



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E  
INFORMÁTICOS**

**Tema:**

---

APLICATIVO MÓVIL DE GEO-UBICACIÓN EN TIEMPO REAL, PARA EL  
TRANSPORTE PRIVADO DE PLASTICAUCHO INDUSTRIAL S.A DE LA  
CIUDAD DE AMBATO

---

**Trabajo de Titulación Modalidad:** Proyecto de Investigación, presentado previo la  
obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

**LINEA DE INVESTIGACIÓN:** Desarrollo de Software

**AUTOR:** Franklin Vicente Lesano Pérez

**TUTOR:** Ing. Mg. Carlos Núñez

Ambato – Ecuador

Septiembre – 2022

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: APLICATIVO MÓVIL DE GEO-UBICACIÓN EN TIEMPO REAL, PARA EL TRANSPORTE PRIVADO DE PLASTICAUCHO INDUSTRIAL S.A DE LA CIUDAD DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Franklin Vicente Lesano Pérez estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022.

-----  
Ing. Mg. Carlos Núñez  
TUTOR

## **AUTORÍA**

El presente Proyecto de Investigación titulado: APLICATIVO MÓVIL DE GEO-UBICACIÓN EN TIEMPO REAL, PARA EL TRANSPORTE PRIVADO DE PLASTICAUCHO INDUSTRIAL S.A DE LA CIUDAD DE AMBATO es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022.

-----  
Franklin Vicente Lesano Pérez

C.C. 1804220265

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Lesano Pérez Franklin Vicente, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICATIVO MÓVIL DE GEO-UBICACIÓN EN TIEMPO REAL, PARA EL TRANSPORTE PRIVADO DE PLASTICAUCHO INDUSTRIAL S.A DE LA CIUDAD DE AMBATO**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, septiembre 2022.

-----

Ing. Pilar Urrutia, Mg.

**PRESIDENTE DEL TRIBUNAL**

-----

Ing. Franklin Mayorga

**PROFESOR CALIFICADOR**

-----

Ing. David Guevara

**PROFESOR CALIFICADOR**

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022.

-----  
Franklin Vicente Lesano Pérez

C.C. 1804220265

AUTOR

## ÍNDICE GENERAL DE CONTENIDO

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO.....	iv
DERECHOS DE AUTOR.....	v
Índice de Tablas .....	xi
Índice de Figuras .....	xvi
RESUMEN EJECUTIVO .....	xviii
ABSTRACT .....	xix
CAPITULO I.- MARCO TEÓRICO .....	1
1.1. Tema de Investigación.....	1
1.2. Antecedentes Investigativos .....	1
1.2.1. Contextualización del Problema .....	1
1.2.2. Fundamentación Teórica.....	2
1.2.2.1. Geolocalización.....	3
1.2.2.1.1. Ventajas de la Geolocalización .....	3
1.2.2.1.2. GPS .....	3
1.2.2.2. Geo-vallas o Geo-cercas .....	5
1.2.2.2.1. Ventajas o Beneficios de las Geo-cercas. ....	6
1.2.2.2.2. Principio de funcionamiento de las Geo-cercas.....	6
1.2.2.2.3. Tipos de Geo-cercas.....	6
1.2.2.3. Visual Estudio .....	7
1.2.2.3.1. Ediciones .....	8
1.2.2.3.2. Características de productividad populares.....	8
1.2.2.3.3. Licencias de Visual Estudio .....	9
1.2.2.4. PostgreSQL .....	9

1.2.2.4.1. ¿Quién utiliza PostgreSQL?.....	10
1.2.2.4.2. ¿Por qué usar PostgreSQL?.....	10
1.2.2.5. ASP.NET MVC.....	12
1.2.2.5.1. Patrón MVC .....	13
1.2.2.6. Xamarin Forms.....	15
1.2.2.6.1. A quién va destinado Xamarin.Forms.....	15
1.2.2.6.2. Cómo funciona Xamarin.Forms.....	16
1.2.2.6.3. Xamarin.Essentials.....	17
1.2.2.6.4. Características específicas de las plataformas.....	17
1.2.2.7. Manifiesto Ágil .....	17
1.2.2.7.1. Metodología Ágil Scrum.....	17
1.2.2.7.2. Roles de Scrum .....	18
1.2.2.7.3. Fases de Scrum.....	18
Preparación del Proyecto.....	18
Planificación del Sprint.....	19
Etapas de Desarrollo .....	19
Ventajas y Desventajas de Scrum .....	20
1.2.2.7.4. Metodología Ágil Kanban.....	20
Reglas de Kanban.....	21
Ventajas y Desventajas .....	22
1.2.2.7.5. Metodología Ágil XP .....	23
1.2.2.8. Fases XP.....	23
Fase de Exploración.....	23
Fase de Planificación.....	23
Fase de Iteraciones.....	24
Fase de Producción.....	24
Fase de Mantenimiento.....	24

1.2.2.9. Roles de XP.....	24
Ventajas y Desventajas .....	25
1.3. Objetivos.....	25
1.3.2. Objetivo General.....	25
1.3.3. Objetivos Específicos .....	25
<b>CAPITULO II.- METODOLOGÍA .....</b>	<b>26</b>
2.1. Materiales.....	26
2.2. Métodos.....	29
2.2.1. Modalidad de la Investigación .....	29
2.2.1.1. Nivel de investigación.....	29
2.2.2. Población y Muestra.....	29
2.2.3. Recolección de Información .....	30
2.2.3.1. Resultados de Encuesta.....	33
2.2.4. Procesamiento y Análisis de Datos .....	38
<b>CAPITULO III.- RESULTADOS Y DISCUSIÓN .....</b>	<b>41</b>
3.1. Análisis y discusión de resultados.....	41
3.1.1. Análisis de los datos para el seguimiento en tiempo real de los recorridos de buses privados de Plasticaucho industrial. ....	41
3.1.2. Análisis del proceso de seguimiento GPS del dispositivo móvil.....	44
3.1.3. Análisis Comparativo de la Metodología Agiles. ....	46
3.1.3.1. Metodología seleccionada.....	49
3.1.4 Gestores de Base de Datos .....	49
3.1.4.1. PostgreSQL .....	49
3.1.4.1.1. Características .....	49
3.1.4.2. MySQL.....	49
3.1.4.2.1 Características .....	49
3.1.5. Análisis Comparativo de los Gestores de Base de Datos.....	50



3.1.5.1. Gestor de Base de Datos seleccionada .....	53
3.2. Desarrollo de la Propuesta .....	53
3.2.1. Fase 1: Panificación .....	53
3.2.1.1. Levantamiento de la información.....	53
3.2.1.2. Definición de roles .....	53
3.2.1.3. Historias de usuario.....	54
3.2.1.4. Tareas .....	62
3.3.1.5. Valoración de Historias de Usuario .....	73
3.3.1.6. Estimación de las Historias de Usuario.....	73
3.3.1.7. Plan de Entrega .....	77
3.3.2. Fase 2: Diseño .....	78
3.3.2.1. Metáfora .....	78
3.3.2.2. Tarjetas CRC.....	81
3.3.2.3. Estructura del proyecto .....	83
Hardware .....	83
Software .....	83
Arquitectura del proyecto.....	84
3.3.2.4. Diseño de la base de datos .....	84
3.3.2.5. Diseño de Interfaces Aplicación Móvil.....	86
Diseño de Loguin Aplicativo Conductor .....	86
Diseño Ingreso Datos de la Ruta.....	86
Diseño Pantalla de Recorrido.....	87
Diseño Pantalla de Clientes.....	88
3.3.2.6. Diseño de Interfaces Gestor Web.....	88
Diseño de Pantalla Conductores.....	88
Diseño de Pantalla Conductores.....	89
3.3.3. Fase 3 Codificación.....	92

3.3.3.1. Bases de datos .....	92
3.3.3.2. Webs APIs.....	96
3.3.3.2.1. Conexión a la base de datos .....	96
3.3.3.2.2. Creación APIS REST Conductores.....	96
3.3.3.2.3. Creación APIS REST Bus.....	98
3.3.3.2.4. Creación APIS REST Línea.....	99
3.3.3.2.5. Creación APIS REST Parada.....	101
3.3.3.2.6. Creación APIS REST Ruta .....	102
3.3.3.2.7. Creación APIS REST Recorrido.....	103
3.3.3.2.8. Creación APIS REST Notificaciones.....	103
3.3.3.3. Aplicación móvil.....	104
3.3.3.3.1. Construcción del Servicio para consumo de las APIs.....	104
3.3.3.3.2. Construcción del Servicio de Geocercas.....	108
3.3.3.3.3. Desarrollo Back y Frond de los aplicativos .....	112
3.3.3.4. Gestor Web.....	118
3.3.4. Fase 3 Pruebas.....	146
3.3.4.1. Configuración del entorno de pruebas .....	146
3.3.4.2. Pruebas de aceptación .....	151
<b>CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>159</b>
4.1. Conclusiones .....	159
4.2. Recomendaciones.....	160
Manual de usuario.....	166

## Índice de Tablas

Tabla N.- 1 Población y muestra.....	29
Tabla N.- 2 Resultados de la entrevista.....	32
Tabla N.- 3 Pregunta 1 Encuesta.....	33
Tabla N.- 4 Pregunta 2 Encuesta.....	34
Tabla N.- 5 Pregunta 3 Encuesta.....	35
Tabla N.- 6 Pregunta 4 Encuesta.....	36
Tabla N.- 7 Pregunta 5 Encuesta.....	37
Tabla N.- 8 Análisis Comparativo de la metodología ágiles. ....	48
Tabla N.- 9 Tabla de Comparación de los gestores de base de datos .....	52
Tabla N.- 10 Definición de roles.....	54
Tabla N.- 11 Historias de usuario 001 .....	54
Tabla N.- 12 Historias de usuario 002 .....	55
Tabla N.- 13 Historias de usuario 003 .....	55
Tabla N.- 14 Historias de usuario 004 .....	55
Tabla N.- 15 Historias de usuario 005 .....	56
Tabla N.- 16 Historias de usuario 006 .....	56
Tabla N.- 17 Historias de usuario 007 .....	57
Tabla N.- 18 Historias de usuario 008 .....	57
Tabla N.- 19 Historias de usuario 009 .....	57
Tabla N.- 20 Historias de usuario 010 .....	58

Tabla N.- 21 Historias de usuario 011 .....	58
Tabla N.- 22 Historias de usuario 012 .....	59
Tabla N.- 23 Historias de usuario 013 .....	59
Tabla N.- 24 Historias de usuario 014 .....	60
Tabla N.- 25 Historias de usuario 015 .....	60
Tabla N.- 26 Historias de usuario 016 .....	60
Tabla N.- 27 Historias de usuario 017 .....	61
Tabla N.- 28 Historias de usuario 018 .....	61
Tabla N.- 29 Historias de usuario 019 .....	62
Tabla N.- 30 Historias de usuario 020 .....	62
Tabla N.- 31 Tarea 001 .....	62
Tabla N.- 32 Tarea 002 .....	63
Tabla N.- 33 Tarea 003 .....	63
Tabla N.- 34 Tarea 004 .....	63
Tabla N.- 35 Tarea 005 .....	63
Tabla N.- 36 Tarea 006 .....	64
Tabla N.- 37 Tarea 007 .....	64
Tabla N.- 38 Tarea 008 .....	64
Tabla N.- 39 Tarea 009 .....	64
Tabla N.- 40 Tarea 010 .....	65
Tabla N.- 41 Tarea 011 .....	65

Tabla N.- 42 Tarea 012 .....	65
Tabla N.- 43 Tarea 013 .....	65
Tabla N.- 44 Tarea 014 .....	66
Tabla N.- 45 Tarea 015 .....	66
Tabla N.- 46 Tarea 016 .....	66
Tabla N.- 47 Tarea 017 .....	66
Tabla N.- 48 Tarea 018 .....	67
Tabla N.- 49 Tarea 019 .....	67
Tabla N.- 50 Tarea 020 .....	67
Tabla N.- 51 Tarea 021 .....	67
Tabla N.- 52 Tarea 022 .....	68
Tabla N.- 53 Tarea 023 .....	68
Tabla N.- 54 Tarea 024 .....	68
Tabla N.- 55 Tarea 025 .....	68
Tabla N.- 56 Tarea 026 .....	69
Tabla N.- 57 Tarea 027 .....	69
Tabla N.- 58 Tarea 028 .....	69
Tabla N.- 59 Tarea 029 .....	70
Tabla N.- 60 Tarea 030 .....	70
Tabla N.- 61 Tarea 031 .....	70
Tabla N.- 62 Tarea 032 .....	70

Tabla N.- 63 Tarea 033 .....	71
Tabla N.- 64 Tarea 034 .....	71
Tabla N.- 65 Tarea 035 .....	71
Tabla N.- 66 Tarea 036 .....	72
Tabla N.- 67 Tarea 037 .....	72
Tabla N.- 68 Tarea 038 .....	72
Tabla N.- 69 Tarea 039 .....	73
Tabla N.- 70 Tarea 040 .....	73
Tabla N.- 71 Historias de Usuario.....	74
Tabla N.- 72 Historias de Usuario Iteración 1 .....	75
Tabla N.- 73 Historias de Usuario Iteración 2 .....	75
Tabla N.- 74 Historias de Usuario Iteración 3 .....	75
Tabla N.- 75 Historias de Usuario Iteración 4 .....	76
Tabla N.- 76 Plan de Entrega .....	78
Tabla N.- 77 Metáfora.....	79
Tabla N.- 78 Tarjetas CRC Ingreso al aplicativo conductor .....	81
Tabla N.- 79 Tarjeta CRC Iniciar ruta conductor .....	81
Tabla N.- 80 Tarjetas CRC Navegación recorrido conductor.....	81
Tabla N.- 81 Tarjeta CRC Visualizar línea pasajero.....	82
Tabla N.- 82 Tarjeta CRC Autenticación gestor web .....	82
Tabla N.- 83 Tarjeta CRC Crud conductor .....	82

Tabla N.- 84 Tarjeta CRC Crud bus.....	82
Tabla N.- 85 Tarjeta CRC Crud líneas y paradas.....	83
Tabla N.- 86 Tarjeta CRC Reporte rutas.....	83
Tabla N.- 87 Modelo de prueba de aceptación .....	151
Tabla N.- 88 Prueba de aceptación 01 .....	152
Tabla N.- 89 Prueba de aceptación 02 .....	153
Tabla N.- 90 Prueba de aceptación 03 .....	154
Tabla N.- 91 Prueba de aceptación 04 .....	154
Tabla N.- 92 Prueba de aceptación 05 .....	155
Tabla N.- 93 Prueba de aceptación 06 .....	155
Tabla N.- 94 Prueba de aceptación 07 .....	156
Tabla N.- 95 Prueba de aceptación 08 .....	157
Tabla N.- 96 Prueba de aceptación 09 .....	157
Tabla N.- 97 Prueba de aceptación 10 .....	158

## Índice de Figuras

Figura 1: Mapa de la Tierra mostrando la longitud y la latitud .....	5
Figura 2 Visual Studio - Proyecto abierto con ventanas principales y su funcionalidad .....	7
Figura 3 Componentes principales de Patron MVC .....	13
Figura 4 Como funciona Xamarin.Forms .....	16
Figura 5 Ejemplo Tablero Kanban.....	21
Figura 6 Planificación de procesos de la Metodología XP .....	23
Figura 7 Pregunta 1 .....	33
Figura 8 Pregunta 2.....	34
Figura 9 Pregunta 3 .....	35
Figura 10 Pregunta 4.....	36
Figura 11 Pregunta 5 .....	37
Figura 12 Análisis de los datos para el seguimiento en tiempo real de los recorridos de buses privados de Plasticaucho industrial. ....	43
Figura 13 Estructura de geolocalización.....	45
Figura 14 Arquitectura del sistema .....	84
Figura 15 Diseño de pantalla login conductor .....	86
Figura 16 Diseño de pantalla de ingreso de la ruta .....	87
Figura 17 Diseño de pantalla del recorrido .....	87
Figura 18 Diseño de pantalla de pasajeros.....	88
Figura 19 Diseño de pantalla login administrativo .....	89
Figura 20 Diseño de pantalla de conductores .....	89
Figura 21 Diseño de pantalla de buses.....	90
Figura 22 Diseño de pantallas de líneas.....	90
Figura 23 Diseño de pantalla de paradas.....	91
Figura 24 Diseño de pantallas obtención de coordenadas geográficas .....	91
Figura 25 Base de datos modelo, entidad, relación.....	92
Figura 26 Webs Apis Conductor.....	98
Figura 27 Webs Apis bus .....	99
Figura 28 Webs apis línea.....	100
Figura 29 Webs apis parada .....	101



