manera concurrente.</p> |

Fuente: Elaboración Propia

5.7 Otros resultados obtenidos durante el proceso fueron los siguientes:

- Se cuenta con un aplicativo móvil desarrollado, validado y con interfaz intuitiva.
- Se cuenta con una base de datos que almacena la información de manera consistente, coherente y segura.

5.8 Comparación de los resultados obtenidos

Según el objetivo general, desarrollar un aplicativo móvil que permita registrar y mostrar información del cumplimiento y desempeño del servicio brindado por las empresas de transporte público de pasajeros en la ciudad de Lima, los resultados obtenidos evidencian que se logró alcanzar el objetivo, en comparación con Silva y Torres (15) quienes en el Proyecto Técnico: Calidad del Servicio de Transporte Urbano en la Ciudad de Cuenca, gracias a la encuesta que aplicaron lograron concluir que: "el 41% de los encuestados manifestó que la calidad del transporte es buena y el restante 59% de usuarios indicaron que es malo, ya que se sienten inconformes con factores como: la higiene, contaminación ambiental, el estado físico y ubicación de las paradas, tiempo de viaje del origen al destino, trato del conductor hacia los usuarios, irrespeto de los asientos para mujeres embarazadas, falta de información, dificultad para ingresar al autobús, incomodidad de los asientos". Este trabajo aportó algunos factores de inconformidad de los usuarios que pudieron incluirse en el desarrollo de esta tesis.

Según el objetivo, implementar una aplicación móvil que registre las incidencias en el transporte urbano de pasajeros en la ciudad de Lima, los resultados obtenidos evidencian que el 91% de los usuarios logró registrar y visualizar las siguientes incidencias: Mal servicio del Conductor o Cobrador No respetó la ruta y/o paraderos, Se pasó la luz roja, El bus estaba en malas condiciones o sucio; en comparación con Magallanes Reyes (18) en su Informe De Investigación Transporte Público de Pasajeros en Lima, donde obtuvo como resultado la confirmación de que el "sistema de transporte público en Lima Metropolitana se ve afectado por una serie de factores que repercuten en su calidad y eficiencia, entre los que se pueden mencionar: el incremento de la densidad poblacional, el aumento constante en las ventas de vehículos motorizados, el incremento en la cantidad de accidentes de tránsito, la contaminación ambiental, entre otros".

Según el objetivo, diseñar una aplicación móvil que facilite el registro, lectura y análisis de las incidencias ocurridas en el servicio de transporte urbano de pasajeros en la ciudad de Lima los resultados obtenidos evidencian que al 94% de los encuestados se les facilitó el registro y la visualización de la información de incidencias presentadas en una unidad de transporte (Bus), en comparación con Torres Sanchez (17), en su Tesis: Simulación Del Sistema De Transporte Urbano de La Ciudad de Tarapoto Para Planificar Escenarios Futuros, donde logró “determinar los escenarios actuales del sistema de transporte urbano, población con crecimiento acelerado, cantidad creciente de mototaxis, motocicletas, y vehículos mayores, y los problemas consecuentes en el servicio, estos escenarios permitieron delimitar las variables que se simuló para lograr obtener indicadores estadísticos en el futuro”.

Según el objetivo, diseñar una aplicación móvil que sea de utilidad tanto a los usuarios como a las entidades de control del transporte urbano en la ciudad de Lima, se evidenció que para el 77% de los encuestados le fue de utilidad la información visualizada a través del aplicativo en comparación con Heredia Bedoya (16), quien desarrolló un Modelo de satisfacción de los usuarios de transporte público tipo bus integrando variables latentes, donde concluyó que “las personas encuentran necesario que se realicen este tipo de estudios, argumentando que ellos quieren ser escuchados y tenidos en cuenta en las decisiones que se tomen acerca del transporte público, y los resultados de esta investigación demuestran que el usuario es un elemento importante dentro del proceso de transporte, aportando parámetros para su mejoría”.

CONCLUSIONES

Después de haber desarrolla la presente Tesis y de haber analizado los resultados obtenidos se llegó a las siguientes conclusiones:

Se desarrolló un aplicativo móvil que permitió registrar y mostrar información del cumplimiento y desempeño del servicio de transporte público de pasajeros en la ciudad de Lima.

Se desarrollo una aplicación móvil que permitió al 91% de los encuestados registrar las incidencias ocurridas durante la prestación del servicio de transporte urbano de pasajeros en la ciudad de Lima.

Se desarrollo una base de datos que permitió almacenar información que registran los usuarios sobre las incidencias ocurridas durante la prestación del servicio de transporte urbano de pasajeros en la ciudad de Lima.

Se desarrollo una aplicación móvil que facilitó al 94% de los encuestados visualizar la información referente a las incidencias ocurridas en el servicio de transporte urbano de pasajeros en la ciudad de Lima.

Se desarrollo un aplicativo móvil que para el 66% de las personas la Municipalidad de Lima debe tomar como insumo lo registrado para evaluar la renovación de licencias de líneas de transporte y el 70% considera que debe usar la información para evaluar la renovación de las licencias de los conductores.

Se desarrollo una aplicación móvil que para el 67% de los encuestados tiene excelente funcionamiento y para el 77% de los encuestados le resultado útil.

TRABAJOS FUTUROS

Se recomienda el desarrollo e implementación de nuevas funcionalidades al aplicativo móvil planteado utilizando tecnología GPS y Google Maps para el control, seguimiento y monitoreo de la calidad del servicio brindado por las empresas de transporte urbano de pasajeros.

Se recomienda analizar la implementación de un tablero de control sobre el cumplimiento y desempeño en base a las incidencias registradas por los pasajeros para poder realizar las notificaciones respectivas a las empresas de transporte que estén bajando su calificación y puedan realizar acciones de mejora para no tener problemas en la renovación de sus licencias.

Impulsar nuevas tecnologías de seguimiento y aseguramiento de la calidad de servicio brindado por las empresas de transporte de pasajeros en Lima Metropolitana.

Implementar un mapa, a través de Google Maps, en la cual se evidencie las zonas o lugares de mayores incidencias registradas con el servicio de transporte de pasajeros en Lima Metropolitana.

Se recomienda implementar nuevas funcionalidades al desarrollo del aplicativo, de tal manera que los buses de transporte de pasajeros puedan visualizar en línea las solicitudes de servicio por parte de los pasajeros en los paraderos autorizados.

REFERENCIAS BIBLIOGRÁFICAS

1. **La República, Diario.** Limeños pierden el 25% de sus ingresos por el tránsito vehicular. *Limeños pierden el 25% de sus ingresos por el tránsito vehicular.* [En línea] 15 de 04 de 2017. [Citado el: 15 de 06 de 2018.] <https://larepublica.pe/economia/865633-limenos-pierden-el-25-de-sus-ingresos-por-el-transito-vehicular>.
2. **Defensoría del Pueblo.** El Transporte Urbano en Lima Metropolitana: Un desafío en defensa de la vida. *El Transporte Urbano en Lima Metropolitana: Un desafío en defensa de la vida.* [En línea] Mayo de 2018. https://www.defensoria.gob.pe/wp-content/uploads/2018/05/informe_137.pdf.
3. **AAP, Asociación Automotriz del Perú.** Asociación Automotriz del Perú. *Importación de Vehículos nuevos.* [En línea] 15 de 07 de 2019. https://aap.org.pe/estadisticas/importaciones_vehiculos_nuevos/importacion-de-vehiculos-nuevos-2019/.
4. **Rpp Obras.** El 75% de limeños se moviliza en transporte público y el 25% viaja 2 horas. [En línea] 24 de 09 de 2016. [Citado el: 15 de 06 de 2018.] <http://rpp.pe/lima/obras/el-75-de-limenos-se-moviliza-en-transporte-publico-y-el-25-viaja-2-horas-noticia-992721>.
5. **RPP, Diario.** Congestión vehicular y la autoridad de transporte urbano de Lima y Callao. *Congestión vehicular y la autoridad de transporte urbano de Lima y Callao.* [En línea] 03 de Diciembre de 2018. <https://rpp.pe/columnistas/alexandrealmeyda/congestion-vehicular-y-la-autoridad-de-transporte-urbano-de-lima-y-callao-noticia-1166651>.
6. **MORAN Muñiz, Johan.** Problemática del Transporte y Tránsito - Presupuesto participativo 2017. *Problemática del Transporte y Tránsito - Presupuesto participativo 2017.* [En línea] 2017. http://www.munimolina.gob.pe/descargas/pdf/Presupuesto_Participativo_2017/6_2_problematika_transporte_transito.pdf.
7. **Gestión, Diario.** Antigüedad del parque de transporte público en Lima y Callao es de 12.5 años. *Antigüedad del parque de transporte público en Lima y Callao es de 12.5 años.* [En línea] 12 de Setiembre de 2018. <https://gestion.pe/economia/antigüedad-parque-transporte-publico-lima-callao-12-5-anos-244201>.
8. **Congreso de la Republica del Perú.** Ley 27181 Ley General de Transporte y Tránsito Terrestre. *Ley 271811.* Lima, Lima, Perú : s.n., 7 de octubre de 2015.