Figura 30 Webs Apis Ruta .....	102
Figura 31 Webs Apis recorrido .....	103
Figura 32 Webs Apis notificaciones .....	104
Figura 33 Pantalla login conductor .....	113
Figura 34 Pantalla inicio de ruta .....	115
Figura 35 Pantalla navegación ruta .....	118
Figura 36 Pantalla master page .....	119
Figura 37 Pantalla crud conductor .....	123
Figura 38 Pantalla crud conductor ingreso de datos .....	123
Figura 39 Pantalla crud bus.....	127
Figura 40 Pantalla crud lineas- paradas .....	134
Figura 41 Pantalla obtener coordenadas .....	136
Figura 42 Pantalla selección de linea por fecha .....	137
Figura 43 Pantalla visualización de rutas.....	137
Figura 44 Selección de ruta.....	139
Figura 45 Pantalla de lista de coordenadas .....	139
Figura 46 Pantalla de coordenadas segmentadas .....	141
Figura 47 Pantalla de lista de coordenadas graficadas en puntos .....	145
Figura 48 Pantalla de lista de coordenadas graficadas en línea .....	145
Figura 49 Pantalla graficación punto único .....	146
Figura 50 Ambiente de pruebas apis.....	147
Figura 51 Ambiente de pruebas base de datos .....	147
Figura 52 Emulador tipo Tablet .....	148
Figura 53 Emulador tipo celular .....	148
Figura 54 Apis web services .....	149
Figura 55 Estructura del proyecto app móvil conductor.....	150
Figura 56 Estructura de proyecto gestión administrativa.....	151

## **RESUMEN EJECUTIVO**

Retos logísticos en el transporte personal y público actualmente necesitan de sistemas para conocer su ubicación en tiempo real, entonces se pretende trabajar en un Aplicativo Móvil de Geo-Ubicación en tiempo real, para el Transporte Privado de Plasticaucho Industrial S.A de la Ciudad de Ambato, el presente aplicativo nos servirá para la administración y control de los procesos de movilizaciones. Por esta razón para alcanzar nuestro objetivo principal, primero hay que analizar los procesos de recorrido que realiza Plasticaucho Industrial S.A., seguido a esto establecer la estructura de geolocalización para integrar a la aplicación móvil y por último integrar la aplicación a los procesos de movilización. La presente investigación fue de campo y bibliográfica, se hizo el uso de: encuestas, entrevistas y tesis, el nivel que abarco la investigación fue descriptiva, explicativa y exploratoria al considerarse que la investigación requería de varios puntos de vista. El desarrollo del aplicativo móvil recoge los puntos geográficos a través del GPS del dispositivo móvil, los envía a una base de datos y después es recuperado para visualizarlo en un mapa en tiempo real, con esto la administración puede sacar reportes de los recorridos. Como utilidad adicional, de la misma manera pueden observar la ubicación en tiempo real nuestros propios consumidores mediante el aplicativo móvil. Y cabe resaltar que el aplicativo registra de manera exitosa y con una exactitud alta, mejorando en la parte operativa para que puedan gestionar y anticipar su línea, reducir los retrasos y desde la aparte corporativa reducir costos de retribución por atrasos.

Palabras clave: Aplicativo móvil, GPS, tiempo real, rutas, transporte.

## **ABSTRACT**

Logistical challenges in personal and public transport currently require systems to know their location in real time, so it is intended to work on a Mobile Application of Geo-Location in real time, for the Private Transport of Plasticaucho Industrial S.A of the City of Ambato, this application will serve us for the administration and control of the mobilization processes. For this reason, to achieve our main objective, we must first analyze the travel processes carried out by Plasticaucho Industrial S.A., followed by establishing the geolocation structure to integrate the mobile application and finally integrate the application to the mobilization processes. The present investigation was field and bibliographical, the use of: surveys, interviews and thesis were made, the level that the investigation covered was descriptive, explanatory and exploratory, considering that the investigation required several points of view. The development of the mobile application collects the geographical points through the GPS of the mobile device, sends them to a database and is later retrieved to display it on a map in real time, with this the administration can draw reports of the routes. As an additional utility, in the same way our own consumers can observe the location in real time through the mobile application. And it should be noted that the application registers successfully and with high accuracy, improving in the operational part so that they can manage and anticipate their line, reduce delays and from the corporate side reduce compensation costs for arrears.

**Keywords:** Mobile application, GPS, real time, routes, transport.

## **CAPITULO I.- MARCO TEÓRICO**

### **1.1.Tema de Investigación**

“Aplicativo Móvil de Geo-Ubicación en tiempo real, para el Transporte Privado de Plasticaucho Industrial S.A de la Ciudad de Ambato.”

### **1.2. Antecedentes Investigativos**

#### **1.2.1. Contextualización del Problema**

Los continuos desafíos logísticos en el transporte personal y público requieren sistemas de ubicación en tiempo real, donde los sistemas de ubicación desempeñan un papel en la toma de decisiones y la gestión de la movilidad.

Las aplicaciones móviles son de gran interés a nivel mundial, es por ello que las aplicaciones móviles ya están disponibles; para algunas misiones, como en Bogotá, Colombia, tenemos una aplicación móvil que brinda información sobre patrones de viaje y rutas de origen y destino que el usuario utilizará para organizar los datos. Teniendo así la oportunidad de elegir compañeros de viaje para llegar más rápido y seguro a su destino [1].

Camargo Julián, González Luis, Segura Diego, Garay Fabian y Rincón Nubia publicaron un trabajo de investigación en 2017, “Lineamientos para Pasajeros con Discapacidad Visual en el Sistema de Transporte Público Transmilenio con Geolocalización Satelital”, enfocado en dispositivos portátiles de bajo costo elaborados con geolocalización satelital Posicionamiento, dirigido a aumentar la autonomía de las partes móviles de la población dentro del sistema Transmilenio, estos dispositivos se combinan con herramientas para traer a las poblaciones con discapacidad visual a la política [2].

En Ecuador se encontraron proyectos como “el desarrollo de un sistema web y una aplicación Android para smartphones que permita el control y seguimiento de la ubicación geográfica de las unidades de transporte pesado de Transporte y Logística Honores Méndez”. El sistema permite automatizar, administrar y controlar los procesos y recursos del negocio. Además, se obtuvo la ubicación actual de toda la flota

de Truck-Head que mueve y transporta mercancías en la ruta ecuatoriana [3].

Tungurahua cuenta con un proyecto “Desarrollo de una Aplicación Móvil con Geolocalización de Rutas de Buses y sus Paradas para el Gobierno Autónomo Descentralizado del Municipio Ambato” el cual señala la utilidad y necesidad de una aplicación móvil de geolocalización dentro de los límites de su destino. Aspectos a considerar “por transporte en autobús” ofrece grandes beneficios a los usuarios que pueden usar aplicaciones móviles de viaje para ver las rutas de manera eficiente. Y brindar recorridos guiados en la ciudad de Ambato que le permitan obtener información precisa y detallada sobre las diferentes rutas, así como saber qué ruta es mejor tomar para llegar a un lugar en particular [4].

### **1.2.2. Fundamentación Teórica**

Hay una gran cantidad de investigaciones que respaldan este argumento, ya que la inclusión de aplicaciones móviles para resolver problemas de tráfico mediante la geolocalización se ha vuelto importante en los últimos años.

Un estudio de Edison Vega de la Universidad de las Fuerzas Armadas ESPE titulado "Desarrollo de un Sistema de Geolocalización para Vigilancia Vehicular de Cooperativa Andina" propone el uso de Cooperativa Andina para crear una plataforma de monitoreo en tiempo real. La plataforma consta de tres componentes: una aplicación que puede ser instalada en un smartphone y ubicada en el bus de transporte, un sistema back-end instalado en un servidor, esta parte es la Api Rest, la última parte permite visualizar las unidades de transporte en el mapa y le permite ver el movimiento en tiempo real y también se muestra información adicional sobre la unidad de transporte mostrada [5].

Otro autor de Castro Alvarez Andres Arturo, cuyo proyecto "Desarrollo de un sistema GPS para la geolocalización y gestión de rutas de flotas de transporte público en Internet" se basa en el método de prototipo rápido para ayudar a los usuarios a controlar mejor las rutas de transporte público urbano Geolocalización de autobuses utilizando dispositivos con GPS (Posicionamiento Global) para verificar las rutas establecidas, esto se realiza mediante la tecnología GPS integrada en los teléfonos inteligentes - Android, que ahora tiene la misma funcionalidad que el GPS tradicional para determinar la ubicación exacta del autobús, los conductores pueden almacenar

información actualizada sobre tu ruta y control detallado en tiempo real por parte de un gestor asociado, mejorando la toma de decisiones [6].

Elias Chukija Arakayo es uno de los autores de "Desarrolló una aplicación móvil habilitada para geolocalización para mejorar la seguridad de los conductores de la exitosa empresa de taxis E.I.R.L." para monitorear y administrar. Para el desarrollo de la aplicación móvil se utilizó el método SCRUM en relación a las necesidades de la empresa, lo que ayudó al control y gestión de los conductores y la seguridad de los conductores de la exitosa empresa de taxis E.I.R.L. 20% esto es para evitar rutas peligrosas y saqueos [7].

### **1.2.2.1. Geolocalización**

Este es un término relativamente nuevo que abarca muchas técnicas destinadas a utilizar información relacionada con la ubicación geográfica en cualquier parte del mundo [8].

La geolocalización en Internet es un instrumento de comunicación entre la oferta y la demanda en un estado que llamaremos SOLOMO (Social, Local y Móvil), dicha información es emitida diariamente y compartida a través de las redes sociales, con un mecanismo local y enviada mediante los móviles desde cualquier sitio del mundo [9].

#### **1.2.2.1.1. Ventajas de la Geolocalización**

El uso de la geolocalización nos contribuye numerosas ventajas en todos los horizontes de Google, como los siguientes:

Posicionamiento Natural SEO: cuanto más uso se dé, la geolocalización mejor se posiciona de forma natural en el buscador de Google [9].

Posicionamiento por geolocalización GEO: con la entrada en Google Places y haciendo que la gente señale y valore el negocio, esto dará mayor relevancia en Google Maps, en Google Local y en el buscador de Google [9].

#### **1.2.2.1.2. GPS**

Este servicio es vital para que las personas sepan llegar a un punto concreto de nuestro territorio, ya que el GPS es uno de los principales métodos en este campo y precursor de nuevas propuestas [10].

Los satélites GPS operan con la Tierra, realizando dos órbitas precisas al día, enviando señales a la Tierra indicando su posición y hora, proporcionadas por relojes atómicos a bordo. Todos los satélites están sincronizados, por lo que las señales se envían al mismo tiempo. El receptor GPS toma esta información y la utiliza para triangular y automatizar la ubicación exacta del receptor. La señal viaja a la velocidad de la luz y llega al receptor en momentos diferentes porque algunos satélites están más lejos que otros. Básicamente, un receptor GPS en tierra compara el tiempo que tarda una señal en transmitirse desde uno de los satélites al espacio con el tiempo que tarda el receptor en recibir la misma señal. La diferencia le dice al receptor GPS qué tan lejos estás de los satélites. Ahora, al recibir múltiples mediciones de diferentes satélites, un receptor GPS puede determinar la posición del usuario con mucha precisión y mostrarla en un mapa electrónico en la unidad receptora. Cuando el receptor estima la distancia a por lo menos cuatro satélites GPS, puede calcular su ubicación a partir de tres dimensiones: longitud, latitud y altitud.

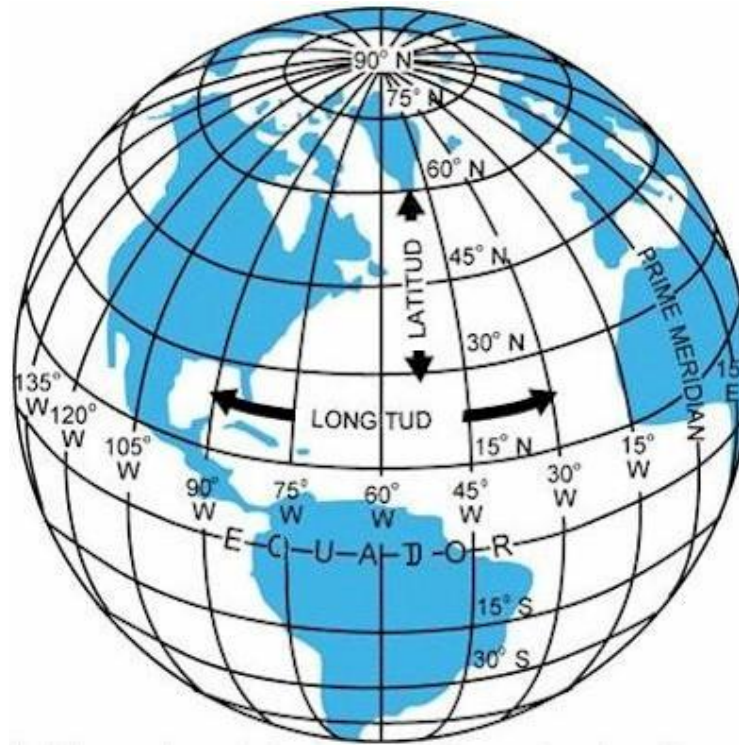
Los paralelos y meridianos de una red geográfica crean líneas imaginarias que permiten determinar la ubicación de cualquier punto de la superficie terrestre. Se definen por coordenadas geográficas o geográficas como la latitud y la longitud.

- **Latitud:** La distancia entre el punto y el ecuador. El ecuador es la línea de base y por lo tanto se considera latitud  $0^\circ$ , y los puntos dentro del mismo rango tienen la misma latitud. Los puntos en el sur de Ecuador reciben el mismo nombre Sur (S) con un signo menos; los del norte se denominan Norte (N) con un signo más. Tenga en cuenta que la latitud es siempre inferior a  $90^\circ$ .
- **Longitud:** a distancia entre este punto y el Meridiano de Greenwich. El meridiano de Greenwich corresponde a la longitud  $0^\circ$ , y los puntos del mismo meridiano tienen la misma longitud. Un punto al este del Meridiano de Greenwich se llama Este (E); un punto al oeste del Meridiano de Greenwich se llama Oeste (W). Las distancias se miden de  $0^\circ$  a  $180^\circ$ , sin longitud en los polos norte y sur.

Si conoces las coordenadas geográficas de la tierra, puedes encontrar cualquier punto en la superficie terrestre. Basta con tomar el ecuador, trazar un arco desde el meridiano de Greenwich igual a la longitud, y si hay un polo norte en la parte superior, ya sea a

la izquierda (longitud oeste) o a la derecha (longitud este), en el caso del polo sur, serán opuestos.

Dibujar una línea de longitud al final del arco, mostrando un arco igual a la latitud, y marcar el punto correspondiente a las coordenadas conocidas.



**Figura 1:** Mapa de la Tierra mostrando la longitud y la latitud  
**Fuente:** [http://www.radiofrecuencia.com/tema.php?ID=QUE\\_SIGNIFICA\\_GPS](http://www.radiofrecuencia.com/tema.php?ID=QUE_SIGNIFICA_GPS)

Los receptores del GPS en la tierra calculan con precisión y rapidez los tres valores que son necesarios para ubicar en un mapa electrónico a un usuario:

- Valor de Longitud.
- Valor de Latitud.
- Valor de Altitud.

#### 1.2.2.2. Geo-vallas o Geo-cercas

Las geo-cercas son áreas circulares determinadas en torno de un recorrido específico en un mapa virtual, el usuario puede definir la posición y el radio de esta área, en la aplicación si un vehículo se localiza en un determinado lugar es porque está dentro de una geo-cerca [11].



#### **1.2.2.2.1. Ventajas o Beneficios de las Geo-cercas.**

Algunos de las principales ventajas de las Geo-cercas son:

- Al tener registrados los puntos geográficos se puede determinar una distancia de monitoreo.
- Al tener una cerca activa se puede registrar la salida o entrada a la cerca marcada y realizar varios procesos como registro en una base local o externa.
- Envío de notificaciones locales al dispositivo móvil al salir de la cerca.

#### **1.2.2.2.2. Principio de funcionamiento de las Geo-cercas**

Como ya se mencionó una geo-cerca es la marcación de un área o recorrido específico en un mapa virtual, con el principal objetivo de obtener avisos en tiempo real en cada momento que un vehículo con GPS incorporado ingrese o salga de dicha área.

Al implementar un sistema GPS en los vehículos contratados de la empresa, cuando el conductor inicia la ruta hace una petición al server para extraer una lista de coordenadas geográficas (latitud y longitud) una vez obtenido estos datos la aplicación transforma estas coordenadas geográficas en cercas virtuales a continuación el dispositivo está escuchando en segundo plano si el conductor pasa por una de estas cercas virtuales, mediante xamarin esencial el dispositivo envía una alerta sonora que está en una parada registrada en la base de datos, cuando el dispositivo detecta que se ha salido de la cerca virtual envía una notificación local al dispositivo señalando que está fuera de la geo-cerca o que ha salido de la parada.

#### **1.2.2.2.3. Tipos de Geo-cercas**

Tenemos 2 tipos de Geo-cercas:

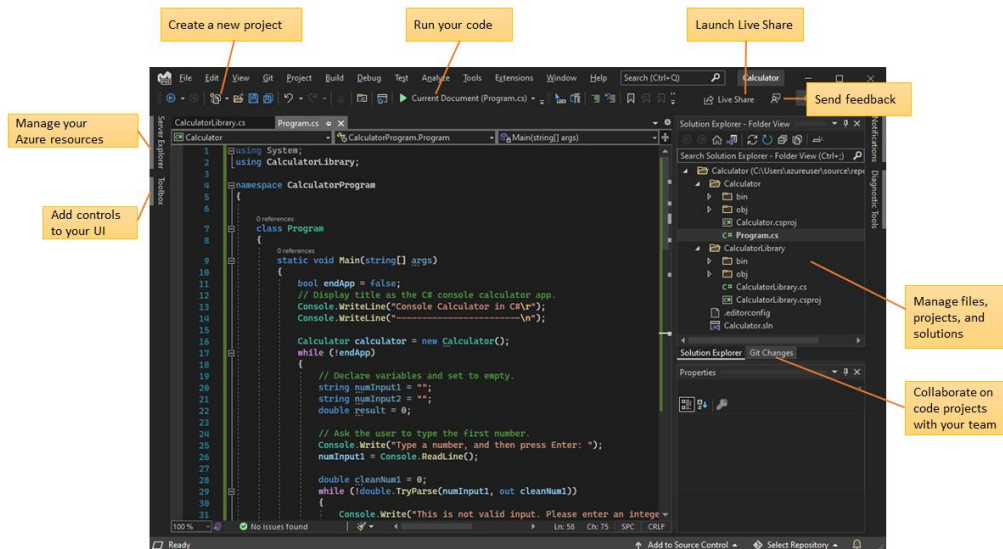
- **Círculo:** Este tipo de área contempla un centro y un radio de un círculo cualquiera, donde el radio mínimo considerado es de 20 metros. En este caso tanto el radio como el centro son definidos por el usuario del servicio.

- **Poligonal:** Este tipo de área está definida por un polígono cualquiera; donde el número de vértices de este puede ser cualquiera.

### 1.2.2.3. Visual Estudio

Interfaz de desarrollo de Microsoft. Está compuesto de un conjunto de herramientas que permiten a los desarrolladores la posibilidad de crear aplicaciones para plataformas .NET. Visual Studio [12].

El IDE de Visual Studio es un panel de inicio creativo que se puede utilizar para editar, depurar y compilar código y, posteriormente, publicar una aplicación. Aparte del editor y el depurador estándar que suministran la mayoría de IDE, Visual Studio contiene compiladores, herramientas de terminación de código, diseñadores gráficos y varias características que facilitan el proceso de desarrollo de software [13].



*Figura 2 Visual Studio - Proyecto abierto con ventanas principales y su funcionalidad  
Elaborado por: Microsoft*

Ventanas principales y su funcionalidad de Visual Studio [13]:

- En la parte superior derecha del Explorador de soluciones, se puede observar y administrar los archivos de código y navegar por ellos.
- La ventana del editor central, muestra el contenido del archivo. Aquí, puede editar código o diseñar una interfaz de usuario, como una ventana con botones y cuadros de texto.

- En la parte inferior derecha de Git Changes, puede realizar el respectivo seguimiento de elementos de trabajo y compartir código con otros usuarios por medio de tecnologías de control de versiones, como son Git y GitHub.

#### **1.2.2.3.1. Ediciones**

Visual Studio se encuentra disponible para Windows y Mac. Para Mac tiene características similares a las de Visual Studio para Windows y se optimiza para desarrollar aplicaciones móviles y multiplataforma. Existen tres ediciones de Visual Studio: Community, Professional y Enterprise.

#### **1.2.2.3.2. Características de productividad populares**

- **Subrayados ondulados y Acciones rápidas**

Los subrayados ondulados son rayas con formas de onda debajo de las palabras que ayudan a alertar de los errores o posibles problemas en el código a medida que se va escribiendo. Estas pistas visuales ayudan a corregir problemas inmediatamente, sin esperar a detectar los errores durante la compilación o el tiempo de ejecución. Además, si mantiene el puntero sobre un subrayado rojo, obtendrá más información sobre el error. También puede aparecer una bombilla en el margen de lado izquierdo que muestra acciones rápidas que puede hacer para corregir el error [13].

- **Limpieza de código**

Con dar el clic de un botón, se puede dar formato al código y utilizar cualquier corrección de dicho código sugerida por la configuración del estilo. La limpieza de código, que se encuentra disponible actualmente solo para el código de C#, ayuda a solucionar problemas antes de que pase a revisión [13].

- **Refactorización**

En la refactorización se incluye ciertas operaciones como el cambio de nombre de variables, la extracción de una o varias líneas de código en un método nuevo y el cambio del orden de los métodos de parámetros [13].

- **IntelliSense**

IntelliSense es un conjunto de características que brindan información sobre el código directamente en el editor y se escriben fragmentos de código automáticamente. Es como poder tener documentación primordial insertada en el editor, lo que evita tener que buscar información escrita en cualquier otro lugar [13].

#### **1.2.2.3.3. Licencias de Visual Estudio**

Hay varias formas de adquirir Visual Studio según sus necesidades. Lo primordial es comprar una suscripción. Proporciona un conjunto completo de herramientas y recursos para crear, implementar y administrar su próxima gran aplicación. Todo disponible en la plataforma y el dispositivo que prefiera, con la eficacia del IDE de Visual Studio [13].

- **Visual Studio Community**

Según microsoft el mejor IDE para desarrolladores de NET y C++ en Windows. Completamente equipado con una matriz de herramientas y características para enaltecer y mejorar todas las fases del desarrollo de software.

- **Visual Studio Community para Mac**

Incluye el soporte técnico de primera categoría para desarrollo web, en la nube y de juegos, también de excelentes herramientas para crear aplicaciones móviles multiplataforma.

- **Visual Studio Code**

Es un editor de código fuente independiente que se establece en Windows, macOS y Linux. La elección principal para desarrollar web y JavaScript, con gran cantidad de extensiones para admitir cualquier lenguaje de programación.

#### **1.2.2.4. PostgreSQL**

PostgreSQL es un gestor de bases de datos que se orienta a objetos conocido por el conjunto de funciones avanzadas que soporta, es un gestor de base de datos relacional,

el origen de PostgreSQL tiene como inicio en el gestor de bases de datos POSTGRES desarrollado en la Universidad de Berkeley y que posteriormente lo retomaría en favor de PostgreSQL a partir de 1994 [14].

PostgreSQL se distribuye bajo licencia BSD, lo que permite su uso, redistribución, modificación es decir es de código abierto, con la única restricción de mantener el copyright del software a sus autores [14].

#### **1.2.2.4.1. ¿Quién utiliza PostgreSQL?**

Lo utiliza una gran variedad de campos como servicios de fabricación, financieros comercio mayorista y minorista y logística. PostgreSQL ayuda a los desarrolladores a mantener la integridad de datos, escalar recursos y administrar las cargas de trabajo de todos los tamaños [15].

Tiene una reputación sólida y ofrece ventajas importantes para las aplicaciones geoespaciales, los administradores reconocen la confiabilidad que posee al momento de protegerlos datos y se ha ganado la popularidad por sus características en expansión y sus colaboradores [15].

#### **1.2.2.4.2. ¿Por qué usar PostgreSQL?**

Los desarrolladores visualizan sus excelentes características como su seguridad, solides, es compatible con los sistemas operativos principales como Windows, Linux y Macintosh, además que admite texto, sonidos, videos e imágenes [15].

Hay muchas razones por las que los desarrolladores prefieren a PostgreSQL en lugar de MySQL como las siguientes:

- **Acceso a características muy eficaces**

Las características más importantes son la recuperación de los datos un momento dado, los controles de acceso, el registro de escritura, los espacios de las tablas, las transacciones y las copias de seguridad que se pueden guardar en línea.

- **Confiabilidad y cumplimiento normativo**

Varios años de desarrollo han ayudado a que sea muy tolerante a errores, es coherente, durable y tiene propiedades de aislamiento, cabe mencionar también que puede admitir varios lenguajes, clave externa, procedimientos de almacenaje y combinaciones, permite los tipos de datos más comunes y admite caracteres Unicode [15].

- **Licencia de código abierto**

Ya que tiene una licencia de código abierto, esto permite a que los usuarios sean más flexibles e innovadores. No tiene costos de licencia, permite que los usuarios puedan explorar un sin número de posibilidades de desarrollo, además de modificar el código como deseen [15].

- **Escalabilidad**

El software tiene la capacidad para administrar con facilidad una gran cantidad de datos además puede contener una cantidad de usuarios simultáneamente que puede administrar [15].

- **Variedad de tipos de índice**

Permite obtener a los usuarios una gran cantidad de técnicas indexadas, como el índice de árbol B+, el índice invertido y el árbol de búsqueda, además de las búsquedas de texto para búsquedas de cadena y cadenas de operaciones vectoriales [15].

- **Flexibilidad**

PostgreSQL es compatible con muchos de los principales lenguajes y protocolos de programación, incluidos C, C++, Go, Perl, Python, Java, .Net, Ruby, ODBC y Tcl. Esto significa que los usuarios pueden trabajar en el idioma con el que están más familiarizados sin riesgo de conflictos en el sistema [15].

- **Un completo ecosistema de soporte técnico**

El código fuente abierto de PostgreSQL brinda a los usuarios soporte técnico de una comunidad dedicada de colaboradores que mejoran continuamente el sistema para

hacerlo más seguro y actualizado. Los usuarios pueden aprovechar el conocimiento colectivo de esta comunidad para informar errores, proporcionar comentarios sobre proyectos de desarrollo y responder preguntas. El código fuente abierto de PostgreSQL brinda a los usuarios múltiples formas de obtener soporte técnico, ya sea que trabajen en las plataformas Ruby on Rails, Tableau o Datadog. Además de acceder al conocimiento y la asistencia colectivos de la comunidad, los desarrolladores también pueden conectarse con expertos y servicios de PostgreSQL para solucionar problemas o determinar los próximos pasos. [15].

- **JSON**

Dado que PostgreSQL admite consultas relacionales y no relacionales, los usuarios pueden acceder a datos JSON mediante expresiones de ruta SQL y JSON [15].

- **Extensibilidad**

PostgreSQL hace más que solo almacenar datos. El software permite a los usuarios definir lenguajes funcionales y tipos de datos, incluidos tipos personalizados o definidos por el usuario. Además, los desarrolladores de su empresa pueden aprovechar muchas extensiones y complementos disponibles para personalizar PostgreSQL, como PostGIS, Citus, pg\_cron, HyperLogLog y t-digest. PostgreSQL logra este grado de escalabilidad porque, a diferencia de la mayoría de los sistemas de administración de bases de datos relacionales, almacena más información en el catálogo que en las tablas y columnas tradicionales. Los usuarios también pueden modificar tablas y extender PostgreSQL [15].

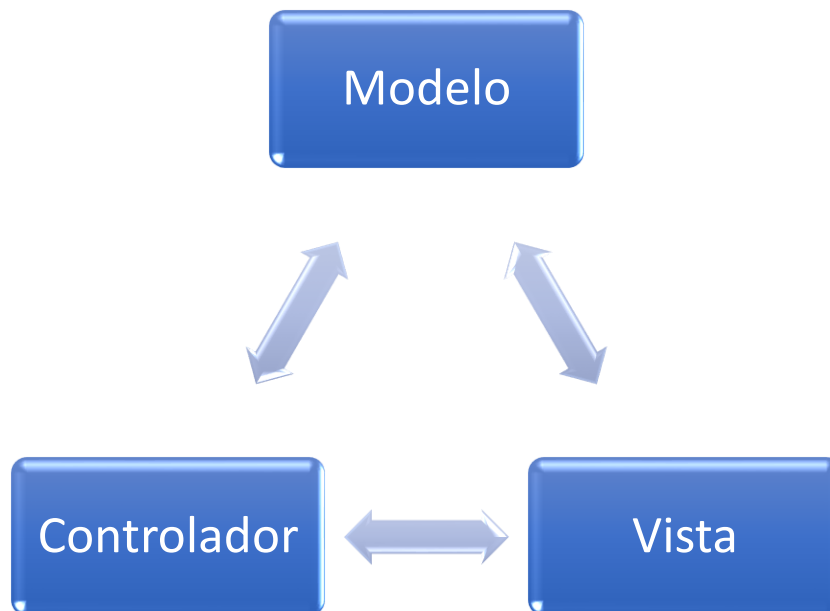
#### **1.2.2.5. ASP.NET MVC**

ASP. Net MVC (Model, View, Controller) es una tecnología web desarrollada por Microsoft y tiene la estructura de Model View Controller, por lo que se denomina MVC para abreviar. Es una tecnología que favorece el desarrollo de ASP. NET WEB Forms, que ayuda a los desarrolladores a desarrollar un entorno web [16].

### 1.2.2.5.1. Patrón MVC

El patrón arquitectónico Model View Controller (MVC) divide una aplicación en tres grupos principales de componentes: modelo, vista y controlador. Este modelo crea una separación de intereses. Usando este patrón, la solicitud del usuario se reenvía al controlador, que es responsable de trabajar con el modelo para realizar la acción del usuario u obtener el resultado de la solicitud. El controlador selecciona la vista para mostrar al usuario y proporciona todos los datos necesarios para el modelo [17].

El siguiente diagrama muestra tres componentes principales, uno de los cuales hace referencia a los demás:



*Figura 3 Componentes principales de Patron MVC  
Elaborado por: Franklin Lesano*

Esta separación de funciones facilita la extensión de una aplicación, ya que es más fácil codificar, depurar y probar cosas que solo tienen un trabajo (modelo, pantalla o controlador). Es más difícil actualizar, probar y depurar el código que tiene dependencias en dos o más regiones distribuidas en las tres regiones. Por ejemplo, la lógica de la interfaz de usuario cambia con más frecuencia que la lógica empresarial. Si el código de presentación y la lógica comercial se combinan en un objeto, el objeto que contiene la lógica comercial debe cambiarse cada vez que cambia la interfaz de usuario. Esto a menudo conduce a errores y la lógica comercial debe volver a probarse después de cada pequeño cambio en la interfaz de usuario.



- **Responsabilidades del modelo**

Los modelos en una aplicación MVC representan el estado de la aplicación y cualquier lógica comercial u operación que la aplicación necesite realizar. La lógica empresarial debe estar encapsulada en el modelo junto con cualquier lógica de implementación para mantener el estado de la aplicación. Una vista fuertemente tapada generalmente usa el tipo ViewModel para almacenar los datos que se muestran en la vista. El controlador crea y rellena estas instancias de ViewModel a partir del modelo.

- **Responsabilidades de las vistas**

Las vistas son responsables de presentar el contenido a través de la interfaz de usuario. Utilizan el motor de vista Razor para inyectar código .NET en el marcado HTML. Debe haber una lógica mínima entre las vistas y toda la lógica debe ser relativa a la vista de contenido. Si encuentra que tiene que poner mucha lógica en sus archivos de vista para mostrar datos de modelos complejos, considere usar componentes de vista, modelos de vista o plantillas de vista para simplificar sus vistas.