9. **Municipalidad de Lima Metropolitana.** Sub Gerencia de Transporte Urbano. *Sub Gerencia de Transporte Urbano.* [En línea] 15 de 10 de 2019.

<http://www.gtu.munlima.gob.pe/index.php/es-es/nosotros/subgerencias/subgerencia-de-regulacion-del-transporte-srt>.

10. **Municipalidad de Lima.** Obtención / renovación de credencial de conductor / conductor-cobrador / cobrador. *Obtención / renovación de credencial de conductor / conductor-cobrador / cobrador.* [En línea] 14 de 10 de 2019. <http://www.munlima.gob.pe/gerencias/item/27479-credencial-para-conductor-y-cobrador-del-servicio-de-transporte-regular-no-regular-y-especial-de-personas-obtencion-renovacion-de-credencial-de-conductor-conductor-cobrador-cobrador>.

11. **ALEGRE Escorza, Mariana.** Transporte urbano: ¿Cómo resolver la movilidad en Lima y Callao? *Documento de Política.* Lima, Lima, Perú : CIES - Consorcio de Investigación económica y social, 2016.

12. **AARON Jara, Alata.** Transporte público sostenible en Lima: Una aproximación al análisis coste-beneficio entre los sistemas BRT y LRT. Lima, Lima, Perú : Universidad Politécnica de Cataluña, 2016.

13. **Movertis.** Las 7 mejores apps para transportistas. *Las 7 mejores apps para transportistas.* [En línea] 15 de Junio de 2018. <https://www.movertis.com/blog/transporte/7-mejores-apps-para-transportistas>.

14. **RPP, Diario.** Moovit y TuRuta: las aplicaciones que te guían por el transporte público de Lima. *Moovit y TuRuta: las aplicaciones que te guían por el transporte público de Lima.* [En línea] 18 de 01 de 2019. <https://rpp.pe/tecnologia/apps/transporte-publico-en-lima-moovit-y-tu-ruta-aplicaciones-que-te-ayudan-a-moverte-por-la-ciudad-android-e-ios-noticia-1162889>.

15. **SILVA Morales, Neiro Andrés y TORRES Sánchez, Camilo.** CALIDAD DEL SERVICIO DE TRANSPORTE URBANO EN LA CIUDAD DE CUENCA. *CALIDAD DEL SERVICIO DE TRANSPORTE URBANO EN LA CIUDAD DE CUENCA.* Cuenca, Ecuador : UNIVERSIDAD POLITÉCNICA SALESIANA, 05 de 2017.

16. **HEREDIA Bedoya, Julieth Katherine.** Modelo de satisfacción de los usuarios de transporte público tipo bus integrando variables latentes. *Modelo de satisfacción de los usuarios de transporte público tipo bus integrando variables latentes.* Medellín, Colombia : Universidad Nacional de Colombia, 01 de 2015.

17. **TORRES Sánchez, Carlos Enrique.** Simulación del Sistema de Transporte Urbano de La Ciudad de Tarapoto para Planificar Escenarios Futuros. *Simulación del Sistema de*

Transporte Urbano de La Ciudad de Tarapoto para Planificar Escenarios Futuros. Tarapoto, Perú : UNIVERSIDAD NACIONAL DE SAN MARTÍN - T, 01 de 2013.

18. **MAGALLANES Reyes, Claudio Alberto**. Transporte Público de Pasajeros en Lima. *Transporte Público de Pasajeros en Lima*. Lima, Perú : Congreso de la República, 24 de 11 de 2014.

19. **CPI-INEI**. Marketreport - Perú: población 2019. *Marketreport - Perú: población 2019*. [En línea] Mayo de 2019.

http://cpi.pe/images/upload/paginaweb/archivo/26/mr_poblacional_peru_201905.pdf.

20. **JOSE Luis, Bonifaz y CARLOS, Aparicio**. *La Gestión del Sistema de Transporte Público Peruano al 2050*. Lima : s.n., 2013.

21. **BORJAS Giraldo, GianCarlo**. *Análisis, diseño e implementación de un sistema de información para la administración de horarios y rutas en empresas de transporte público*. 2013.

22. **Cómo Vamos**. *Cómo vamos en movilidad. Cómo vamos en movilidad*. [En línea] 2016. <http://www.limacomovamos.org/cm/wp-content/uploads/2016/09/InformeMovilidad2015-1.pdf>.

23. **Municipalidad de Lima Metropolitana**. Ordenanza 1599. *Ordenanza que Regula la Prestación del Servicio de Transporte Público Regular de Personas en Lima Metropolitana*. Lima, Perú : Municipalidad de Lima, 17 de 04 de 2012.

24. **JARA Risco, Maria y VASQUEZ Córdova, Joaquin**. El Servicio de Transporte Público Urbano de Personas: Libre mercado y regulación. *El Servicio de Transporte Público Urbano de Personas: Libre mercado y regulación*. 10 de Consulta (2019).

25. **MENDOZA Alegre, Milgaros**. Servicio de transporte terrestre de pasajeros. [En línea] [Citado el: 28 de 05 de 2019.]

<http://revistas.pucp.edu.pe/index.php/derechoadministrativo/article/download/14045/14667>.

26. **Municipalidad de Lima**. Gerencia de Transpor Urbano. [En línea] [Citado el: 28 de 05 de 2019.] <http://www.gtu.munlima.gob.pe/index.php/es-es/nosotros/gerencias>.

27. **López Castañeda, Maryuri** . Qué son las Apps y Tipos de Apps. [En línea] [Citado el: 18 de 03 de 2020.] <http://univirtual.utp.edu.co/pandora/recursos/2000/2591/2591.pdf>.

28. **ServiSoftCorp © 2010**. Definición y cómo funcionan las aplicaciones móviles. [En línea] [Citado el: 18 de Abril de 2020.] <https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>.

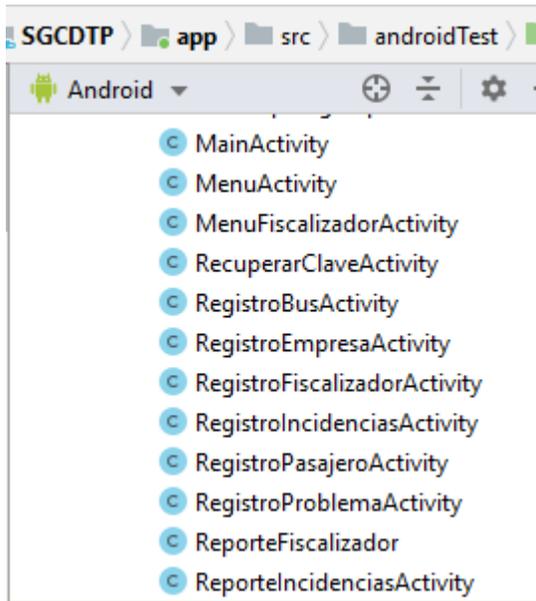
29. *Tecnología Móvil: desarrollo de sistemas y aplicaciones*. **GARITA Araya, Raúl Alberto**. 2013, E-Ciencias de la Información, págs. Volumen 3, número 2, artículo 2.

30. *La Usabilidad en Ingeniería de Software*. **Sanchez, Walter**. 2011, Reporte de Investigación, pág. 8.
31. *Guía metodológica de pruebas especializada en control y calidad de aplicaciones móviles fundamentados en la norma ISO 9241-11*. **Ramos Lopez1, Crhistian Michael, y otros**. 2017, 4to Congreso Internacional AmITIC 2017, Popayán, Colombia., pág. 71.
32. *Agile: The Necessitate Of Contemporary Software Developers*. **Khurana, Harsimarjeet y J.S. Sohal**. s.l. : International Journal of Engineering Science and Technology, 2011, International Journal of Engineering Science and Technology., pág. 3.
33. **NAVARRO Cadavid, Andrés, FERNANDEZ Martínez, Juan Daniel y MORALES Vélez, Jonathan**. *Revisión de metodologías ágiles para el desarrollo de software*. s.l. : Grupo de investigación i2T, 2013.
34. **40defiebre.com**. ¿Qué es el diseño responsive? ¿Qué es el diseño responsive? [En línea] <https://www.40defiebre.com/que-es/disenio-responsive>.
35. **Quispe Limaylla, Josué**. *Desarrollo de un aplicativo web para la reserva del servicio de estacionamiento mediante geolocalización en la probvincia de huancayo*. Huancayo : Universidad Continental, 2018.
36. *Metodología Scrum*. **Trigas Gallego, Manuel**. 2019, Gestión de Proyectos Informáticos.