- **Responsabilidades del controlador**

Un controlador es un componente que maneja la interacción del usuario, interactúa con el modelo y finalmente selecciona la vista para mostrar. En una aplicación MVC, las vistas solo muestran información; los controladores monitorean y responden a las interacciones y datos ingresados por el usuario. En el patrón MVC, el controlador es el primer punto de entrada y es responsable de elegir el tipo de modelo a usar y la vista a mostrar (de ahí el nombre porque determina cómo responde la aplicación a una solicitud en particular).

- **API web**

ASP.NET Core MVC no solo es una gran plataforma para crear sitios web, sino que también brinda un soporte excelente para crear API web. Puede crear servicios que sean fácilmente accesibles para una variedad de clientes, incluidos navegadores y dispositivos móviles.

El marco incluye soporte para la negociación de contenido HTTP y soporte integrado para formatear datos como JSON o XML. Escriba su propio formateador para agregar compatibilidad con sus propios formatos.

Utilice la generación de enlaces para habilitar la compatibilidad con hipermedia. Habilite fácilmente la compatibilidad con el uso compartido de recursos de origen cruzado (CORS) para que las API web se puedan compartir entre varias aplicaciones web.

- **Componentes de vista**

Los componentes de visualización le permiten empaquetar y reutilizar la lógica de visualización en toda su aplicación. Son similares a las vistas parciales, pero con la lógica correcta.

#### **1.2.2.6. Xamarin Forms**

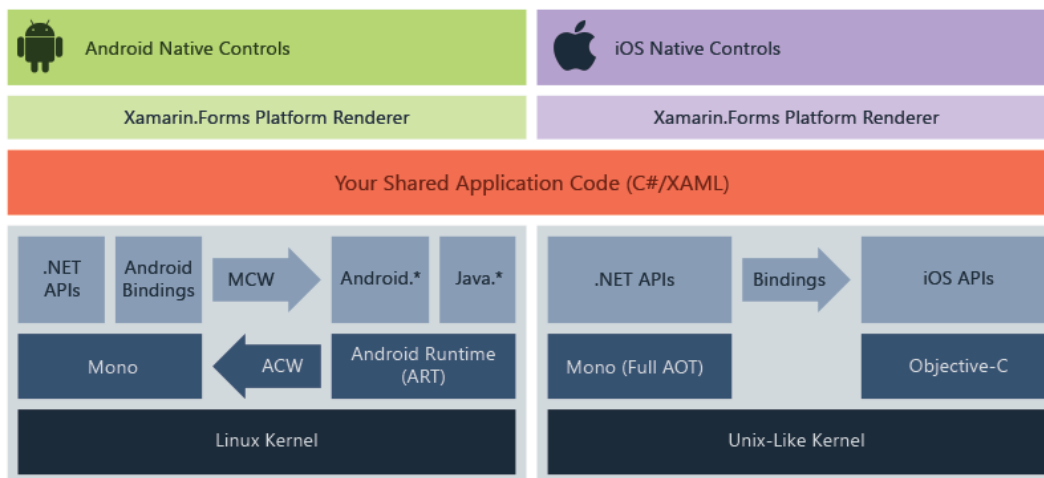
XAMARIN es una herramienta multiplataforma que permite crear aplicaciones Android e IOS desde un mismo desarrollo utilizando C# como framework y librería. Este entorno de desarrollo de programación integrado (IDE) hace que el desarrollo sea más rápido y sencillo. aplicación [18].

##### **1.2.2.6.1. A quién va destinado Xamarin.Forms**

Xamarin.Forms es para los desarrolladores que tienen los siguientes objetivos:

- Diseño de interfaz de usuario universal en todas las plataformas.
- Comparta código, pruebas y lógica empresarial entre plataformas.
- Escribir aplicaciones multiplataforma en C# utilizando Visual Studio.

### 1.2.2.6.2. Cómo funciona Xamarin.Forms



*Figura 4 Como funciona Xamarin.Forms  
Elaborado por: Franklin Lesano*

Xamarin.Forms proporciona una API coherente para crear elementos de interfaz de usuario en todas las plataformas. Esta API se puede efectuar en XAML o C# y acepta el enlace de estos datos para patrones como son Model-View-ViewModel (MVVM) [19].

En tiempo de ejecución, Xamarin.Forms usa procesadores de plataforma para convertir elementos de interfaz de usuario multiplataforma en controles nativos en Xamarin.Android, Xamarin.iOS y UWP. Esto permite a los desarrolladores obtener una apariencia natural, así como los beneficios de compartir código entre plataformas.

Las aplicaciones de Xamarin.Forms normalmente constan de bibliotecas .NET Standard compartidas y proyectos de plataforma independientes. Las bibliotecas compartidas contienen vistas XAML o C# y cualquier lógica comercial, como servicios, modelos u otro código. Un proyecto de plataforma contiene lógica o paquetes específicos de la plataforma que su aplicación necesita.

Xamarin.Forms usa Xamarin Framework para ejecutar aplicaciones .NET nativas en todas las plataformas. Para obtener más información sobre Xamarin Framework, consulte.

#### **1.2.2.6.3. Xamarin.Essentials**

Xamarin.Essentials es una biblioteca que proporciona API multiplataforma para la funcionalidad de dispositivos nativos. Xamarin.Essentials es una abstracción que simplifica el acceso a herramientas nativas.

#### **1.2.2.6.4. Características específicas de las plataformas**

Xamarin.Forms proporciona una API común para mostrar controles nativos en diferentes plataformas, pero una plataforma puede tener funciones que no están disponibles en otras plataformas. Por ejemplo, la plataforma Android tiene una funcionalidad integrada para el desplazamiento rápido en los objetos ListView, pero iOS no. Las características específicas de la plataforma de Xamarin.Forms le permiten usar características que solo están disponibles en plataformas específicas sin crear renderizadores o efectos personalizados.

Xamarin.Forms incluye soluciones listas para usar para varias características específicas de las plataformas Android e IOS.

#### **1.2.2.7. Manifiesto Ágil**

El factor humano es el principal aporte de éxito del proyecto de desarrollo de software, se valora más al equipo que el entorno, se basa en primero formar el equipo para que ellos construyan en entorno y no viceversa.

El desarrollo es más importante que la documentación, “No se debe producir documentos a menos que sean necesarios”, la generación de documentos debe ser cortos y muy objetivos.

La colaboración es importante entre el equipo de desarrollo y el cliente, en colaboración entre ambos para que el proyecto marche de forma exitosa. [21].

##### **1.2.2.7.1. Metodología Ágil Scrum**

Entre las metodologías de desarrollo ágil existentes, Scrum es una de las más reconocidas a nivel mundial. Su auge comenzó en la década de 1980, en un análisis de Ikujiro Nonaka e Hirotaka Takeuchi, quienes señalaron el trabajo en equipo en el desarrollo de productos de software y la autonomía que tienen. debería tener. A

principios de la década de 1990 fue adquirido por Jeff Sutherland y Ken Schwaber, quienes crearon un marco formal. El método se llama Scrum porque Nonaka y Takeuchi encontraron similitudes en el trabajo en equipo, formando un espíritu de equipo que empuja todo hacia adelante y persigue los mismos objetivos [22].

#### **1.2.2.7.2. Roles de Scrum**

Product owner (Dueño del producto) Es el Dueño que se encarga de financiar, controlar, gestionar y supervisar el proyecto, tomar las decisiones finales y convertir sus elementos en características. en desarrollo.

Sus áreas de responsabilidad son:

- Financiación del proyecto.
- Requisitos del sistema.
- Retorno de la inversión del proyecto.
- Lanzamiento del proyecto.

Scrum Master (Líder de Proyecto) Es el responsable de seguir el proceso scrum para alcanzar las metas y solucionar los problemas que se presenten en el proyecto, es el responsable de controlar el tiempo, los valores y seguir reglas que le ayuden a avanzar según lo planeado. Condicional. Se comunica con clientes y equipos. Coordina las reuniones diarias y es responsable de la mediación y resolución de problemas. Tiene que ser parte del equipo y trabajar con él.

Los equipos son responsables de crear y desarrollar software iterativo para aumentar la funcionalidad del software, es decir, convertir historias y documentos en software terminado. El equipo tiene la autoridad para reorganizar y definir actividades para garantizar un progreso óptimo del proyecto, así como para eliminar varias iteraciones y obstáculos propuestos. Tamaño ideal para equipos de 8 a 10 personas [23].

#### **1.2.2.7.3. Fases de Scrum**

##### **Preparación del Proyecto**

Es conocido como Sprit 0, o la fase inicial en esta se intenta comprender el ejercicio de la institución con la finalidad de tomar decisiones que añadan valor al producto.

Durante el tiempo que dura esta fase existen un gran número de inexactitudes con las estimaciones se presentan, sin embargo, esto es normal y se debe mejor enfocar en el desarrollo del producto [24].

Las tareas para realizar en la fase de inicio son las siguientes:

- Definir el proyecto.
- Definir terminado del proyecto.
- Definir el Backlog inicial.
- Definir los entregables.

### **Planificación del Sprint**

Se denomina Sprint Planning Meeting cuya finalidad es realizar una reunión para seleccionar de la lista de Backlog del producto las funcionalidades sobre las que se va a iniciar a trabajar. La reunión se realiza en 8 horas que se dividen en 2 partes de 4 horas [24].

Primera parte:

- Se selecciona los ítems para hacerlos entregables.
- El equipo hace sugerencias, pero quien decide es el Product Owner si serán parte o no del proyecto.

Segunda parte:

- El equipo realiza las preguntas que sean necesarias del Product Backlog al Product Owner.
- Responsable de encontrar soluciones que conviertan los proyectos seleccionados en funciones terminadas.

### **Etapas de Desarrollo**

Cuando el trabajo del sprint está en curso, los encargados deben avalar que no se generen permutaciones de último momento que puedan inquietar los objetivos de este. Además, que se asegura el cumplimiento de los plazos establecidos para su término, el plazo más conveniente está entre 2 y 4 semanas o 30 días consecutivos [24].

## **Ventas y Desventajas de Scrum**

### **Ventajas de Scrum**

Scrum es una de las mejores metodologías ágiles para trabajar en equipo, aquí podemos visualizar algunas ventajas de hacerlo.

- Satisfacción del cliente: esta metodología hace al cliente que sea parte del equipo de trabajo por lo que lo comprometen con el resultado.
- La flexibilidad del método Scrum permite modificar los hechos si así es necesario en cualquier momento, lo que ayuda a la resolución de problemas u obstáculos sobre la marcha.
- Evita el perfeccionismo innecesario.
- Se obtienen resultados rápidos y etapas de prueba muy cortos. Así, un producto puede estar listo para salir al público en poco tiempo.

### **Desventajas de Scrum**

Nada puede ser perfecto. Por ello, tenemos que hablar de las desventajas de Scrum [22]:

- Es posible que la funcionalidad pactada en un inicio cambie a causa de diversos motivos.
- En gran parte, el éxito de un proyecto conducido con el método Scrum dependerá de la preparación y experiencia del Scrum Master.
- Llegar a discusiones con los clientes debido a diferentes puntos de vista en cuando a las fases del Scrum.

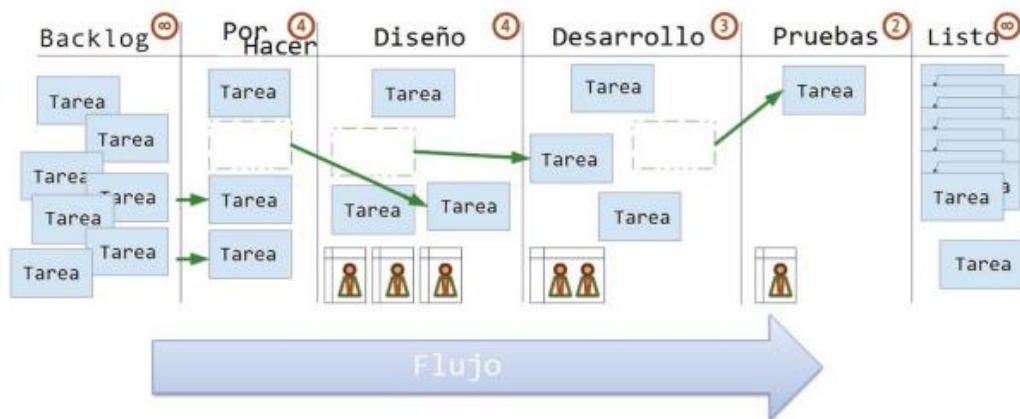
#### **1.2.2.7.4. Metodología Ágil Kanban**

La metodología Kanban es un sistema que sirve principalmente para asegurar una producción continua y para no sobrecargar el equipo de producción. Es decir, un sistema de producción *just in time* lo que es que produce exactamente la cantidad de producción que el sistema es capaz de asumir y así evita sobrantes innecesarios de stock [25].

## Reglas de Kanban

- **Visualizar el flujo de trabajo**

La metodología Kanban recomienda la utilización de tarjetas lo que brinda el nombre al tablero Kanban en el que se define y divide cada tarea en columnas que indican todas las fases del proyecto haciendo que sean visible para todos los miembros del equipo como en la imagen a continuación [26].



*Figura 5 Ejemplo Tablero Kanban*

*Elaborado por: Propuesta para nuevo enfoque de aplicación ágil con Scrum, LSD y Kanban [26].*

- **Limitar el trabajo en curso**

Cuando se inicia algo hay que terminarlo, no sirve de nada dejarlo a medias, este es el principio fundamental de la metodología Kanban y un pilar fundamental para que el proyecto funcione [26].

- **Hacer Políticas Explícitas**

Permite que todos los miembros del equipo conozcan cómo se trabaja. Puede definirse también como un criterio de aceptación y cada miembro del grupo debe tener su propio criterio [26].

- **Medir y Administrar el Flujo**

Existen varios conceptos que permiten medir el ritmo en el que el equipo está trabajando, algunos de ellos son:



- Lead Time: tiempo que pasa desde que una tarea ingresa al Backlog hasta que se entrega
- Cycle Time: lo que se demora justamente en desarrollarse una tarea, desde que entra a desarrollo
- Cuadro de Flujo Acumulativo: permite conocer el estado en el que se encuentra el proyecto [26].

- **Mejora Continua**

El tablero debe estar actualizado para que cualquiera que lo mire sepa exactamente en donde se encuentra el proyecto, esto para ayudar a que existan acumulaciones o retrasos. Aquí se sugiere realizar reuniones en las que se pueda re-priorizar las tareas para continuar con el proyecto [26].

## **Ventas y Desventajas**

### **Ventajas**

- Brinda información actualizada del estado del proyecto.
- Tiene una gran capacidad de detectar errores.
- Permite limitar y controlar el trabajo que se encuentra en proceso.
- El ciclo de producción se ve afectado positivamente gracias a que se optimiza el tiempo
- No se necesita una gran organización para implementarlo
- Se beneficia la productividad al mantener a cada parte del equipo haciendo alguna actividad.

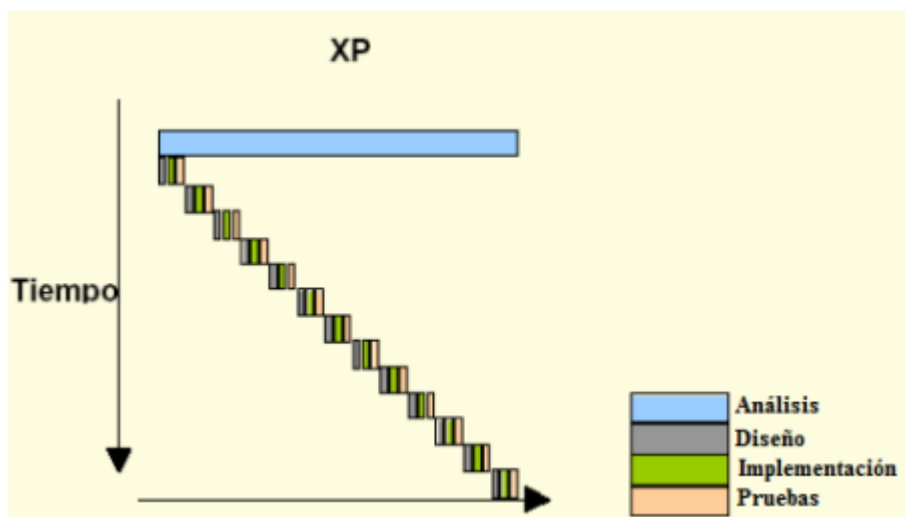
### **Entre las desventajas se encuentran:**

- Dificulta el prever posibles problemas.
- No se adapta a variaciones grandes de volúmenes de pedidos
- Debido a que la metodología Kanban limita el trabajo en proceso, esto puede causar cuellos de botella.