ANEXOS

Anexo A

Estructura de desarrollo del aplicativo



Anexo B Código de Inicio del Aplicativo

```
package com.example.sgcdtp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.StrictMode;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class MainActivity extends AppCompatActivity {

    EditText editTextUsuario, editTextClave;
    Button
    botonAceptar, botonSalir, botonRegistrarPasajero, botonRegistrarFiscalizador,
    botonOlvideClave;

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

        editTextUsuario=(EditText) findViewById(R.id.editTextNumAsientos);
        editTextClave=(EditText) findViewById(R.id.editTextClave);
        botonAceptar=(Button) findViewById(R.id.botonAceptar);
        botonSalir=(Button) findViewById(R.id.botonCerrar);

        botonRegistrarPasajero=(Button) findViewById(R.id.botonRegistrarPasajero);
        botonOlvideClave=(Button) findViewById(R.id.botonOlvideClave);

        botonAceptar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ConsultarBD();
            }
        });

        botonSalir.setOnClickListener(new View.OnClickListener() {
            @Override
```

```

        public void onClick(View v) {

            Intent intent = new Intent(Intent.ACTION_MAIN);
            intent.addCategory(Intent.CATEGORY_HOME);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);

        }
    });

    buttonRegistrarPasajero.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v) {

            Intent i3 = new Intent(MainActivity.this,
            RegistroPasajeroActivity.class);
            startActivity(i3);
        }
    });

    buttonOlvideClave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent i6 = new Intent(MainActivity.this,
            RecuperarClaveActivity.class);
            startActivity(i6);
        }
    });

    public Connection conexionDBSQL() {

        Connection conexion = null;

        try {

            StrictMode.ThreadPolicy policy = new
            StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
            Class.forName("net.sourceforge.jtds.jdbc.Driver").newInstance();
            conexion =
            DriverManager.getConnection("jdbc:jtds:sqlserver://bduardo.database.windows.
            net:1433;databaseName=SGCDTP;user=usuario;password=<A13jandr0>");
        } catch (Exception e) {
            Toast.makeText(getApplicationContext(), e.getMessage(),
            Toast.LENGTH_SHORT).show();
        }
        return conexion;
    }

    public void ConsultarBD() {

```

```

        if (ValidarVariables()) {

            try {

                PreparedStatement pst=conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Pasajero] @Type = 'S', @usuario = '" +
editTextUsuario.getText().toString()+"' , @clave ='" +
editTextClave.getText().toString()+"'");

                ResultSet rs = pst.executeQuery();

                if (rs.next()) {

                    Toast.makeText(getApplicationContext(), "Encontro
Registro", Toast.LENGTH_SHORT).show();
                    Intent i2 = new Intent(MainActivity.this,
MenuActividad.class);
                    i2.putExtra("IdPasajero", rs.getString(1) );
                    i2.putExtra("usuario", rs.getString(2));
                    startActivity(i2);
                }
                else {

                    PreparedStatement pst2=conexionDBSQL().prepareStatement("
EXEC [dbo].[SP_SGCDTP_Fiscalizador] @Type = 'S', @usuario = '" +
editTextUsuario.getText().toString()+"' , @clave ='" +
editTextClave.getText().toString()+"'");

                    String base2 = conexionDBSQL().getCatalog();
                    ResultSet rs2 = pst2.executeQuery();

                    if (rs2.next()) {

                        Toast.makeText(getApplicationContext(), "Encontro
Registro", Toast.LENGTH_SHORT).show();
                        Intent i7 = new Intent(MainActivity.this,
MenuFiscalizadorActividad.class);
                        i7.putExtra("IdFiscalizador", rs2.getString(1) );
                        i7.putExtra("usuario", rs2.getString(2));
                        startActivity(i7);
                    }
                    else
                    {
                        Toast.makeText(getApplicationContext(), "Usuario o
Clave no existe", Toast.LENGTH_SHORT).show();
                    }
                }
            }
            catch (SQLException e) {

                Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show
                ();
            }
        }
    }
}

```

```

    }
}

private boolean ValidarVariables() {

    if (editTextUsuario.getText().toString().trim().equalsIgnoreCase(""))
    {
        Toast.makeText(getApplicationContext(), "Debe Ingresar Usuario",
Toast.LENGTH_SHORT).show();
        return false;
    }
    if (editTextClave.getText().toString().trim().equalsIgnoreCase("")) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar Clave",
Toast.LENGTH_SHORT).show();
        return false;
    }

    return true;
}
}

```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y DESEMPEÑO DE LAS LINEAS TRANSPORTE PÚBLICO

Usuario

Clave

ACEPTAR SALIR

Registrar Pasajero

Olvide mi Clave

Anexo C Código de Opciones del Pasajero

```
package com.example.sgcdtp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MenuActivity extends AppCompatActivity {

    Button buttonSalir,buttonRegistrarInc,buttonVerInfo;
    String idPasajero2,usuario2;
    TextView textViewUsuario2;

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

        buttonSalir=(Button) findViewById(R.id.buttonCerrar);
        buttonRegistrarInc=
        (Button) findViewById(R.id.buttonRegistrarProblema);
        buttonVerInfo = (Button) findViewById(R.id.buttonVerInfo);

        RecibirDatos2();

        buttonVerInfo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent i7 = new Intent(MenuActivity.this,
                ReporteIncidenciasActivity.class);
                startActivity(i7);
            }
        });

        buttonSalir.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(Intent.ACTION_MAIN);
                intent.addCategory(Intent.CATEGORY_HOME);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
            }
        });

        buttonRegistrarInc.setOnClickListener(new View.OnClickListener() {
```

```

@Override
public void onClick(View v) {

    Intent i3 = new Intent(MenuActivity.this,
RegistroIncidenciasActivity.class);
    i3.putExtra("usuario3", usuario2);
    i3.putExtra("IdPasajero3", idPasajero2 );
    startActivity(i3);
    }

});
}

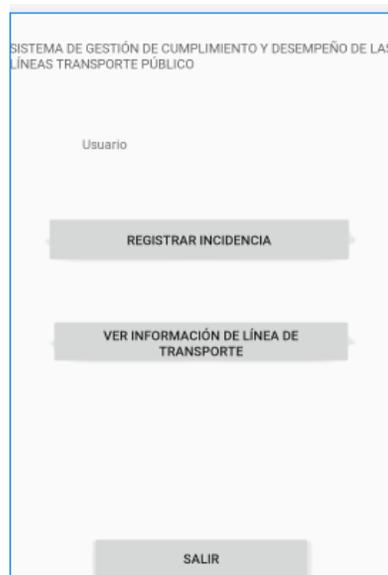
private void RecibirDatos2() {

    Bundle DatosExtras2 = getIntent().getExtras();

    idPasajero2 = DatosExtras2.getString("IdPasajero");
    usuario2 = DatosExtras2.getString("usuario");

    textViewUsuario2 = (TextView) findViewById(R.id.textViewUsuario2);
    textViewUsuario2.setText("Bienvenido " + usuario2 );
}
}

```



Anexo D Código de Opciones del Fiscalizador

```
package com.example.sgcdtp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MenuFiscalizadorActivity extends AppCompatActivity {

    Button
    buttonCerrar, buttonVerReporte, buttonRegistrarEmpresa, buttonRegistrarBus,
buttonRegistrarProblema, buttonRegistrarFiscalizador ;
    String idFiscalizador2, usuario2;
    TextView textViewUsuario2;

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

        buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
        buttonVerReporte= (Button) findViewById(R.id.buttonVerReporte);
        buttonRegistrarEmpresa =
        (Button) findViewById(R.id.buttonRegistrarEmpresa);
        buttonRegistrarBus = (Button) findViewById(R.id.buttonRegistrarBus);
        buttonRegistrarProblema =
        (Button) findViewById(R.id.buttonRegistrarProblema);
        buttonRegistrarFiscalizador =
        (Button) findViewById(R.id.buttonRegistrarFiscalizador);

        RecibirDatos2();

        buttonVerReporte.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent i7 = new Intent(MenuFiscalizadorActivity.this,
                ReporteFiscalizador.class);

                startActivity(i7);
            }
        });

        buttonRegistrarEmpresa.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent i9 = new Intent(MenuFiscalizadorActivity.this,
```

```

RegistroEmpresaActivity.class);
        startActivity(i9);
    }
});

    buttonRegistrarBus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent i10 = new Intent(MenuFiscalizadorActivity.this,
RegistroBusActivity.class);
            startActivity(i10);
        }
    });

    buttonRegistrarProblema.setOnClickListener(new View.OnClickListener()
{
        @Override
        public void onClick(View v) {

            Intent i11 = new Intent(MenuFiscalizadorActivity.this,
RegistroProblemaActivity.class);
            startActivity(i11);
        }
    });

    buttonRegistrarFiscalizador.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent i12 = new Intent(MenuFiscalizadorActivity.this,
RegistroFiscalizadorActivity.class);

            startActivity(i12);
        }
    });

    buttonCerrar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intent = new Intent(Intent.ACTION_MAIN);
            intent.addCategory(Intent.CATEGORY_HOME);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);

        }
    });

}

private void RecibirDatos2() {

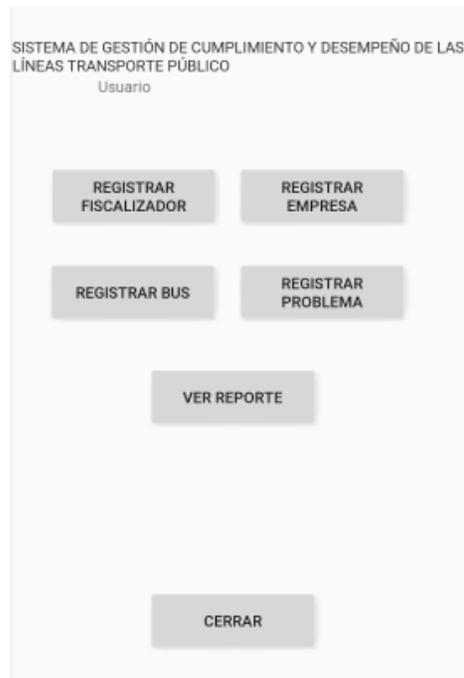
```

```
Bundle DatosExtras2 = getIntent().getExtras();

idFiscalizador2 = DatosExtras2.getString("IdFiscalizador");
usuario2 = DatosExtras2.getString("usuario");

textViewUsuario2 = (TextView) findViewById(R.id.textViewUsuario2);
textViewUsuario2.setText("Bienvenido " + usuario2 );

}
}
```