### 1.2.2.7.5. Metodología Ágil XP

La metodología XP es la más conocida, la principal particularidad de esta son las historias de usuarios, las que corresponden a una técnica de especificación de requisitos, se trata de formatos desarrollados en los cuales el cliente describe las funciones y características que debe tener el sistema [27].

La metodología se desarrolla en un proceso que se le denomina Planning game, que define la fecha del cumplimiento en base a las historias de usuario, estableciendo las características de entrega, costos de implementación y las interacciones para entregarla [27].



*Figura 6 Planificación de procesos de la Metodología XP  
Elaborado por: Metodologías ágiles frente a la tradicionales de desarrollo de software [27].*

### 1.2.2.8. Fases XP

#### **Fase de Exploración.**

En esta fase los clientes describen los requerimientos para las historias de usuarios. Cada una de estas historias se convierten luego en una funcionalidad. El tiempo varía de acuerdo con el conocimiento del desarrollador y complejidad.

#### **Fase de Planificación.**

En esta etapa se definirá las historias más importantes para la primera entrega. Tomar en cuenta que la primera entrega no debe pasar más allá de los dos meses.

### **Fase de Iteraciones.**

El tiempo se dividirá en iteraciones de acuerdo con la planificación, el tiempo puede variar entre una y dos semanas, la primera entrega suele contener toda la arquitectura del sistema. Tomar en cuenta que al final de cada iteración se realizaran pruebas con el usuario

### **Fase de Producción.**

Se realizarán pruebas extras antes de la entrega del proyecto. Si existe algún cambio se decidirá si se corrigen o se espera una próxima versión.

### **Fase de Mantenimiento.**

Una vez en producción se debe tomar en cuenta el soporte al usuario final y modificaciones para mejora del sistema [28].

#### **1.2.2.9. Roles de XP.**

- **Scrum Máster.**

Garantiza el funcionamiento en los procesos y la metodología, Debe ser miembro del equipo para trabajar al a par con su equipo [28].

- **Propietario del producto.**

Es el responsable oficial del proyecto, control, gestión y visibilidad de las historias de usuario [28].

- **Equipo de desarrollo.**

Miembros con los conocimientos de programación y son quienes deciden acciones necesarias o eliminar problemas en el proyecto [28].

- **El cliente.**

Responsable de la creación de las historias de usuarios [28].

- **El gestor.**

Toma las decisiones participa en la elección de objetivos [28].

## **Ventajas y Desventajas**

Existen ciertas ventajas y desventajas de XP las cuales señalamos a continuación [29].

### **Ventajas**

- Una mayor satisfacción del cliente, con diseños y versiones más cortas
- Flexibilidad en el proyecto teniendo más acercamiento y control con el cliente, manteniendo en cuenta las buenas prácticas de desarrollo
- La metodología XP facilita el desarrollo y disminuye los errores haciendo de mayor calidad el producto final.
- La planificación continua se llega a plazos mediáticos para evitar polémicas en el desarrollo entrega

### **Desventajas**

- La Carga de trabajo es inversamente proporcional al número de desarrolladores

No es una metodología ya que no describe un modelo de ciclo de vida ya que es un modelo evolutivo.

## **1.3. Objetivos**

### **1.3.2. Objetivo General**

Desarrollar un aplicativo móvil de geo-ubicación en tiempo real, para el transporte privado de Plasticaucho Industrial S.A de la ciudad de Ambato.

### **1.3.3. Objetivos Específicos**

- Analizar los procesos de recorrido para el transporte privado de Plasticaucho Industrial S.A.
- Establecer la estructura de geolocalización para integrar a la aplicación móvil.
- Implementar el aplicativo de geo-ubicación para el transporte privado de Plasticaucho Industrial S.A.

## CAPITULO II.- METODOLOGÍA

### 2.1. Materiales

Para el presente proyecto de investigación se utilizaron encuestas como entrevistas y tesis, como:

**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTA DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**

Entrevista dirigida al área de servicio laboral para recolección de procesos del manejo de las rutas.

Objetivo: Recolectar información dirigida a servicios laborales para la creación de un aplicativo móvil de geo-ubicación en tiempo real, para el transporte privado de Plasticaucho Industrial S.A de la ciudad de Ambato.

¿Cuántas rutas de transporte tiene la empresa?

---

¿Cuántos buses contratados tiene la empresa?

---

¿Cuántas veces al día realizan cada bus el recorrido?

---

¿Como saben los conductores porque ruta ir?

---

¿Cuál es el tiempo estimado en completar la ruta?

---

¿Como saben los usuarios en que parada deben estar y en que horario?

---

¿Tienen alguna pancarta mapa de las rutas que siguen los buses?

---

¿Qué pasa cuando un bus no completa la ruta?

---

¿Como controlan actualmente el inicio y llegada de los buses?

---

¿Cuántos conductores tiene contratado la empresa?

---

**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTA DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**

Encuesta dirigida para los conductores de transporte privado de la empresa Plasticaucho Industrial.

Objetivo: Recolectar información dirigida a conductores para el desarrollo de un aplicativo móvil de geo-ubicación en tiempo real, para el transporte privado de Plasticaucho Industrial S.A de la ciudad de Ambato.

Le solicitamos responder con la mayor sinceridad. Todos los datos que nos proporcione serán confidenciales como lo exigen las normas éticas dentro de investigación científica. De antemano agradecemos su valiosa colaboración. Instrucciones: Sírvase contestar marcando con una X la alternativa que mejor refleje su opinión. Ejemplo X.

Edad: \_\_\_\_\_

Tipo Licencia: \_\_\_\_\_

Tipo de transporte

Bus( )

Buseta( )

Otro.....

1.- Estaría dispuesto a utilizar una aplicación para control y seguimiento del recorrido de su unidad para que los usuarios sepan su ubicación en tiempo real.

Si \_\_\_\_\_

No \_\_\_\_\_

2.- Le gustaría tener una aplicación que le informara a los usuarios su próxima parada.?

Si \_\_\_\_\_

No \_\_\_\_\_

3.- Le gustaría tener una aplicación que le permita enviar notificaciones a los usuarios en caso de un siniestro o contratiempo.

Si \_\_\_\_\_

No \_\_\_\_\_

4.- A utilizado una aplicación para gestión de rutas o trayectos.

Si \_\_\_\_\_

No \_\_\_\_\_

5.- Actualmente tiene alguna aplicación que le ayude con la gestión y control del recorrido de su unidad.

Si \_\_\_\_\_

No \_\_\_\_\_

**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTA DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL**

Entrevista dirigida al área de Tecnologías de Información para recolección de las tecnologías aplicadas en el área.

Objetivo: Recolectar información dirigida al área de Tecnologías de Información para la creación de un aplicativo móvil de geo-ubicación en tiempo real, para el transporte privado de Plasticaucho Industrial S.A de la ciudad de Ambato.

¿Cuántos Lenguajes de programación manejan dentro del área?

---

¿En qué IDE de programación desarrollan los proyectos?

---

¿Cuántos gestores de base de datos manejan dentro del área?

---

¿Cómo están alojados sus servicios y ERP en la compañía?

---

¿Tiene servicios almacenados en la nube?

---

## 2.2. Métodos

### 2.2.1. Modalidad de la Investigación

**Investigación de Campo:** Porque en el lugar donde se produce el problema permite realizar el análisis de información así de esta manera se mantiene contacto cercano con las personas involucradas.

**Investigación Bibliográfico:** Porque se utiliza investigaciones relacionadas, libros y artículos científicos, las fuentes son fidedignos para realizar el desarrollo de la aplicación con conocimiento sostenible y eficaz.

#### 2.2.1.1. Nivel de investigación

**Descriptivo:** Permite ver de forma más clara y precisa la casusa de los problemas y deficiencias en los procesos administrativos.

**Explicativa:** Para determinar si los procesos de seguimiento mediante el análisis mejoran eficazmente.

**Explorativo:** Para indagar características del problema y aplicar pruebas de factibilidad.

### 2.2.2. Población y Muestra

La población para la investigación es de 11 Administrativos de la empresa Plasticaucho Industrial S.A., teniendo en cuenta que la población de empleados de las Áreas relevantes al estudio del proyecto no tiene una cantidad aceptable no es considerable tener una muestra de esta.

Nº	Área	Numero	Porcentaje
1	Servicios laborales	3	27.27%
2	Transporte privado	6	54.55%
3	Tecnología	2	18.18%
Total		11	100%

*Tabla N.- 1 Población y muestra  
Elaborado por: Franklin Lesano.*



### 2.2.3. Recolección de Información

De la entrevista realizada se pudo obtener gran información la cual tendrá como objetivo el desarrollo de una aplicación móvil que les permita tener control del inicio de los recorridos, visualización en tiempo real del trayecto que sigue y de su finalización. Al tener notificaciones para los usuarios en caso de siniestros, causa mayor para no cumplir la ruta el área de servicios laborales tomara decisiones y controles sobre la misma en tiempo real.

Preguntas	Entrevistado	Conclusiones
1. ¿Cuántas rutas de transporte tiene la empresa?	La empresa cuenta con 6 rutas distribuidas en los 8 horarios de la siguiente manera entrada de los operarios a las 06H00 am, salida de los operarios 06h30am, entrada del personal administrativo a las 08H30 am, entrada de los operarios a las 14H00 pm, salida de los operarios a las 14H30, salida del Personal Administrativo 17H30 pm, entrada de los operarios a las 22H00 pm, salida de los operarios 22H30am.	La implantación de un aplicativo GPS en tiempo real permitirá administrar de mejor manera las rutas y los buses que lo recorren en los diferentes horarios.
2. ¿Cuántos buses contratados tiene la empresa?	La empresa a través de un convenio cuenta con 8 buses de una cooperativa privada los cuales realizan los recorridos asignados.	Con la implementación del sistema GPS se asignará un dispositivo a acá uno de los conductores para registro y validación de recorrido a las rutas asignadas.
3. ¿Cuántas veces al día realizan cada bus el recorrido?	Los buses realizan hasta 3 veces el recorrido de sus rutas, aunque por fallas mecánicas o diferentes situaciones se reasigna o se pide a la empresa el cubrimiento del bus.	El aplicativo permitirá el seguimiento de del bus en la ruta asignada así también el conductor podrá enviar alertas o notificaciones si ocurre un siniestro en el camino.
4. ¿Como saben los conductores porque ruta ir?	El personal de servicios laborales tiene una ruta trazada en Google maps con la ruta que debe seguir estas rutas son aprobadas previamente por gerencia,	Con el aplicativo implantado se espera tener un control de las paradas predeterminadas, teniendo en cuenta que la ruta debe

	esto se informa de forma impresa al conductor.	ser la misma que se aprobó por gerencia.
5. ¿Cuál es el tiempo estimado en completar la ruta?	Dependiendo de la ruta está contemplado entre los 45 minutos y 1 hora en realizar el recorrido.	Con el tiempo estimado podremos saber el uso de datos que necesitara para cubrir la ruta, para que la empresa tome un plan de datos para el envío de la información.
6. ¿Como saben los usuarios en que parada deben estar y en que horario?	El departamento de servicios laborales elabora el recorrido en Google maps, luego de ello imprimen en hojas y los pegan en pizarras dentro de la empresa para que los trabajadores tengan conocimiento.	Con el aplicativo los trabajadores y usuarios de los buses podrán ver en tiempo real por donde están ubicado el bus gestionando mejor el tiempo y anticipando las paradas
7. ¿Tienen alguna pancarta mapa de las rutas que siguen los buses?	El área de servicios laborales elabora un croquis del recorrido y los imprime en papel las cuales son puestas en paneles ubicados alrededor de las plantas y en el comedor de la empresa.	El aplicativo permitirá en tiempo real ver la ruta de los diferentes buses para que los usuarios puedan gestionar sus rutas.
8. ¿Qué pasa cuando un bus no completa la ruta?	Cuando el bus no completa el recorrido, el conductor debe informar a servicios laborales que no se completó la ruta y se debe gestionar con otra unidad para cubrir el recorrido.	Con el Aplicativo y las alertas a los usuarios se tendrá un control en tiempo real de los incidentes que puedan ocurrir en el recorrido.
9. ¿Como controlan actualmente el inicio y llegada de los buses?	No se tiene un control adecuado del inicio y fin de la ruta, la retroalimentación de si no se completó o no se realizó el recorrido es provista por los usuarios del transporte al poner una queja en las oficinas de servicios laborales.	Con el aplicativo se tendrá un control de la hora de inicio, los puntos de recorrido durante todo el trayecto de la ruta y también la finalización de la ruta.
10. ¿Cuántos conductores tiene contratado la empresa?	Actualmente cada bus tiene su conductor asignado por lo que al igual que las unidades son 6, aunque depende de la empresa privada para él envió de unidades de respaldo y conductores.	Con el aplicativo también se tendrá el control de la persona asignada a la unidad y quien está cumpliendo con el recorrido.
11. ¿Cuántos lenguajes de programación	El área de tecnología de la información maneja Abap	Con el manejo de C# y la integración con Xamarin

manejan dentro del área?	para el manejo y modificación del ERP SAP, para modificaciones y otros proyectos manejan C# integrado con ASP, Net Core y Xamarin.	Forms se va a crear el aplicativo de GPS utilizando plugin como Xamarin essentials y Shiny para las cercas virtuales.
12. ¿En qué IDE de programación desarrollan los proyectos?	Los desarrolladores manejan visual estudio community 2019 y 2020, la licencia es community ya que la Enterprise representa un costo de licencia, se manejan otros desarrollos como visual estudio code para proyectos web y el entorno propio de abap para desarrollo.	Con el IDE de desarrollo de visual estudio 2019, se puede acceder al entorno de desarrollo de Xamarin forma para el desarrollo de aplicaciones híbridas en la cual nos permitirá la creación del aplicativo para el seguimiento del bus y para crear otra aplicación para los usuarios finales y la visualización en el mapa.
13. ¿Cuántos gestores de base de datos manejan dentro del área?	Se maneja con diferentes gestores de base de datos, siendo Oracle la principal y como alternativas a proyectos como balanza del muelle SQL Server 2014, SqlLite para dispositivo móviles	En base al requerimiento se puede sugerir un gestor de base de datos con licencia open source para evitar gastos en la compra de licencias, esto tendrá como consecuencia poca inversión en el desarrollo de la estructura del aplicativo.
14. ¿Cómo están alojados sus servicios y ERP en la compañía?	Los servicios tanto de aplicaciones como de DDB se mantienen en servidores diferentes externamente con un proveedor fuera del país, los servidores tienen Windows 2012 en el mismo segmento de red.	Los servicios que consumirán el aplicativo está construido con Net Framework por lo que es necesario que los servidores de alojamiento soporten o tengan IIS necesarios.
15. ¿Tiene servicios almacenados en la nube?	La empresa cuenta con servicios de almacenamiento en la nube, pero mayormente para archivos y datos de los usuarios de la compañía, estos servicios tienen contratado con office 365.	Es necesario saber para la integración directa como servicios de Azure que es nativo de Microsoft y la integración sea más rápida.

*Tabla N.- 2 Resultados de la entrevista  
Elaborado por: Franklin Lesano.*

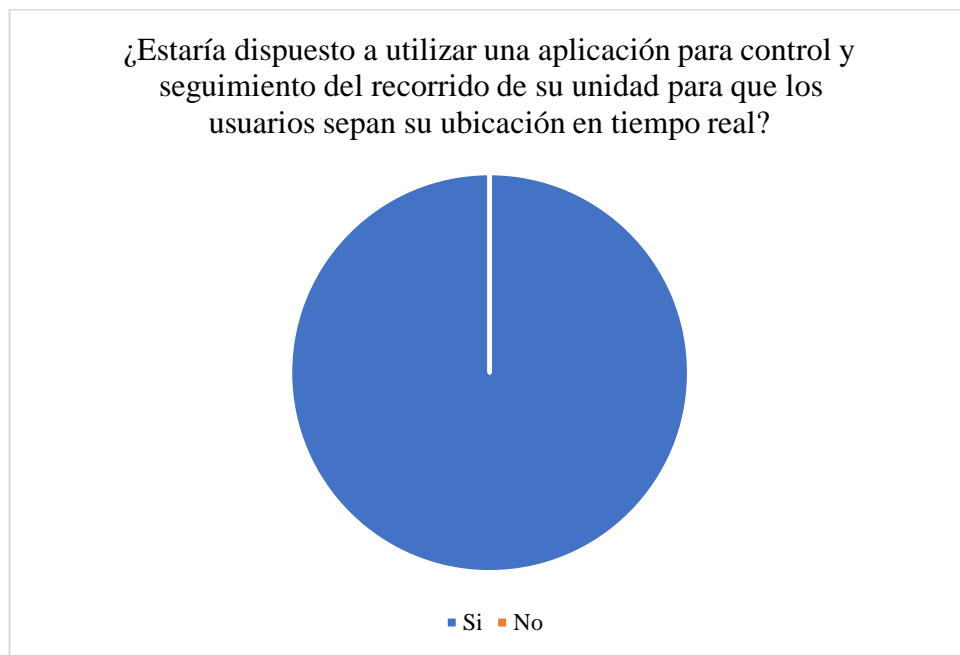
### 2.2.3.1. Resultados de Encuesta

- **Pregunta 1**

¿Estaría dispuesto a utilizar una aplicación para control y seguimiento del recorrido de su unidad para que los usuarios sepan su ubicación en tiempo real?

Alternativas	Frecuencia	Porcentaje
Si	6	100%
No	0	0%
Total	6	100%

*Tabla N.- 3 Pregunta 1 Encuesta  
Elaborado por: Franklin Lesano.*



*Figura 7 Pregunta 1  
Elaborado por: Franklin Lesano.*

#### **Análisis**

De acuerdo con la Figura número 3 se puede observar que la mayoría de los conductores está dispuesta a utilizar un aplicativo de seguimiento de ruta.

#### **Interpretación**

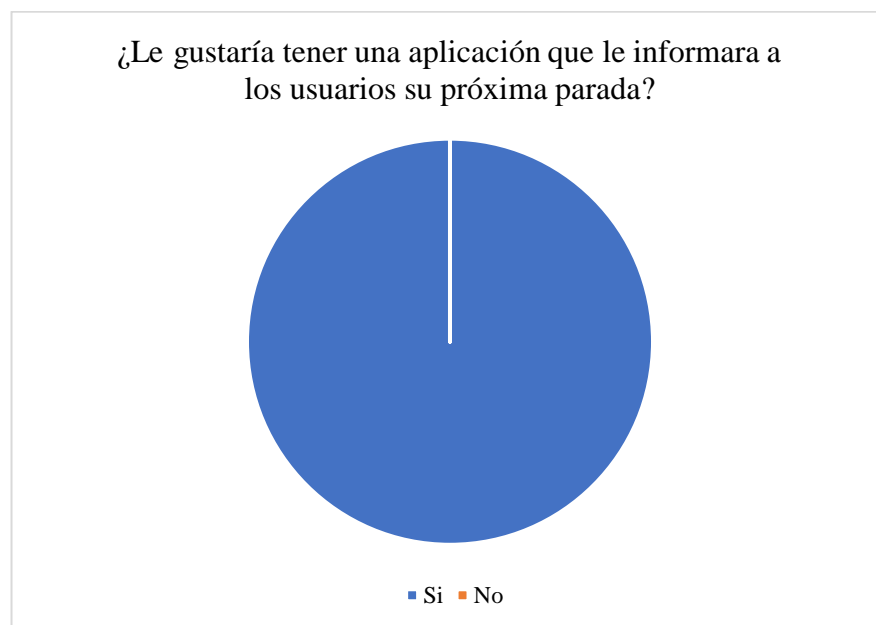
Los conductores están dispuestos a utilizar un aplicativo de seguimiento de las rutas.

- **Pregunta 2**

**¿Le gustaría tener una aplicación que le informara a los usuarios su próxima parada?**

<b>Alternativas</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	6	100%
No	0	0%
Total	6	100%

*Tabla N.- 4 Pregunta 2 Encuesta  
Elaborado por: Franklin Lesano.*



*Figura 8 Pregunta 2  
Elaborado por: Franklin Lesano.*

### **Análisis**

De acuerdo con la Figura número 4 podemos observar que los conductores les gustaría que el aplicativo les informara de sus próximas paradas

### **Interpretación**

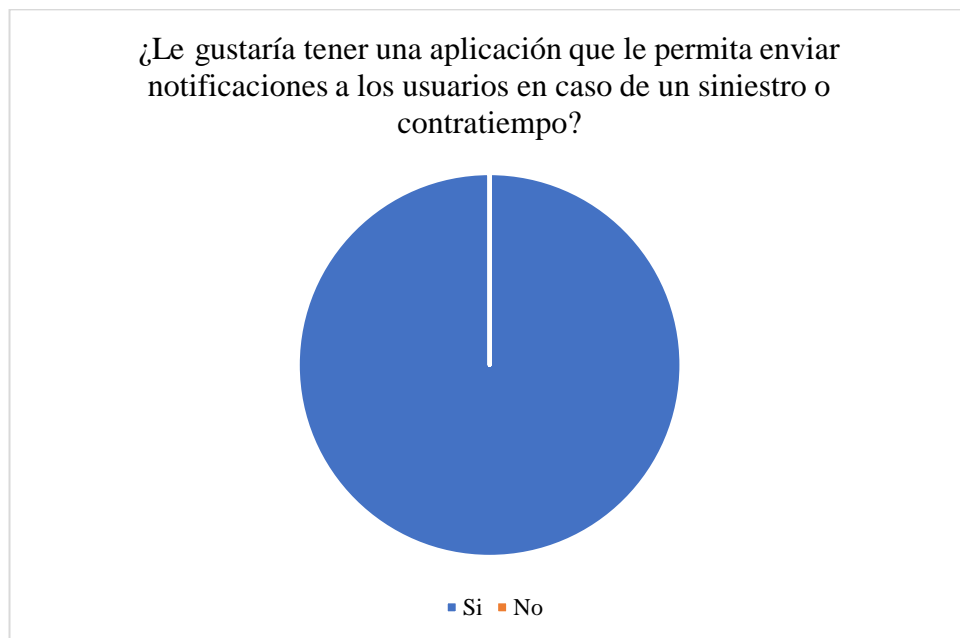
El aplicativo informará tanto a los conductores como a los usuarios de las próximas aradas, esto mantendrá a los conductores informados de las próximas paradas que deben tomar.

- **Pregunta 3**

**¿Le gustaría tener una aplicación que le permita enviar notificaciones a los usuarios en caso de un siniestro o contratiempo?**

Alternativas	Frecuencia	Porcentaje
Si	6	100%
No	0	0%
Total	6	100%

*Tabla N.- 5 Pregunta 3 Encuesta  
Elaborado por: Franklin Lesano.*



*Figura 9 Pregunta 3  
Elaborado por: Franklin Lesano.*

### **Análisis**

Como se observa en la Figura número 5 le gustaría tener un aplicativo para él envío de notificaciones a los usuarios de alertas en caso de algún siniestro su citado en el recorrido.

### **Interpretación**

El aplicativo enviará notificaciones de siniestro en ruta lo que permitirá a los usuarios tomar acciones y tomar acciones para tomar otro autobús.

- **Pregunta 4**

**¿Ha utilizado una aplicación para gestión de rutas o trayectos?**

Alternativas	Frecuencia	Porcentaje
Si	2	33.33%
No	4	66.67%
Total	6	100%

*Tabla N.- 6 Pregunta 4 Encuesta  
Elaborado por: Franklin Lesano.*



*Figura 10 Pregunta 4  
Elaborado por: Franklin Lesano.*

### **Análisis**

Como podemos observar en la Figura número 6 solo el 33.33% de los conductores a utilizado o conoce aplicaciones de trayectos o seguimiento de rutas, mientras el 66.67% de los conductores no ha utilizado algún tipo de aplicativo para el seguimiento o control de ruta.

### **Interpretación**

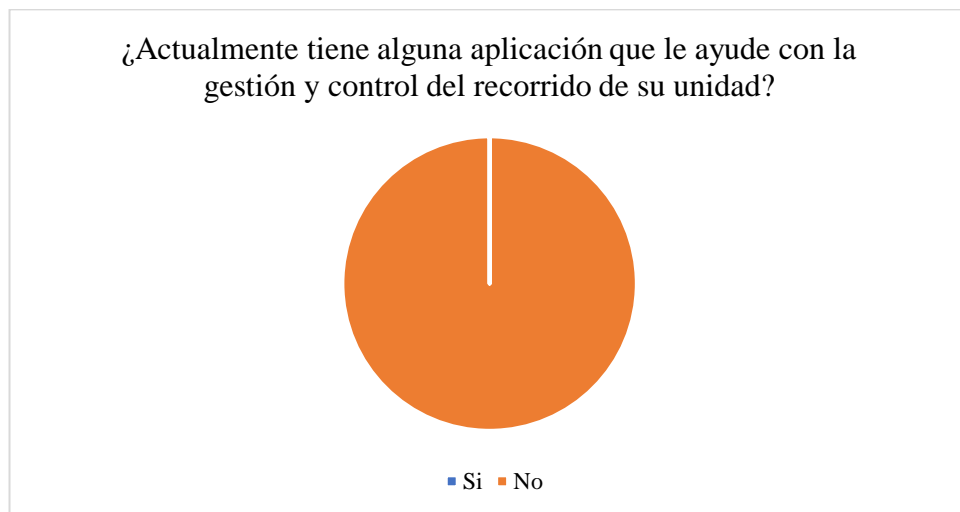
El aplicativo que se desarrolla permitirá a los conductores tener una gestión y control de su ruta, esto permitirá que el área de servicios laborales también tenga gestión y control de los recorridos

- **Pregunta 5**

**¿Actualmente tiene alguna aplicación que le ayude con la gestión y control del recorrido de su unidad?**

<b>Alternativas</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	0	0%
No	6	100%
Total	6	100%

*Tabla N.- 7 Pregunta 5 Encuesta  
Elaborado por: Franklin Lesano.*



*Figura 11 Pregunta 5  
Elaborado por: Franklin Lesano.*

**Análisis**

Como se puede observar en la Figura número 7 que la mayoría de los conductores no posee un aplicativo que le permita la gestión y control del recorrido.

**Interpretación**

El desarrollo de esta aplicación permitirá tener en tiempo real la ubicación y control de las unidades, siendo beneficioso tanto a los conductores como a la gestión de servicios laborales.



#### **2.2.4. Procesamiento y Análisis de Datos**

De acuerdo a la entrevista realizada al área de servicios laborales se puede sacar varias conclusiones que permitirán el desarrollo del proyecto de investigación como:

Es factible la realización de una aplicación móvil de seguimiento de los buses privados, ya que la empresa no cuenta con un sistema de control ni de gestión de rutas.

Con la implementación de un aplicativo móvil GPS a tiempo real tanto usuarios como conductores estarán comunicados de las rutas existentes y activas.

Desde la parte operativa se podrá gestionar de mejor manera las rutas asignando paradas que formaran la ruta para posteriormente ser enviadas a los pasajeros y conductores.

De la encuesta realizada a los conductores se puede concluir que al estar dispuesto a utilizar un aplicativo móvil para mejorar sus rutas el proyecto no tendrá ninguna restricción para su posterior implementación.

De la encuesta realizada al área de tecnología se puede concluir que la infraestructura que manejan actualmente se puede acoplar a la estructura para la implantación del sistema completo de seguimiento de buses a tiempo real

La tecnología que el área de tecnología maneja se base en casi su totalidad es de Microsoft por lo que si en una fase posterior se desea realizar mantenimientos o mejoras la curva de aprendizaje será mínima.

A demás por petición del jefe de tecnología se implementará a excepción de la base de datos en tecnología de Microsoft el aplicativo móvil y los webs services.

La información obtenida por medio de la aplicación de los instrumentos para la recolección de información se revisó e interpreto en una matriz de análisis en donde se establece la pregunta, la respuesta y la conclusión de la entrevista aplicada. En el caso de las encuestas los resultados se los describe bajo representaciones de frecuencia y porcentajes en tablas y gráficos, por lo que, se siguió el siguiente proceso:

- Elaboración de instrumentos.
- Aplicación de entrevista y encuesta

- Análisis de la entrevista.
- Tabulación e interpretación de la encuesta.
- Conclusiones de entrevista y encuesta.

En función a los datos obtenidos en las encuestas y las entrevistas se pudo determinar que la empresa maneja actualmente 5 rutas distribuidas en 8 horario, los cuales se detallan a continuación:

- Entrada de los operarios a las 06H00.
- Salida de los Operarios 06h30.
- Entrada del Personal Administrativo a las 08H30.
- Entrada de los Operarios a las 14H00.
- Salida de los Operarios a las 14H30.
- Salida del Personal Administrativo 17H30.
- Entrada de los Operarios a las 22H00.
- Salida de los Operarios 22H30.

Las rutas se las maneja a través de Google maps en la web en un computador, la asistente de servicio laboral marca los puntos y las paradas en el mapa, para luego imprimirlos, dichas impresiones son autorizadas y supervisadas por el área de servicios laborales, si existe alguna modificación, en una de las rutas debe pasar por la aprobación de Gerencia considerando el costo beneficio que este cambio implicaría.

El proceso de control de inicio del bus no está supervisado, pero mantienen informado a todo el personal operario y administrativo de donde inicia la ruta a través de paneles informativos ubicados estratégicamente en diferentes lugares, solo se registra la llegada de los buses a las afueras de la empresa.

La asistente de servicios laborales al no tener un control de inicio y llegada de los Buses es informada a través de la queja del personal, la cual llama al conductor de esa ruta para confirmar que no ha iniciado o no a completado la ruta, al no tener contratados conductores directamente, si no tiene contratado el servicio de recorrido, se procede a final del mes al descuento por incumplimiento de la ruta en el horario de ese día.

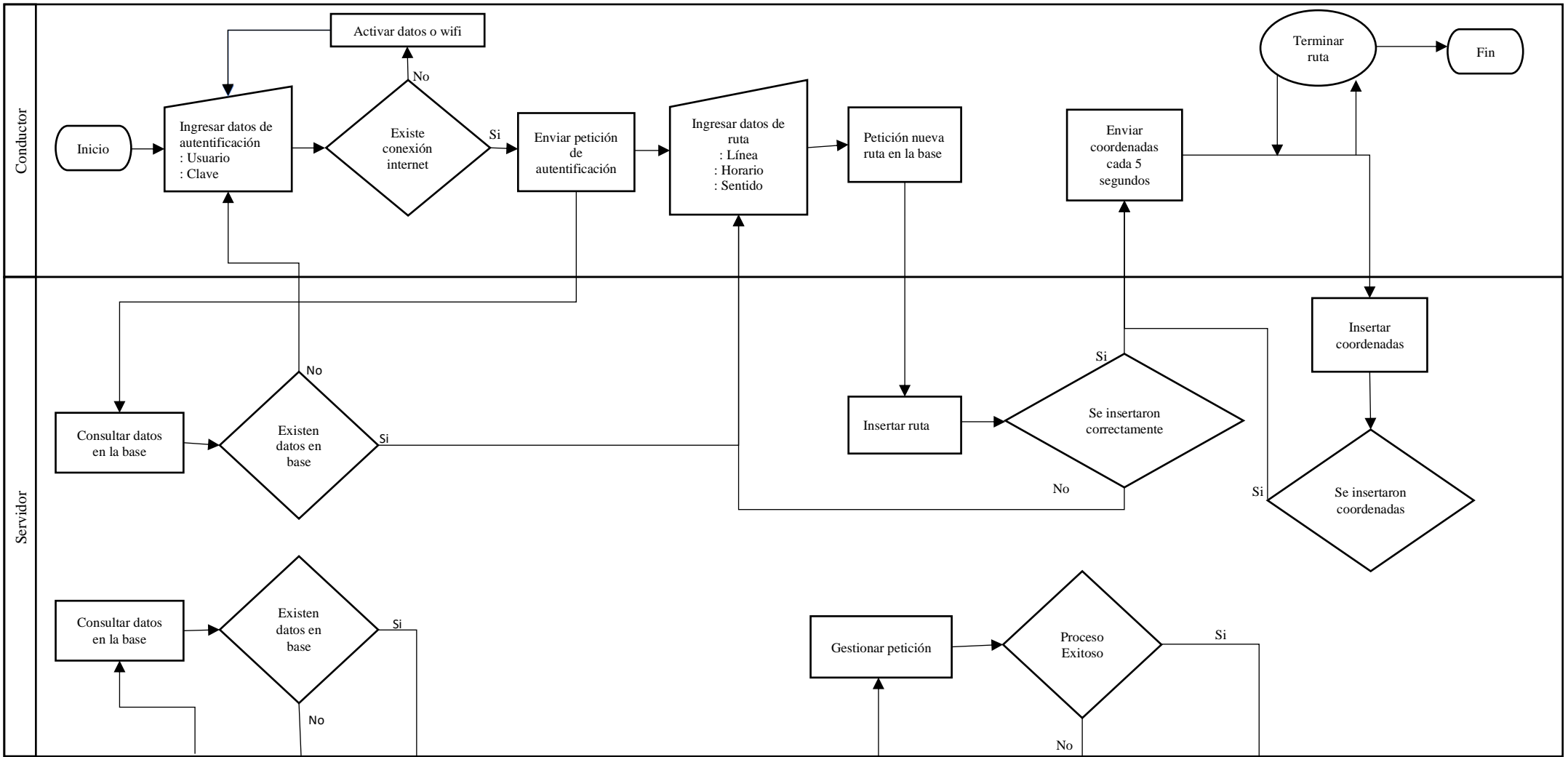
La arquitectura tecnológica que tiene implementada la empresa es adecuada para la implementación de los servicios de aplicación y de DDB, ya que sus servidores tienen sistema operativo Windows server 2012.