Anexo E Código de Recuperar Clave

```
package com.example.sgcdtp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseException;
import com.google.firebase.FirebaseTooManyRequestsException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.concurrent.TimeUnit;

public class RecuperarClaveActivity extends AppCompatActivity {

    Button buttonCerrar, buttonEnviar;
    EditText editTextUsuario, editTextCelular, editTextClave;
    String TextUsuario, TextCelular;

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

        editTextUsuario = (EditText) findViewById(R.id.editTextUsuario);
        editTextCelular = (EditText) findViewById(R.id.editTextCelular);
```

```

editTextClave = (EditText) findViewById(R.id.editTextClave);

buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
buttonEnviar=(Button) findViewById(R.id.buttonEnviar);

buttonCerrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

buttonEnviar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        editTextClave.setText("");
        EnviarClave();
    }
});

editTextClave.setVisibility(View.GONE);
}

private void EnviarClave() {

    if (ValidarVariables () ) {

        MainActivity ConexionSQL = new MainActivity();

        try {
            PreparedStatement
            pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Pasajero] @Type = 'C', @usuario = '" + TextUsuario +"' ,
@celular ='" + TextCelular +"'");
            ResultSet rs = pst.executeQuery( );

            if (rs.next()) {

                String mensaje = "Su clave es: " + rs.getString(1);
                editTextClave.setVisibility(View.VISIBLE);
                editTextClave.setText(mensaje);

                Toast.makeText(getApplicationContext(), "Su clave es: " +
mensaje, Toast.LENGTH_SHORT).show();

            }
            else {
                Toast.makeText(getApplicationContext(), "Usuario y/o
celular no existe", Toast.LENGTH_SHORT).show();
            }
        }
        catch (SQLException e) {

```

```

Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show
();
    }
}

private boolean ValidarVariables() {

    TextCelular = editTextCelular.getText().toString().trim();
    TextUsuario = editTextUsuario.getText().toString().trim();

    if (TextUsuario.equalsIgnoreCase("") || TextUsuario.length() < 3 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar su usuario
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }
    if (TextCelular.equalsIgnoreCase("") || TextCelular.length() != 9 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar Celular
Valido", Toast.LENGTH_SHORT).show();
        return false;
    }

    return true;
}
}

```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y DESEMPEÑO
DE LAS LÍNEAS TRANSPORTE PÚBLICO

Usuario _____

Celular _____

ENVIAR CERRAR

Anexo F Código de Registrar Incidencia de Pasajero

```
package com.example.sgcdtp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.Toast;

import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class RegistroIncidenciasActivity extends AppCompatActivity {

    Button buttonCerrar, buttonGrabar;

    String FechaOcurrencia, Observacion, NumPlaca, HoraMinOcurrencia;

    int IdPasajero, IdProblema1, IdProblema2, IdProblema3, IdProblema4;

    EditText
    editTextPlaca, editTextFechaOcurrencia, editTextHora, editTextMinuto, editTextObse
    rvacion;
    RadioButton radioButtonTodoOk;
    CheckBox checkBox1, checkBox2, checkBox3, checkBox4;

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

        RecibirDatos3();

        buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
        buttonGrabar=(Button) findViewById(R.id.buttonGrabar);

        editTextPlaca = (EditText) findViewById(R.id.editTextPlaca);
        editTextFechaOcurrencia =
        (EditText) findViewById(R.id.editTextFechaOcurrencia);
        editTextHora = (EditText) findViewById(R.id.editTextHora);
        editTextMinuto = (EditText) findViewById(R.id.editTextMinutos);
        editTextObservacion =
        (EditText) findViewById(R.id.editTextObservacion);
```

```

checkbox1= (CheckBox) findViewById(R.id.checkbox1);
checkbox2= (CheckBox) findViewById(R.id.checkbox2);
checkbox3= (CheckBox) findViewById(R.id.checkbox3);
checkbox4= (CheckBox) findViewById(R.id.checkbox4);

buttonGrabar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        GrabarIncedencia();
    }
});

buttonCerrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}
private void RecibirDatos3(){
    Bundle DatosExtras3 = getIntent().getExtras();
    IdPasajero = Integer.parseInt( DatosExtras3.getString("IdPasajero3"));
}

private void GrabarIncedencia() {
    FechaOcurrencia = editTextFechaOcurrencia.getText().toString();
    Observacion = editTextObservacion.getText().toString();
    NumPlaca = editTextPlaca.getText().toString();
    HoraMinOcurrencia = editTextHora.getText().toString() + ":" +
editTextMinuto.getText().toString() ;

    if (ValidarVariables () && ValidarFecha() && ValidarHoraMinuto())
{
    MainActivity ConexionSQL = new MainActivity();

    try {
        PreparedStatement
pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Incidencias] @Type = 'I', @IdPasajero = " + IdPasajero +
",@numPlaca =' " + NumPlaca + "', @idProblema1 = " + IdProblema1 + ",
@idProblema2 = " + IdProblema2 + ", @idProblema3 = " + IdProblema3 + " ,
@idProblema4 = " + IdProblema4 + ", @FecOcurrencia = '" + FechaOcurrencia +
"', @HoraMinOcurrencia = '" + HoraMinOcurrencia + "' , @Comentario = '" +
Observacion +"' " );
        pst.executeUpdate();
        pst.close();
        Toast.makeText(getApplicationContext(), "Datos Grabados
Correctamente", Toast.LENGTH_SHORT).show();
        finish();
    }
}
}

```

```

        }
        catch (SQLException e) {
Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show
();
        }
    }
}

private boolean ValidarVariables(){

    IdProblema1 = 0;
    IdProblema2 = 0;
    IdProblema3 = 0;
    IdProblema4 = 0;

    if (checkBox1.isChecked()) {
        IdProblema1 = 1;
    }

    if (checkBox2.isChecked()) {
        IdProblema2 = 1;
    }

    if (checkBox3.isChecked()) {
        IdProblema3 = 1;
    }

    if (checkBox4.isChecked()) {
        IdProblema4 = 1;
    }

    if (NumPlaca.toString().trim().equalsIgnoreCase("")){
        Toast.makeText(getApplicationContext(), "Debe Ingresar Placa del
Vehículo", Toast.LENGTH_SHORT).show();
        return false;
    }
    if (FechaOcurrencia.toString().trim().equalsIgnoreCase("")){
        Toast.makeText(getApplicationContext(), "Debe Ingresar Fecha de
Ocurrencia", Toast.LENGTH_SHORT).show();
        return false;
    }

    return true;
}

private boolean ValidarFecha() {

    try {

```

```

        SimpleDateFormat formatoFecha = new
SimpleDateFormat("dd/MM/yyyy");
        formatoFecha.setLenient(false);
        formatoFecha.parse(FechaOcurrencia);

    } catch (ParseException e) {
        Toast.makeText(getApplicationContext(), "La Fecha Invalida",
Toast.LENGTH_SHORT).show();
        return false;
    }

    return true;
}

private boolean ValidarHoraMinuto(){

    Integer intHoraOcurrencia, intMinOcurrencia;

    intHoraOcurrencia = Integer.parseInt(
editTextHora.getText().toString());
    intMinOcurrencia = Integer.parseInt(
editTextMinuto.getText().toString());

    if (intHoraOcurrencia < 0 || intHoraOcurrencia > 24 ) {
        Toast.makeText(getApplicationContext(), "La Hora de Ocurrencia es
Invalida", Toast.LENGTH_SHORT).show();
        return false;
    } ;

    if (intMinOcurrencia < 0 || intMinOcurrencia > 60 ) {
        Toast.makeText(getApplicationContext(), "Los Minutos de
Ocurrencia es Invalido", Toast.LENGTH_SHORT).show();
        return false;
    } ;

    return true;
}
}

```

Placa _____

Fecha de Ocurrenc _____

Hor: _____

Min _____

- Mal Servicio del Conductor o Cobrador
- No Respeto la ruta o los paraderos
- Se paso la Luz Roja
- El Bus estaba en malas condiciones o sucio

Cuéntenos qué paso?

GRABAR

CERRAR

Anexo G

Código de Reporte de Incidencia de Pasajero

```
package com.example.sgcdtp;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class ReporteIncidenciasActivity extends AppCompatActivity {

    Button btnCerrar, btnBuscar;
    EditText editTextPlaca;
    private TextView txtCell;
    private TableRow tableRow;

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

        btnCerrar=(Button) findViewById(R.id.btnCerrar);
        btnBuscar=(Button) findViewById(R.id.btnBuscar);
        editTextPlaca = (EditText) findViewById(R.id.editTextPlaca);

        btnCerrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
        btnBuscar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                BuscarPlaca();
            }
        });
    }
}
```

```

public String Registros[] = new String[4] ;
public String cabeceras[] = {"Servicio","Ruta","Semáforos", "Limpieza"};

private void BuscarPlaca() {

    if (ValidarPlaca () ) {

        MainActivity ConexionSQL = new MainActivity();

        try {

            PreparedStatement pst=
ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Reporte_Incidencia] @numPlaca = '" +
editTextPlaca.getText().toString()+"'");
            ResultSet rs = pst.executeQuery();
            if (rs.next()) {
                Registros[0] = rs.getString(2);
                Registros[1] = rs.getString(3);
                Registros[2] = rs.getString(4);
                Registros[3] = rs.getString(5);

                PrepareTableLayout();
            }
            else {
                Toast.makeText(getApplicationContext(), "Placa no existe",
Toast.LENGTH_SHORT).show();
            }
        }
        catch (SQLException e) {

Toast.makeText(getApplicationContext(),e.getMessage(),Toast.LENGTH_SHORT).show
();
        }

    }

}

private boolean ValidarPlaca() {

    if (editTextPlaca.getText().toString().trim().equalsIgnoreCase("")) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar Placa",
Toast.LENGTH_SHORT).show();
        return false;
    }
    return true;
}

private void PrepareTableLayout( ){

    TableLayout tabla;

```

```

tabla = (TableLayout)findViewById(R.id.tableLayout);

if (tabla.getChildCount()>1) {
    tabla.removeAllViews();
}

TableRow FiladeCabecera = new TableRow(this);

for (int i=0; i < cabeceras.length; i++ ) {

    newCell();
    txtCell.setText(cabeceras[i]);
    txtCell.setBackgroundColor(Color.BLUE);
    txtCell.setTextColor(Color.WHITE);
    FiladeCabecera.addView(txtCell,newTableRowParams());

}
tabla.addView(FiladeCabecera);

TableRow FiladeDatos = new TableRow(this);

for (int f = 0; f < Registros.length; f++) {

    newCell();
    txtCell.setText(Registros[f]);
    txtCell.setBackgroundColor(Color.WHITE);
    if ( Integer.parseInt(Registros[f]) <=1) {

txtCell.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_bueno, 0, 0, 0);
    }
    else {

txtCell.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_malo, 0, 0, 0);
    }

    FiladeDatos.addView(txtCell, newTableRowParams());
}
tabla.addView(FiladeDatos,newTableRowParams());

}

private TableRow.LayoutParams newTableRowParams(){

    TableRow.LayoutParams params=new TableRow.LayoutParams();
    params.setMargins(1,1,1,1);
    params.weight=1;
    //params.width=225;
    return params;
};

private void newCell(){

    txtCell =new TextView(this);
    txtCell.setGravity(Gravity.CENTER);

```

```
txtCell1.setTextSize(20);  
}  
}
```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y DESEMPEÑO
DE LAS LÍNEAS TRANSPORTE PÚBLICO

Placa

BUSCAR

CERRAR

Anexo H

Código de Registrar Fiscalizador

```
package com.example.sgcdtp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RegistroFiscalizadorActivity extends AppCompatActivity {

    String TextNombre, TextPaterno, TextMaterno, TextDNI, TextUsuario,
    TextClave, TextCelular, TextCodigo;
    Button buttonCerrar, buttonGrabar;
    EditText editTextNombre, editTextPaterno, editTextMaterno, editTextDNI,
    editTextUsuario, editTextClave, editTextCelular;

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

        buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
        buttonGrabar=(Button) findViewById(R.id.buttonGrabar);

        editTextNombre = (EditText) findViewById(R.id.editTextNombre);
        editTextPaterno = (EditText) findViewById(R.id.editTextPaterno);
        editTextMaterno = (EditText) findViewById(R.id.editTextMaterno);
        editTextDNI = (EditText) findViewById(R.id.editTextDNI);
        editTextCelular = (EditText) findViewById(R.id.editTextCelular);
        editTextUsuario = (EditText) findViewById(R.id.editTextUsuario);
        editTextClave = (EditText) findViewById(R.id.editTextClave);

        buttonCerrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }
}
```

```

        buttonGrabar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                GrabarFiscalizador();

            }
        });
    }

    private void GrabarFiscalizador() {

        if (ValidarVariables () ) {

            MainActivity ConexionSQL = new MainActivity();

            try {
                PreparedStatement
                pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
                [dbo].[SP_SGCDTP_Fiscalizador] @Type = 'I', @Nombre = '" + TextNombre + "',
                @Apellido_Paterno = '" + TextPaterno + "', @Apellido_Materno = '" +
                TextMaterno + "', @DNI = '" + TextDNI + "', @celular = '" + TextCelular +
                "', @usuario = '" + TextUsuario + "', @Clave ='" + TextClave + "'");
                pst.executeUpdate();
                pst.close();
                Toast.makeText(getApplicationContext(), "Datos Grabados
                Correctamente", Toast.LENGTH_SHORT).show();
                finish();
            }
            catch (SQLException e) {

                Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show
                ();

            }

        }

    }

    private boolean ValidarVariables() {

        TextNombre = editTextNombre.getText().toString().trim();
        TextPaterno = editTextPaterno.getText().toString().trim();
        TextMaterno = editTextMaterno.getText().toString().trim();
        TextDNI = editTextDNI.getText().toString().trim();
        TextCelular = editTextCelular.getText().toString().trim();
        TextUsuario = editTextUsuario.getText().toString().trim();
        TextClave = editTextClave.getText().toString().trim();

        if (TextNombre.equalsIgnoreCase("") || TextNombre.length() < 3 ) {
            Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos
            con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (TextPaterno.equalsIgnoreCase("") || TextPaterno.length() < 3 ) {
            Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos

```

```

con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextMaterno.equalsIgnoreCase("") || TextMaterno.length() < 3 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextDNI.equalsIgnoreCase("") || TextDNI.length() < 3 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos
de 3 números", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextUsuario.equalsIgnoreCase("") || TextUsuario.length() < 3 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar su usuario
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextClave.equalsIgnoreCase("") || TextClave.length() < 3 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar su Clave
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextCelular.equalsIgnoreCase("") || TextCelular.length() != 9 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar Celular
Valido", Toast.LENGTH_SHORT).show();
        return false;
    }

    TextCelular = "+51" + TextCelular;
    return true;
}

};

```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y
DESEMPEÑO DE LAS LINEAS TRANSPORTE PÚBLICO
(DATOS DEL FISCALIZADOR)

Nombre _____

Paterno _____

Materno _____

DNI _____

Celular _____

usuario _____

Clave _____

GRABAR

CERRAR

Anexo I Código de Registrar Empresa

```
package com.example.sgcdtp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RegistroEmpresaActivity extends AppCompatActivity {

    String TextRuc, TextDescripcion, TextDireccion;
    Button buttonCerrar, buttonGrabar;
    EditText editTextRuc, editTextDescripcion, editTextDireccion;

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

        buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
        buttonGrabar=(Button) findViewById(R.id.buttonGrabar);

        editTextRuc = (EditText) findViewById(R.id.editTextRuc);
        editTextDescripcion =
        (EditText) findViewById(R.id.editTextDescripcion);
        editTextDireccion = (EditText) findViewById(R.id.editTextDireccion);

        buttonCerrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        buttonGrabar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                GrabarEmpresa();
            }
        });
    }
}
```

```

    });

}

private void GrabarEmpresa() {

    if (ValidarVariables () ) {

        MainActivity ConexionSQL = new MainActivity();

        try {
            PreparedStatement
pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_EmpTransporte] @Type = 'I', @idRuc = '" + TextRuc + "',
@Descripcion = '" + TextDescripcion + "', @Direccion = '" + TextDireccion +
"'");
            pst.executeUpdate();
            pst.close();
            Toast.makeText(getApplicationContext(), "Datos Grabados
Correctamente", Toast.LENGTH_SHORT).show();
            finish();
        }
        catch (SQLException e) {

Toast.makeText(getApplicationContext(),e.getMessage(),Toast.LENGTH_SHORT).show
();

        }

    }

}

private boolean ValidarVariables() {

    TextRuc = editTextRuc.getText().toString().trim();
    TextDescripcion = editTextDescripcion.getText().toString().trim();
    TextDireccion = editTextDireccion.getText().toString().trim();

    if (TextRuc.equalsIgnoreCase("") || TextRuc.length() != 11 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos
con mas de lma caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextDescripcion.equalsIgnoreCase("") || TextDescripcion.length()
< 3 ) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (TextDireccion.equalsIgnoreCase("") || TextDireccion.length() < 3
) {
        Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos