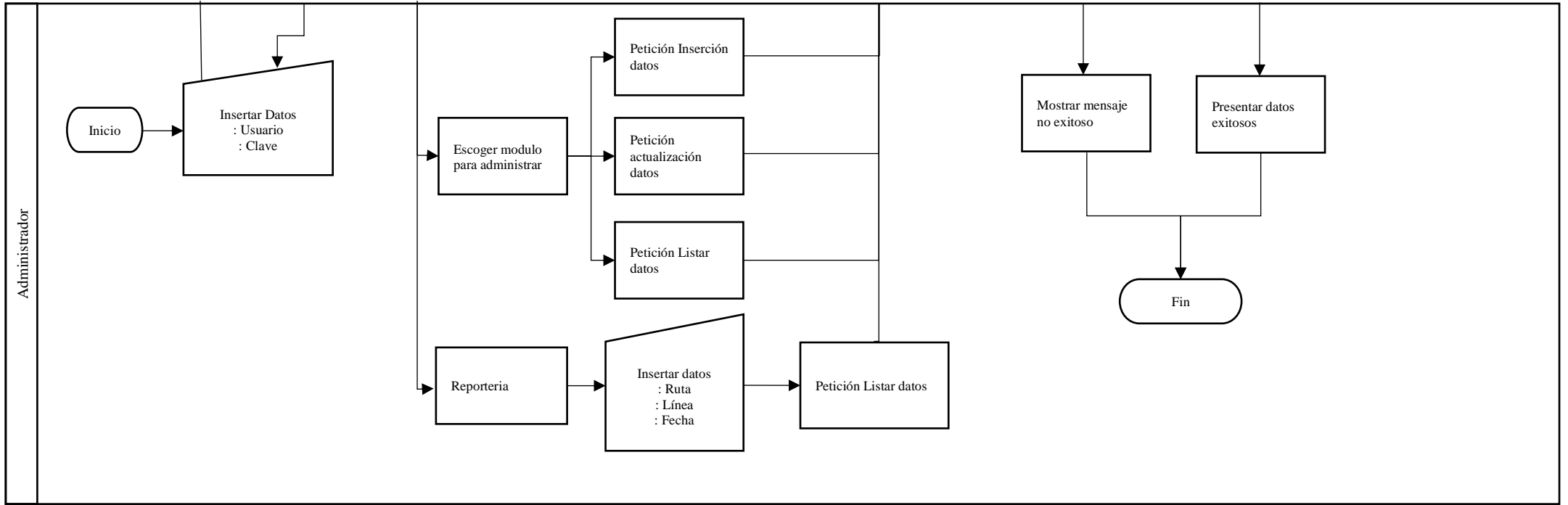
## **CAPITULO III.- RESULTADOS Y DISCUSIÓN**

### **3.1. Análisis y discusión de resultados**

#### **3.1.1. Análisis de los datos para el seguimiento en tiempo real de los recorridos de buses privados de Plasticaucho industrial.**

Diagrama de procesos, intervienen 3 principales actores necesarios para el desarrollo de geolocalización donde se iniciará con el conductor autenticándose en la aplicación para ser enviada la información al servidor y posteriormente ser coordinada del lado del personal operativo en el gestor de administración, con el fin de mantener actualizado los datos y tener una constante comunicación entre los actores.





**Figura 12** Análisis de los datos para el seguimiento en tiempo real de los recorridos de buses privados de Plasticaucho industrial.

*Elaborado por:* Franklin Lesano

El diagrama consta de 3 principales escenarios donde la autenticación al sistema lo realiza el conductor en la aplicación, internamente la aplicación deberá constatar el acceso a GPS del dispositivo móvil y mantener una conexión a internet, una vez que se valide los requerimientos el aplicativo deberá enviar una petición al servidor para guardar los datos de las coordenadas obtenidas, inmediatamente el servidor deberá responder con una mensaje de afirmación o negación, el dispositivo móvil debería enviar los datos cada cierto tiempo para ser guardadas en la base de datos esto con el fin de observar el recorrido del bus en tiempo real.

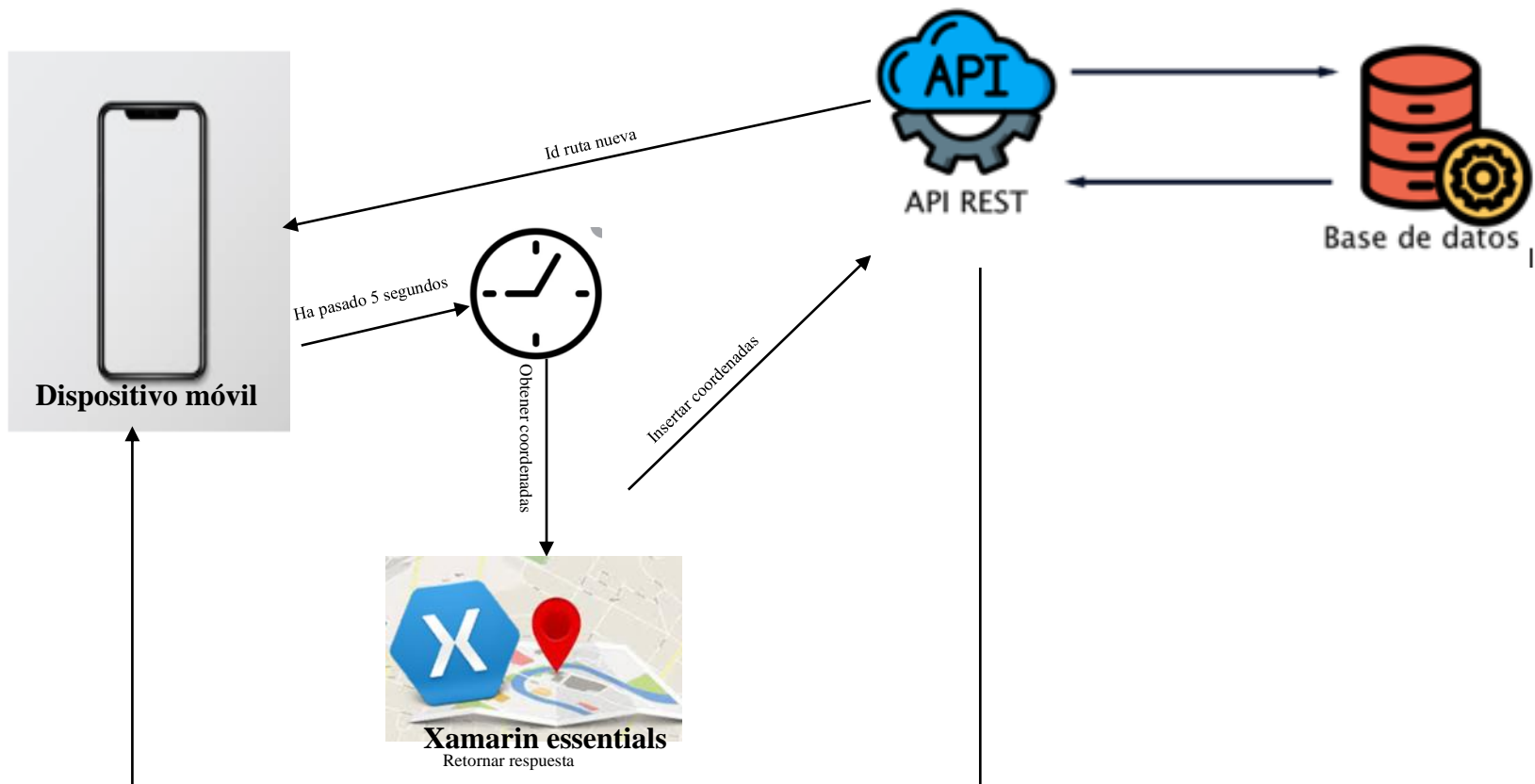
### **3.1.2. Análisis del proceso de seguimiento GPS del dispositivo móvil**

El aplicativo móvil será desarrollado en Xamarin forms que es una herramienta multiplataforma de Microsoft que permite desarrollar aplicaciones en código C# y compilar en código nativo para IOS y Android

En obtención de las coordenadas geográficas se requiere un plugin que permita utilizar el GPS interno del dispositivo móvil para lo cual Xamarin essentials que está desarrollado para obtener y utilizar los recursos del dispositivo móvil es ideal y estable en Xamarin forms.

La comunicación entre el dispositivo móvil y el servidor se deberá realizar a través de la de las webs services las cuales deberán funcionar como intermediarias para obtener y extraer información de la base de datos en servidor.

El diagrama de seguimiento con el GPS muestra la comunicación que se deberá establecerse entre el dispositivo móvil y el servidor para que las coordenadas sean almacenadas posteriormente en la base de datos con el fin de ser recuperadas en lo posterior para su seguimiento o administración.



*Figura 13 Estructura de geolocalización.  
Elaborado por: Franklin Lesano.*



Como se puede apreciar en la figura 13, para el seguimiento de los buses en tiempo real el aplicativo deberá enviar las coordenadas cada cierto tiempo para ello se debería utilizar un hilo de procesos el cual envíe constantemente la información, dentro del aplicativo este debería esperar que pasen 5 segundos y realizar una nueva petición de coordenadas a través de Xamarin essentials, luego se deben insertar las coordenadas en la base de datos mediante las Webs Apis y de la misma manera deberán regresar la respuesta del servidor mediante estas, de esta forma estableciendo el bucle de seguimiento a través del GPS interno del dispositivo móvil.

### **3.1.3. Análisis Comparativo de la Metodología Agiles.**

En este punto se procederá a realizar la comparación de las metodologías para poder escoger a la cual represente una mayor factibilidad para nuestra investigación, las 3 metodologías que se consideraron son: Metodología XP, Metodología Scrum y por último la Metodología Kanban. Los campos que se tomaron en cuenta para evaluar las dichas metodologías son: Los tipos de proyectos para las cuales son aptas las metodologías, los participantes que se pueden abarcar, el modelo de desarrollo que poseen, la duración del ciclo de interacción, cual es el rol que ocupa el usuario, como se presenta la documentación, las prácticas principales que tiene, si es que tienen un desarrollo de las características de una forma simultánea, los principios que presentan, después se destacan las ventajas que poseen y por consiguiente sus desventajas.

## Metodologías Ágiles

	<b>Metodología XP</b>	<b>Metodología Scrum</b>	<b>Metodología Kanban</b>
<b>Tipos de proyectos</b>	Pequeños y Medianos	Pequeños, medianos y grandes	Pequeños y medianos
<b>Número de participantes</b>	10 o menos	Varios equipos menores a 10	Varios equipos menores a 10
<b>Modelo de desarrollo</b>	Iterativo y rápido	Iterativo y rápido	Iterativo y rápido
<b>Período del ciclo de iteración</b>	1-6 semanas	2 – 4 semanas	1-6 semanas
<b>Participación de Usuario</b>	Activamente involucrado	A través del propietario del producto	Activamente involucrado
<b>Documentación</b>	Documentación básica	Documentación básica	Documentación básica
<b>Prácticas principales</b>	Simplicidad, programación en pares desarrollo basado en pruebas	Reuniones de scrum	
<b>Desarrollo de características simultáneas</b>	Posible	Posible	Posible
<b>Principios</b>	Retroalimentación, asumir la simplicidad y aceptar el cambio.	Conjunto de roles, responsabilidades y reuniones que nunca cambian.	Visualizar el flujo de trabajo, limitar el trabajo en curso, administrar y mejorar el flujo, hacer políticas explícitas y mejorar continuamente.
<b>Ventajas</b>	Códigos simples, fáciles de mejorar. Todo el ciclo de XP es visible Las pruebas constantes agilizan el proceso Potenciando el talento de los equipos.	Más transparencia y visibilidad del proyecto. Mayor responsabilidad del equipo. Fácil de adaptarse a los cambios. Mayor ahorro de costos.	Aumenta la flexibilidad. Reduce el desperdicio. Fácil de entender. Mejora el flujo de entrega. Minimiza el tiempo del ciclo.

<b>Desventajas</b>	Más atención al código y menos al diseño. No trabajar en equipos de trabajo remoto Si los problemas no se documentan correctamente, posibilidad de repetir los mismos problemas.	Riesgo de aumento del alcance. El equipo requiere experiencia y compromiso. Las tareas mal definidas pueden generar imprecisiones.	El tablero desactualizado puede generar problemas. Falta de tiempo.
--------------------	--	--	--

*Tabla N.- 8 Análisis Comparativo de la metodología ágiles.  
Elaborado por: Franklin Lesano*

### **3.1.3.1. Metodología seleccionada**

La metodología XP fue la metodología seleccionada gracias a los resultados que se reflejan en la Tabla N-. 8, las razones son: es flexible y da la facilidad de modelado, mejorando tiempos calidad, facilidad, es importante minimizar los errores durante el desarrollo y la metodología con sus buenas prácticas la vuelven rápida, dinámica. Lo que cabe destacar es que el usuario va a estar involucrado en el proceso y donde todo el ciclo será visible. Aunque el diseño es algo importante el diseño, es decir, el cómo se visualizará el aplicativo, es de mucha más prioridad prestar atención a los códigos para que no se tenga un error, donde los mismos códigos están en la posibilidad de ser mejorados. La metodología permite hacer pruebas al final de cada sprint lo que da como resultado la corrección de errores en menos tiempo y buscando la mejor solución, y no depender de una prueba final y tener que cambiar todo el proyecto.

### **3.1.4 Gestores de Base de Datos**

#### **3.1.4.1. PostgreSQL**

##### **3.1.4.1.1. Características**

PostgreSQL es un potente gestor de base de datos relacional de código abierto que utiliza el lenguaje SQL combinado con muchas características, almacenan y escalan las cargas de trabajo de datos más complicadas. Los inicios de PostgreSQL son alrededor del año 1986 como parte del proyecto POSTGRES de la Universidad de California en Berkeley.

PostgreSQL es gratuito y de código abierto, puede definir sus propios tipos de datos, crear funciones personalizadas estas características lo hace altamente sensible e incluso escribir código de diferentes lenguajes de programación [30].

#### **3.1.4.2. MySQL**

##### **3.1.4.2.1 Características**

MySql cuenta con una licencia libre ideal para usuarios que inician, sus características son similares a PostgreSQL, una de sus características es que es un gestor de base de datos muy rápido para proyectos Web.

Fue desarrolla originalmente para manejar bases de datos extensas y ser rápido y eficiente, Se ha utilizado con éxito en entornos de producción siendo estable y eficiente, como se había

mencionado su licencia y uso son libres y no comerciales, hay que aclarar que MySQL es un gestor de base datos relacional y utiliza lenguaje SQL, su lanzamiento sucedió en 1995 [32].

### **3.1.5. Análisis Comparativo de los Gestores de Base de Datos**

Uno de los principales motivos que se decidió hacer la comparativa entre las bases de datos PostgreSQL y MySQL, fue fundamentada en varios puntos esenciales que se hicieron notar indispensables para seguir con el desarrollo del proyecto.