```

```
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
    return false;
}

return true;
}

};
```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y
DESEMPEÑO DE LAS LÍNEAS TRANSPORTE PÚBLICO -
REGISTRAR EMPRESA

Ruc

Descripcion

Direccion

GRABAR CERRAR

Anexo J Código de Registrar BUS

```
package com.example.sgcdtp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class RegistroBusActivity extends AppCompatActivity {

    String TextNumPlaca, TextNombreConductor, TextNombreCobrador, TextRuta,
    TextRucEmpresa ;
    Button buttonCerrar, buttonGrabar, buttonBuscar ;
    EditText editTextRucEmpresa, editTextEmpTransporte, editTextNumPlaca,
    editTextNombreConductor, editTextNombreCobrador,
    editTextNumAsientos, editTextRuta;
    int intIdEmpresa, intNumAsientos;

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

        buttonBuscar=(Button) findViewById(R.id.buttonBuscar);
        buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
        buttonGrabar=(Button) findViewById(R.id.buttonGrabar);

        buttonGrabar.setEnabled(false);

        editTextRucEmpresa= (EditText) findViewById(R.id.editTextRucEmpresa);
        editTextEmpTransporte=
        (EditText) findViewById(R.id.editTextEmpTransporte);
        editTextNumPlaca = (EditText) findViewById(R.id.editTextNumPlaca);
        editTextNombreConductor =
        (EditText) findViewById(R.id.editTextConductor);
        editTextNombreCobrador =
        (EditText) findViewById(R.id.editTextCobrador);
        editTextNumAsientos =
        (EditText) findViewById(R.id.editTextNumAsientos);
        editTextRuta = (EditText) findViewById(R.id.editTextRuta);

        editTextNumPlaca.setEnabled(false);
        editTextNombreConductor.setEnabled(false);
        editTextNombreCobrador.setEnabled(false);
    }
}
```

```

editTextNumAsientos.setEnabled(false);
editTextRuta.setEnabled(false);

buttonCerrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

buttonBuscar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        BuscarEmpresa();
    }
});

buttonGrabar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        GrabarBus();
    }
});
}

private void BuscarEmpresa() {
    TextRucEmpresa = editTextRucEmpresa.getText().toString();

    if (ValidarRucEmpresa () ) {
        MainActivity ConexionSQL = new MainActivity();

        try {
            PreparedStatement
pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_EmpTransporte] @Type = 'S', @idRuc = '" + TextRucEmpresa
+ "'" );

            ResultSet rs = pst.executeQuery();

            if (rs.next()) {

                Toast.makeText(getApplicationContext(), "Encontro
Registro", Toast.LENGTH_SHORT).show();

                intIdEmpresa = rs.getInt(1);
                ediTextEmpTransporte.setText(rs.getString(2));

```

```

        editTextRucEmpresa.setEnabled(false);
        editTextEmpTransporte.setEnabled(false);
        buttonBuscar.setEnabled(false);

        editTextNumPlaca.setEnabled(true);
        editTextNombreConductor.setEnabled(true);
        editTextNombreCobrador.setEnabled(true);
        editTextNumAsientos.setEnabled(true);
        editTextRuta.setEnabled(true);

        buttonGrabar.setEnabled(true);
    }
}
catch (SQLException e) {
Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show
();
}
}
}

private void GrabarBus() {
    if (ValidarVariables () ) {
        MainActivity ConexionSQL = new MainActivity();

        try {
            PreparedStatement
pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Buses] @Type = 'I', @NumPlaca = '" + TextNumPlaca + "',
@idEmpresa = '" + intIdEmpresa + "', @NombreConductor = '" +
TextNombreConductor + "', @NombreCobrador = '" + TextNombreCobrador + "',
@NumAsientos = '" + intNumAsientos + "', @Ruta = '" + TextRuta + "'"); ;
            pst.executeUpdate();
            pst.close();
            Toast.makeText(getApplicationContext(), "Datos Grabados
Correctamente", Toast.LENGTH_SHORT).show();
            finish();
        }
        catch (SQLException e) {
Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show
();
}
}
}

private boolean ValidarRucEmpresa() {

```

```

        if (TextRucEmpresa.length() != 11 ) {
            Toast.makeText(getApplicationContext(), "RUC no es valido",
Toast.LENGTH_SHORT).show();
            return false;
        }
        return true;
    }

    private boolean ValidarVariables() {

        TextNumPlaca = editTextNumPlaca.getText().toString().trim();
        TextNombreConductor =
editTextNombreConductor.getText().toString().trim();
        TextNombreCobrador =
editTextNombreCobrador.getText().toString().trim();
        intNumAsientos =
Integer.parseInt(editTextNumAsientos.getText().toString());
        TextRuta = editTextRuta.getText().toString().trim();

        if (TextNumPlaca.equalsIgnoreCase("") || TextNumPlaca.length() < 3 )
        {
            Toast.makeText(getApplicationContext(), "Debe Ingresar datos de
Placa con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (TextNombreConductor.equalsIgnoreCase("") ||
TextNombreConductor.length() < 3 ) {
            Toast.makeText(getApplicationContext(), "Debe Ingresar datos del
Conductor con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (TextNombreCobrador.equalsIgnoreCase("") ||
TextNombreCobrador.length() < 3 ) {
            Toast.makeText(getApplicationContext(), "Debe Ingresar datos del
Cobrador con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (intNumAsientos < 5 ) {
            Toast.makeText(getApplicationContext(), "Debe tener mas de 5 de
asientos", Toast.LENGTH_SHORT).show();
            return false;
        }

        if (TextRuta.equalsIgnoreCase("") || TextRuta.length() < 3 ) {
            Toast.makeText(getApplicationContext(), "Debe Ingresar su ruta con
mas de 3 caracteres", Toast.LENGTH_SHORT).show();
            return false;
        }

        return true;
    }

```

}

};

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y
DESEMPEÑO DE LAS LÍNEAS TRANSPORTE PÚBLICO -
REGISTRAR BUSES

Ruc de la Empresa **BUSCAR**

Placa Asientos

Conductor

Cobrador

Ruta

GRABAR **CERRAR**

Anexo K Código de Registrar Problema

```
package com.example.sgcdtp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RegistroProblemaActivity extends AppCompatActivity {

    String    TextProblema;
    Button    buttonCerrar, buttonGrabar;
    EditText  editTextProblema;

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

        buttonCerrar=(Button) findViewById(R.id.buttonCerrar);
        buttonGrabar=(Button) findViewById(R.id.buttonGrabar);

        editTextProblema = (EditText) findViewById(R.id.editTextProblema);

        buttonCerrar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        buttonGrabar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                GrabarProblema();
            }
        });
    }
}
```

```

private void GrabarProblema() {
    if (ValidarVariables () ) {
        MainActivity ConexionSQL = new MainActivity();

        try {
            PreparedStatement
pst=ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Problemas] @Type = 'I', @Descripcion = '" + TextProblema +
"'" ) ;
            pst.executeUpdate();
            pst.close();
            Toast.makeText(getApplicationContext(), "Datos Grabados
Correctamente", Toast.LENGTH_SHORT).show();
            finish();
        }
        catch (SQLException e) {
Toast.makeText(getApplicationContext(),e.getMessage(),Toast.LENGTH_SHORT).show
();
        }
    }
}
private boolean ValidarVariables() {
    TextProblema = editTextProblema.getText().toString().trim();

    if (TextProblema.equalsIgnoreCase("") || TextProblema.length() < 3 )
{
        Toast.makeText(getApplicationContext(), "Debe Ingresar sus Datos
con mas de 3 caracteres", Toast.LENGTH_SHORT).show();
        return false;
    }

    return true;
}
};

```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y
DESEMPEÑO DE LAS LÍNEAS TRANSPORTE PÚBLICO -
REGISTRAR EMPRESA

Ruc

Descripcion

Direccion

GRABAR

CERRAR

Anexo L Código de Reporte de Fiscalizador

```
package com.example.sgcdtp;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class ReporteFiscalizador extends AppCompatActivity {

    Button buttonBuscar, buttonCerrar;
    EditText editTextRuc, editTextEmpresa;
    TextView TextViewEmpresa;
    String strTextRuc, strEmpresa;

    String Registros[] = new String[2] ;
    String cabeceras[] = {"Problema", "#Casos"};
    Integer CabeceraTextTamano[] = {20,20};
    Integer FilaTextTamano[]= {15,15};

    TableLayout tabla;

    Integer intIdEmpresa;

    TextView txtCell;
    TableRow tableRow;

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

        buttonBuscar = (Button) findViewById(R.id.buttonBuscar);
        buttonCerrar = (Button) findViewById(R.id.buttonCerrar);
        editTextRuc = (EditText) findViewById(R.id.editTextRuc);

        TextViewEmpresa= (TextView) findViewById(R.id.textView9);
        tabla = (TableLayout) findViewById(R.id.tableLayout2);
```

```

buttonCerrar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

buttonBuscar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        BuscarEmpresa();
    }
});

}

private void BuscarEmpresa() {

    strTextRuc = editTextRuc.getText().toString();

    if (ValidarRucEmpresa()) {

        MainActivity ConexionSQL = new MainActivity();

        try {

            PreparedStatement pst =
ConexionSQL.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_EmpTransporte] @Type = 'S', @idRuc = '" + strTextRuc + "'");
            ResultSet rs = pst.executeQuery();

            if (rs.next()) {
                Toast.makeText(getApplicationContext(), "Encontro
Registro", Toast.LENGTH_SHORT).show();
                intIdEmpresa = rs.getInt(1);
                TextViewEmpresa.setText(rs.getString(2));
                Reporte();
            }
        } catch (SQLException e) {
            Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_SHORT).show();
        }

    }

}

private boolean ValidarRucEmpresa () {

    if (strTextRuc.length() != 11) {
        Toast.makeText(getApplicationContext(), "RUC no es valido",
Toast.LENGTH_SHORT).show();
        return false;
    }
}

```

```

    }
    return true;
}

private void Reporte() {
    MainActivity ConexionSQL2 = new MainActivity();

    try {
        PreparedStatement pst2 =
ConexionSQL2.conexionDBSQL().prepareStatement(" EXEC
[dbo].[SP_SGCDTP_Reporte_Fiscalizar_Incidencias] @idEmpresa = '" +
intIdEmpresa + "'");
        ResultSet rs2 = pst2.executeQuery();

        Toast.makeText(getApplicationContext(), "Encontro Registro",
Toast.LENGTH_SHORT).show();

        AddCabecera();

        while (rs2.next()) {

            Registros[0] = rs2.getString(1);
            Registros[1] = rs2.getString(2);
            AddFila();
        }

    } catch (SQLException e) {
        Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_SHORT).show();
    }

}

private void AddCabecera() {

    if (tabla.getChildCount()>1) {
        tabla.removeAllViews();
        tabla.setColumnStretchable (1, true);
        tabla.setColumnStretchable(2, true);
    }

    TableRow FiladeCabecera = new TableRow(this);

    for (int i=0; i < cabeceras.length; i++ ) {

        txtCell =new TextView(this);

        txtCell.setGravity(Gravity.CENTER);
        txtCell.setTextSize(CabeceraTextTamano[i]);
    }
}

```

```

        txtCell.setBackgroundColor(Color.BLUE);
        txtCell.setTextColor(Color.WHITE);
        txtCell.setText(cabeceras[i]);

        FiladeCabecera.addView(txtCell, newTableRowParams());
    }
    tabla.addView(FiladeCabecera);
}

private void AddFila() {
    TableRow FiladeDatos = new TableRow(this);

    for (int f = 0; f < Registros.length; f++) {

        txtCell = new TextView(this);

        if (f==0){txtCell.setGravity(Gravity.LEFT);}
        else {txtCell.setGravity(Gravity.RIGHT); }
        txtCell.setTextSize(FilaTextTamano[f]);
        txtCell.setBackgroundColor(Color.WHITE);
        txtCell.setTextColor(Color.BLACK);
        txtCell.setText(Registros[f]);

        FiladeDatos.addView(txtCell, newTableRowParams());
    }
    tabla.addView(FiladeDatos, newTableRowParams());
}

private TableRow.LayoutParams newTableRowParams() {
    TableRow.LayoutParams params=new TableRow.LayoutParams();
    params.setMargins(1,1,1,1);
    params.weight=1;
    return params;
};
}

```

SISTEMA DE GESTIÓN DE CUMPLIMIENTO Y DESEMPEÑO
DE LAS LINEAS TRANSPORTE PÚBLICO - REPORTE
FISCALIZADOR

RUC

ConstraintLayout

Anexo M

Procedimiento de Registro de Pasajero

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER Procedure [dbo].[SP_SGCDTP_Pasajero]

    @Type char(1) = 'S'
    ,@Nombre varchar(20) = ''
    ,@Apellido_Paterno varchar(20) = ''
    ,@Apellido_Materno varchar(20) = ''
    ,@DNI varchar(20) = ''
    ,@celular varchar (20) = ''
    ,@usuario char (10) = ''
    ,@clave varchar (20) = ''

as

if upper(@Type) = 'I'
begin

    if Exists (select * from [dbo].[Pasajero] where celular = @celular)

begin
    RAISERROR ('El Celular ingresado ya existe', 11,1)
    return
end

Begin Try

    Begin Transaction "Ins_Trans_Pasajero"

    INSERT INTO [dbo].[Pasajero]
        ([Nombre]
        , [Apellido_Paterno]
        , [Apellido_Materno]
        , [DNI]
        , [celular]
        , [usuario]
        , [Clave])
    VALUES
        (@Nombre
        , @Apellido_Paterno
        , @Apellido_Materno
        , @DNI
        , @celular
```

```

        ,@usuario
        ,@clave)

    Commit Transaction "Ins_Tran_Pasajero"

end try

Begin Catch

    ROLLBACK TRANSACTION

    declare @ErrorMessage nvarchar(4000);
    declare @ErrorSeverity INT;
        declare @ErrorState INT;

    select
        @ErrorMessage = 'Ocurrio un error: ' + ERROR_MESSAGE(),
        @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE();

    RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)

End Catch

End

if upper(@Type) = 'C'
Begin

    SELECT
        [Clave]
    FROM [dbo].[Pasajero]

    Where

        [usuario] = @usuario
        and [celular] = @celular

End

if upper(@Type) = 'S'
Begin

    SELECT
        [IdPasajero]
        , [Nombre]
        , [Apellido_Paterno]
        , [Apellido_Materno]
        , [DNI]
        , [usuario]
        , [celular]

```

```

    ,[Clave]
FROM [dbo].[Pasajero]
Where

    ([usuario] = iif(@usuario='', [usuario],@usuario) and [Clave] =
iif(@clave='', [Clave],@clave)
and [celular] = iif(@celular='', [celular],@celular))

End

if upper(@Type) = 'D'
Begin

    Begin Transaction "Del_Trans_Pasajero"

    delete
    FROM [dbo].[Pasajero]
    Where [celular] = @celular

    Commit Transaction "Del_Trans_Pasajero"

End

```

Anexo N Procedimiento de Registro de Fiscalizador

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER Procedure [dbo].[SP_SGCDTP_Fiscalizador]

    @Type char(1) = 'S'
    ,@Nombre varchar(20) = ''
    ,@Apellido_Paterno varchar(20) = ''
    ,@Apellido_Materno varchar(20) = ''
    ,@DNI varchar(20) = ''
    ,@celular varchar (20) = ''
    ,@usuario char (10) = ''
    ,@clave varchar (20) = ''

as

if upper(@Type) = 'I'

begin

    if Exists (select * from [dbo].[Fiscalizador] where celular = @celular)

begin
    RAISERROR ('El Celular ingresado ya existe', 11,1)
    return
end

Begin Try

    Begin Transaction "Ins_Trans_Fiscalizador"

    INSERT INTO [dbo].[Fiscalizador]
        ([Nombre]
        , [Apellido_Paterno]
        , [Apellido_Materno]
        , [DNI]
        , [celular]
        , [usuario]
        , [Clave])
    VALUES
        (@Nombre
```

```

        ,@Apellido_Paterno
        ,@Apellido_Materno
        ,@DNI
        ,@celular
        ,@usuario
        ,@clave)

    Commit Transaction "Ins_Tran_Fiscalizador"

end try

Begin Catch

    ROLLBACK TRANSACTION

    declare @ErrorMessage nvarchar(4000);
    declare @ErrorSeverity INT;
    declare @ErrorState INT;

    select
        @ErrorMessage = 'Ocurrio un error: ' + ERROR_MESSAGE(),
        @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE();

    RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)

End Catch

End

if upper(@Type) = 'S'
Begin

    SELECT
        [IdFiscalizador]
        , [Nombre]
        , [Apellido_Paterno]
        , [Apellido_Materno]
        , [DNI]
        , [usuario]
        , [celular]
        , [Clave]
    FROM [dbo].[Fiscalizador]
    Where

        ([usuario] = iif(@usuario='', [usuario],@usuario) and [Clave] =
iif(@clave='', [Clave],@clave)
        and [celular] = iif(@celular='', [celular],@celular))

End

```

```
if upper(@Type) = 'D'  
Begin  
  
    Begin Transaction "Del-Tran-Fiscalizador"  
  
    delete  
        FROM [dbo].[Fiscalizador]  
        Where [celular] = @celular  
  
    Commit Transaction "Del-Tran-Fiscalizador"  
  
End
```

Anexo O Procedimiento de Registro de Incidencias

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER Procedure [dbo].[SP_SGCDTP_Incidencias]

    @Type char(1) = 'I'
    ,@IdPasajero smallint = 0
    ,@numplaca varchar(20)
    ,@idProblema1 smallint = 0
    ,@idProblema2 smallint = 0
    ,@idProblema3 smallint = 0
    ,@idProblema4 smallint = 0
    ,@FecOcurrencia varchar(10)
    ,@HoraMinOcurrencia varchar(6) = ''
    ,@Comentario varchar(1000) = ''

as

set dateformat dmy

if upper (@Type)= 'I'

Begin

    if not Exists (select * from [dbo].[Buses] where numPlaca = @numPlaca)

        begin
            RAISERROR ('La placa ingresada no existe', 11,1)
            return
        end

    declare @AxidProblema smallint = 0

    Begin Try

        Begin Transaction

            while (@idProblema1 = 1 or @idProblema2 = 1 or @idProblema3 = 1 or
@idProblema4 = 1)

                begin

                    if @idProblema1 =1
```

```

begin
    set @AxidProblema = 1
    set @idProblema1 = 0
end
else
begin
    if @idProblema2 =1
    begin
        set @AxidProblema = 2
        set @idProblema2 = 0
    end
else
begin
    if @idProblema3 =1
    begin
        set @AxidProblema = 3
        set @idProblema3 = 0
    end
else
begin
    if @idProblema4 =1
    begin
        set @AxidProblema = 4
        set @idProblema4 = 0
    end
end
end
end
end

```

```

INSERT INTO [dbo].[Incidencias]

```

```

    ([IdPasajero]
    , [numPlaca]
    , [idProblema]
    , [FecOcurrencia]
    , [HoraMinOcurrencia]
    , [FecRegistro]
    , [Comentario])

```

```

VALUES

```

```

    (@IdPasajero
    , @numPlaca
    , @AxidProblema
    , cast(@FecOcurrencia as date)
    , @HoraMinOcurrencia
    , getdate()
    , @Comentario
    )

```

```

end
Commit Transaction

```

```

end try

```

Begin Catch

```
ROLLBACK TRANSACTION
```

```
declare @ErrorMessage nvarchar(4000);
```

```
declare @ErrorSeverity INT;
```

```
declare @ErrorState INT;
```

```
select
```

```
  @ErrorMessage = ERROR_MESSAGE(),
```

```
  @ErrorSeverity = ERROR_SEVERITY(),
```

```
  @ErrorState = ERROR_STATE();
```

```
RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)
```

End Catch

End

Anexo P

Procedimiento de Reporte de Incidencia

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER procedure [dbo].[SP_SGCDTP_Reporte_Incidencia]

    @numPlaca varchar(20)

as

Select *

from (
select
    a.numPlaca
    ,a.idProblema
    ,d.Descripcion

from [dbo].[Incidencias] a

        inner join [dbo].[Problema] d on

            a.idProblema = d.idProblema

        Where a.numPlaca = @numPlaca)

S

pivot (count(idProblema)
For [Descripcion] in ([Mal Servicio del Conductor o Cobrador] ,[No Respeto los
paraderos],[Se paso la luz Roja],[El bus estaba en malas condiciones o sucio]) ) p
```

Anexo Q Procedimiento de Registro de Empresas

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[SP_SGCDTP_EmpTransporte]

    @type char(1) = 'S'
    ,@idEmpresa int = 0
    ,@idRuc varchar(20) = ''
    ,@Descripcion varchar(500) = ''
    ,@Direccion varchar(200) = ''

as

if @type = 'I'

Begin

    Begin Try

        Begin Transaction "Ins_Tran_Buses"

            set @idEmpresa = (select max(isnull(idEmpresa,0)) + 1 from dbo.EmpTransporte)

            INSERT INTO [dbo].[EmpTransporte]
                ([idEmpresa]
                , [idRuc]
                , [Descripcion]
                , [Direccion])

            VALUES
                (@idEmpresa
                , @idRuc
                , @Descripcion
                , @Direccion)

            commit Transaction "Ins_Tran_Fiscalizador"

        end try

        Begin Catch

            ROLLBACK TRANSACTION

            declare @ErrorMessage nvarchar(4000);
            declare @ErrorSeverity INT;
            declare @ErrorState INT;
```

```

select
    @ErrorMessage = 'Ocurrio un error: ' + ERROR_MESSAGE(),
    @ErrorSeverity = ERROR_SEVERITY(),
    @ErrorState = ERROR_STATE();

    RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)

End Catch
end

if @type = 'S'

Begin
select a.idEmpresa
    ,a.Descripcion
    ,a.Direccion
    ,a.idRuc
from dbo.EmpTransporte a where idRuc = @idRuc

end

```

Anexo R

Procedimiento de Registro de Buses

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[SP_SGCDTP_Buses]

    @type char(1) = 'S'
    ,@numPlaca varchar(20)
    ,@idEmpresa int
    ,@NombreConductor varchar(30)
    ,@NombreCobrador varchar(30)
    ,@NumAsientos smallint
    ,@Ruta varchar(150)

as

if @type = 'I'

Begin

    Begin Try

        Begin Transaction "Ins_Tran_Buses"

            INSERT INTO [dbo].[Buses]
                ([numPlaca]
                , [idEmpresa]
                , [NombreConductor]
                , [NombreCobrador]
                , [NumAsientos]
                , [Ruta])
            VALUES
                (@numPlaca
                , @idEmpresa
                , @NombreConductor
                , @NombreCobrador
                , @NumAsientos
                , @Ruta)

            Commit Transaction "Ins_Tran_Buses"

        end try

        Begin Catch
```

```
ROLLBACK TRANSACTION

declare @ErrorMessage nvarchar(4000);
declare @ErrorSeverity INT;
    declare @ErrorState INT;

select
    @ErrorMessage = 'Ocurrio un error: ' + ERROR_MESSAGE(),
    @ErrorSeverity = ERROR_SEVERITY(),
    @ErrorState = ERROR_STATE();

    RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)

End Catch

end
```

Anexo S Procedimiento de Registro de Problemas

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[SP_SGCDTP_Problemas]

    @type char(1) = 'S'
    ,@idProblema smallint = 0
    ,@Descripcion char(100)
as

if @type = 'I'

Begin

    Begin Try

        Begin Transaction "Ins_Tran_Problemas"

        set @idProblema = (select max(isnull(idProblema,0)) + 1 from dbo.Problema)

        INSERT INTO [dbo].[Problema]
            ([idProblema]
            ,[Descripcion])
        VALUES
            (@idProblema
            ,@Descripcion)

            Commit Transaction "Ins_Tran_Problemas"

        end try

        Begin Catch

            ROLLBACK TRANSACTION

            declare @ErrorMessage nvarchar(4000);
            declare @ErrorSeverity INT;
            declare @ErrorState INT;

            select
                @ErrorMessage = 'Ocurrio un error: ' + ERROR_MESSAGE(),
                @ErrorSeverity = ERROR_SEVERITY(),
                @ErrorState = ERROR_STATE();
            RAISERROR (@ErrorMessage, @ErrorSeverity,@ErrorState)

        End Catch

    end
```

Anexo T Procedimiento de Reporte del Fiscalizador

```
USE [SGCDTP]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER Procedure [dbo].[SP_SGCDTP_Reporte_Fiscalizar_Incidencias]
    @IdEmpresa as int = 0
    ,@FechaOcurranciaInicial as varchar(10) = ''
    ,@FechaOcurranciaFinal as varchar (10) = ''

as

select
    trim(c.Descripcion) as Descripcion
    ,count(*) as Casos
from [dbo].[Incidencias]
    a inner join dbo.Buses b on
        a.numplaca = b.numPlaca

        inner join dbo.EmpTransporte d on
            b.idEmpresa = d.idEmpresa

            inner join dbo.Problema c on
                a.idProblema = c.idProblema

Where (a.FecOcurrancia >= case @FechaOcurranciaInicial when '' then a.FecOcurrancia else
convert(datetime, @FechaOcurranciaInicial,103) end)
and (a.FecOcurrancia <= case @FechaOcurranciaFinal when '' then a.FecOcurrancia else
convert(datetime, @FechaOcurranciaFinal,103) end )
and (b.idEmpresa = case @IdEmpresa when 0 then b.idEmpresa else @IdEmpresa end)

group by
    c.Descripcion
```

CONFORMIDADES

DOCUMENTO DE CONFORMIDAD

Yo, Javier Alejandro SANTISTEBAN ROMERO, con documento de identidad DNI N° 16766113 ingeniero titulado y colegiatura número: 87394 me dirijo a ustedes Sres. Universidad Continental, E.A.P. de Ingeniería De Sistemas E Informática, a fin de hacer constar mi conformidad como revisor de los módulos de registro de información, visualización de información, uso y revisión de caja negra durante el período de desarrollo del aplicativo móvil del "Sistema de gestión de Cumplimiento y Desempeño del Transporte Público", que el Bach. Eduardo Ydrogo Ramírez tuvo a bien presentarme con fecha 26/06/2020.



Ingeniero Revisor

DOCUMENTO DE CONFORMIDAD

Yo, CARLOS MIGUEL DELGADO REQUE identificado con documento de identidad Nro: 09873244 Ingeniero Titulado y Colegiatura CIP:67305 me dirijo a ustedes Sres. Universidad Continental, E.A.P. de Ingeniería De Sistemas E Informática, a fin de hacer constar mi conformidad como revisor de los módulos de registro de información, visualización de información, uso y revisión de caja negra durante el período de desarrollo del aplicativo móvil del "Sistema de gestión de Cumplimiento y Desempeño del Transporte Público", que el Bach. Eduardo Ydrogo Ramírez tuvo a bien presentarme con fecha 26/06/2020.



Ingeniero Revisor