Entonces, los puntos que se tomaron en cuenta, son: la seguridad que ofrece cada gestor de datos, el tipo de licencia que tiene, los tipos de datos que se encuentran presentes y se pueden llegar a tener, la seguridad que menciona con referencia de los datos, el alcance que tiene de espacio para los datos, es decir, su almacenamiento, el máximo de campos que se puede llegar a tener por tabla, las ventajas y por ende las desventajas que se destacan en cada una

**Gestores de la Base de datos**

	<b>PostgreSQL</b>	<b>MySQL</b>
<b>Seguridad</b>	<ul style="list-style-type: none"> <li>• Autenticación: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificado y más.</li> <li>• Robusto sistema de control de acceso.</li> <li>• Seguridad a nivel de columna y fila.</li> <li>• Autenticación multifactor con certificados y un método adicional.</li> </ul>	<ul style="list-style-type: none"> <li>• Encriptación y Desencriptación Base64 (Advanced Encryption Standard)</li> <li>• Fichero de configuración</li> <li>• Cuentas de usuario</li> <li>• Acceso remoto</li> </ul>
<b>Tipo Licencia</b>	<ul style="list-style-type: none"> <li>• Completamente de software libre y con una licencia BSD, compatible con cualquier uso, ya sea personal o comercial</li> </ul>	<p>MySQL tiene licencia dual:</p> <ul style="list-style-type: none"> <li>• MySQL se distribuye bajo la licencia GPL</li> <li>• MySQL Commercial License for OEMs</li> </ul>
<b>Tipos Datos</b>	<ul style="list-style-type: none"> <li>• Primitives: Integer, Numeric, String, Boolean.</li> <li>• Structured: Date/Time, Array, Range / UUID.</li> <li>• Document: JSON/JSONB, XML,</li> <li>• Geometry: Point, Line, Circle, Polygon.</li> <li>• Customizations: Composite, Custom Types.</li> </ul>	<ul style="list-style-type: none"> <li>• Datos numéricos: mediumint, int o integer, bigint.</li> <li>• Datos alfanuméricos: char, varchar, mediumblob, binary, varbinary text, longtext, enum, set</li> </ul>
<b>Seguridad de datos</b>	<ul style="list-style-type: none"> <li>• Unique, not null.</li> <li>• Primary Keys.</li> <li>• Foreign Keys.</li> <li>• Exclusion Constraints.</li> <li>• Explicit Locks, Advisory Locks.</li> </ul>	<ul style="list-style-type: none"> <li>• Null</li> <li>• default</li> <li>• Binary</li> <li>• Ide</li> <li>• Primary Key</li> <li>• Auto_increment</li> </ul>
<b>Tamaño de la base de datos</b>	<ul style="list-style-type: none"> <li>• Ilimitado</li> </ul>	<ul style="list-style-type: none"> <li>• Ilimitado</li> </ul>
<b>Máximo de campos por tabla</b>	<ul style="list-style-type: none"> <li>• Ilimitado</li> </ul>	<ul style="list-style-type: none"> <li>• Depende del sistema de archivos (64 TB)</li> </ul>

<b>Ventajas</b>	<ul style="list-style-type: none"> <li>• Ideal para proyectos web.</li> <li>• Fácil de administrar.</li> <li>• Sintaxis SQL es fácil de aprender y estandarizado.</li> <li>• Soporta varias plataformas.</li> </ul>	<ul style="list-style-type: none"> <li>• Gran velocidad para proyectos de internet, ideal para ambientes web.</li> <li>• Fácil uso para usuario novatos.</li> <li>• Sólo los usuarios con acceso pueden ver la información de las tablas, los demás pueden acceder a través de vistas.</li> </ul>
<b>Desventajas</b>	<ul style="list-style-type: none"> <li>• No porta Tablas paces, esto para versiones inferiores 9.0.</li> <li>• La orientación a Objetos es una extensión de la herencia, no porta por completo.</li> </ul>	<ul style="list-style-type: none"> <li>• Al poblar podría tener problemas con vistas complejas que consuman tiempo y recursos. Puede causar un cuello de botella durante la transmisión y/o resolución de consultas.</li> </ul>
<b>Licencia</b>	<ul style="list-style-type: none"> <li>• GPLv2 o propietario</li> </ul>	<ul style="list-style-type: none"> <li>• Licencia PostgreSQL</li> </ul>
<b>Seguridad</b>	<ul style="list-style-type: none"> <li>• Soporte TLS</li> </ul>	<ul style="list-style-type: none"> <li>• Soporte SSL nativo</li> </ul>
<b>Replicación</b>	<ul style="list-style-type: none"> <li>• Síncrono unidireccional</li> </ul>	<ul style="list-style-type: none"> <li>• Síncronico</li> </ul>

*Tabla N.- 9 Tabla de Comparación de los gestores de base de datos*

**Elaborado por:** Franklin Lesano

### 3.1.5.1. Gestor de Base de Datos seleccionada

Las razones que se destacaron en PostgreSQL para ser seleccionada son: la licencia libre, al tener licencia GPL y ser uno de los primeros gestores de base de datos más descargados del mundo también es uno de los más rápidos y seguros del mundo, otro de los factores es la gran escalabilidad que tiene y la facilidad de uso que tiene lo hace ideal para sistemas web y sistemas de producción además de ello el manejo no solo de variables primitivas si no de otras variables así como crear el variables tipo objetos que son necesarias para este tipo de proyectos. Y algo a destacar es su base de datos ilimitada y los campos por tabla ilimitados que ofrece.

## 3.2. Desarrollo de la Propuesta

### 3.2.1. Fase 1: Panificación

Con las entrevistas y la encuesta aplicada, se ejecutó la planificación comenzando por el modelado y el diseño de la base de datos, a continuación, se realizará el desarrollo del web APIs REST, por último, se realizará la implementación del y desarrollo del aplicativo para los conductores y para los pasajeros.

#### 3.2.1.1. Levantamiento de la información

Con la entrevista realizada en servicios laborales se pudo conocer que el proceso de control de los buses los lleva de forma manual por lo que este proceso se puede automatizar de esta forma el personal administrativo tendrá más control sobre el inicio y finalización de sus rutas.

La entrevista y la interacción en el campo se pudo aclarar otros aspectos de los requerimientos para este tipo de proyectos como la arquitectura tecnológica que posee la empresa, numero de buses interacción con los pasajeros permitiendo hacer preguntas abiertas hacia el cliente para cubrir todos los aspectos posibles.

#### 3.2.1.2. Definición de roles

Nombre	Rol	Descripción
Ing. Carlos Núñez	Scrum Máster.	Persona que supervisa mensualmente lo que se lleva a cabo y su finalización de



		acuerdo con el cronograma de actividades establecidos.
Ing. Mery López	Propietario del producto.	Persona que necesita el sistema GPS para control del personal.
Lesano Pérez Franklin Vicente	Equipo de desarrollo.	Persona con los conocimientos para el desarrollo y programación del código, se encargará de las pruebas del sistema.
Lic. Roció Vela	El cliente.	Persona que se encargara del manejo y gestión de todo el sistema.

*Tabla N.- 10 Definición de roles  
Elaborado por: Franklin Lesano.*

### 3.2.1.3. Historias de usuario

Las historias de usuario en XP se podrían expresar como requisitos que se expresa en lenguaje común para que sean entendibles por el usuario para luego ser transformadas en tareas para el desarrollo y codificación en el proyecto.

<b>Historias de usuario</b>			
Numero:	001	Usuario:	Desarrollador
Nombre de la historia:	Análisis de la base de datos		
Prioridad:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	5	Iteración	1
Programador responsable:	Franklin Vicente Lesano Pérez		
Descripción: Se requiere analizar en base a la entrevista los actores que intervendrán para generar las tablas.			
Observación: Determinar los atributos para cada tabla.			

*Tabla N.- 11 Historias de usuario 001  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	002	Usuario:	Desarrollador
Nombre de la historia:	Modelamiento y desarrollo de la base de datos		
Prioridad:	Alta	Riesgo de desarrollo:	medio
Puntos estimados:	5	Iteración	1

Programador responsable:	Franklin Vicente Lesano Pérez.
Descripción: Modelamiento de la base de datos en base a la entrevista.	
Observación: Se realizará un script para la base de datos y se revisará antes de dar paso.	

Tabla N.- 12 Historias de usuario 002

Elaborado por: Franklin Lesano.

Historias de usuario			
Numero:	003	Usuario:	Desarrollador
Nombre de la historia:	implementación en un servidor de pruebas		
Prioridad:	Alta	Riesgo de desarrollo:	media
Puntos estimados:	4	Iteración	1
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se buscará un sitio gratuito para almacenar la base de datos con salida a internet para pruebas en el futuro.			
Observación: El sitio para almacenar se la base debe soportar PostgreSQL, ya que es el gestor que se utilizara como gestor de base.			

Tabla N.- 13 Historias de usuario 003

Elaborado por: Franklin Lesano.

Historias de usuario			
Numero:	004	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API para tabla Conductor		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	5	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe desarrollar un API REST para métodos GET, POS, PUT, DELETE para interacción con el CRUD y el APP del conductor y cliente.			
Observación: Al no se datos sensibles no es necesario implementar métodos de seguridad para Cifrar la comunicación.			

Tabla N.- 14 Historias de usuario 004

Elaborado por: Franklin Lesano.

<b>Historias de usuario</b>			
Numero:	005	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API para tabla Buses		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	4	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe desarrollar un API REST de la tabla Buses para métodos GET, POS, PUT, DELETE para interacción con el CRUD y el APP del conductor y cliente.			
Observación: Al no se datos sensibles no es necesario implementar métodos de seguridad para Cifrar la comunicación.			

*Tabla N.- 15 Historias de usuario 005  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	006	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API para tabla Línea		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	4	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe desarrollar un API REST de la tabla línea para métodos GET, POS, PUT, DELETE para interacción con el CRUD y el APP del conductor y cliente.			
Observación: Al no se datos sensibles no es necesario implementar métodos de seguridad para Cifrar la comunicación.			

*Tabla N.- 16 Historias de usuario 006  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	007	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API para tabla parada		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	4	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		

Descripción: Se debe desarrollar un API REST de la tabla parada para métodos GET, POS, PUT, DELETE para interacción con el CRUD y el APP del conductor y cliente.
Observación: Al no se datos sensibles no es necesario implementar métodos de seguridad para Cifrar la comunicación.

*Tabla N.- 17 Historias de usuario 007  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	008	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API para tabla ruta		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	4	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe desarrollar un API REST de la tabla ruta para métodos GET, POS, PUT, DELETE para interacción con el CRUD y el APP del conductor y cliente.			
Observación: Al no se datos sensibles no es necesario implementar métodos de seguridad para Cifrar la comunicación.			

*Tabla N.- 18 Historias de usuario 008  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	009	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API para tabla Recorrido		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	4	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe desarrollar un API REST de la tabla recorrido para métodos GET, POS, PUT, DELETE para interacción con el CRUD y el APP del conductor y cliente.			
Observación: Al no se datos sensibles no es necesario implementar métodos de seguridad para cifrar la comunicación.			

*Tabla N.- 19 Historias de usuario 009  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	010	Usuario:	Desarrollador
Nombre de la historia:	Desarrollo del API Notificaciones		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	3	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe desarrollar un API REST para registro de las keys que genera la API de FireBase, y envío de notificaciones Push.			
Observación: Se debe implementar el APK de FireBase para notificaciones Push a los dispositivos de los clientes.			

*Tabla N.- 20 Historias de usuario 010  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	011	Usuario:	Desarrollador
Nombre de la historia:	Subir las APIS a un entorno de pruebas		
Prioridad:	Alta	Riesgo de desarrollo:	alta
Puntos estimados:	3	Iteración	2
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: Se debe encontrar un servidor gratuito para alojar el servicio de las API REST para luego ser utilizado con el CRUD, App del conductor y del cliente.			
Observación: El servidor debe tener salida a internet para conectividad con las APS se debe tomar en cuenta que no debe tener certificado SSL.			

*Tabla N.- 21 Historias de usuario 011  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	012	Usuario:	Desarrollador
Nombre de la historia:	Pantalla Login Conductor		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	3

Programador responsable:	Franklin Vicente Lesano Pérez.
Descripción:	Se debe pedir el usuario y la clave para que el sistema autentifique los datos en la base y permita el acceso.
Observación:	Se debe considerar un mensaje de alerta si no consta en la base o si ocurre algún error al tratar de autenticarse.

*Tabla N.- 22 Historias de usuario 012  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	013	Usuario:	Desarrollador
Nombre de la historia:	Pantalla Datos para inicial la ruta		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	3
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción:	Luego de la pantalla de autenticación el conductor debe escoger la unidad que va a cubrir la ruta la línea que va a manejar y el horario asignado, si falta algún dato se debe generar un mensaje de alerta.		
Observación:	Se debe considerar un mensaje de alerta si no consta en la base o si ocurre algún error al tratar de autenticarse.		

*Tabla N.- 23 Historias de usuario 013  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	014	Usuario:	Desarrollador
Nombre de la historia:	Pantalla de guía de navegación del conductor		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	3
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción:	La pantalla debe mostrar datos como el nombre del conductor, la ruta, la parada actual y la parada siguiente.		

Observación: Se debe mostrar visualmente la parada siguiente que debe tomar, se debe enviar un mensaje si llega a la parada asignada, en caso de algún incidente se debe notificar a todos los usuarios

*Tabla N.- 24 Historias de usuario 014  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	015	Usuario:	Desarrollador
Nombre de la historia:	Pantalla seguimiento de ruta cliente		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	3
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: La pantalla de seguimiento de la ruta para los clientes para visualizar el sitio del bus en el mapa de Google maps			
Observación: Se debe escoger la línea, estado y horario para visualizar el recorrido del bus en el mapa			

*Tabla N.- 25 Historias de usuario 015  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	016	Usuario:	Desarrollador
Nombre de la historia:	CRUD Conductor		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	4
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: La pantalla ingreso, actualización, consulta de los datos de conductores.			
Observación: Se debe tomar en cuenta que no se debe borrar para tener registro de los históricos de los conductores.			

*Tabla N.- 26 Historias de usuario 016  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	017	Usuario:	Desarrollador
Nombre de la historia:	CRUD Bu		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	4
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: La pantalla ingreso, actualización, consulta de los datos de Bus.			
Observación: Se debe tomar en cuenta que no se debe borrar para tener registro de los históricos de los Buses.			

*Tabla N.- 27 Historias de usuario 017  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	018	Usuario:	Desarrollador
Nombre de la historia:	CRUD líneas		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	4
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: La pantalla ingreso, actualización, consulta de los datos de líneas.			
Observación: Se debe tomar en cuenta que no se debe borrar para tener registro de los históricos de las líneas.			

*Tabla N.- 28 Historias de usuario 018  
Elaborado por: Franklin Lesano.*

<b>Historias de usuario</b>			
Numero:	019	Usuario:	Desarrollador
Nombre de la historia:	CRUD Paradas		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	4
Programador responsable:	Franklin Vicente Lesano Pérez.		



Descripción: La pantalla ingreso, actualización, consulta de los datos de Paradas.
Observación: Se debe tomar en cuenta que no se debe borrar para tener registro de los históricos de los Paradas.

Tabla N.- 29 Historias de usuario 019  
Elaborado por: Franklin Lesano.

<b>Historias de usuario</b>			
Numero:	020	Usuario:	Desarrollador
Nombre de la historia:	Reportes recorridos.		
Prioridad:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	4	Iteración	4
Programador responsable:	Franklin Vicente Lesano Pérez.		
Descripción: La pantalla debe generar una vista en de las rutas de los buses			
Observación: Se debe filtrar por fecha las rutas de los buses comalidas una vez tomado los datos se debe dibujar en Google maps.			

Tabla N.- 30 Historias de usuario 020  
Elaborado por: Franklin Lesano.

### 3.2.1.4. Tareas

Las tareas representan un conjunto de actividades, actividades similares con un comienzo y un final, las tareas son asignadas por el cliente y supervisadas paralas entrega.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T001	Código Historia: 001
Tipo de Tarea: Análisis	Puntos estimados: 1
Nombre Tarea: Análisis de la base de datos	
Fecha de Inicio: 08/11/2021	Fecha Fin: 09/11/2021
Descripción: Análisis de la base de datos para determinar nombre de las tablas y los atributos	

Tabla N.- 31 Tarea 001  
Elaborado por: Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
--------------	---

N° Tareas: T002	Código Historia: 002
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Modelamiento de la base de datos	
Fecha de Inicio: 09/11/2021	Fecha Fin: 11/11/2021
Descripción: Modelamiento de la base de datos en la PostgreSQL.	

*Tabla N.- 32 Tarea 002*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T003	Código Historia: 003
Tipo de Tarea: Desarrollo	Puntos estimados: 3
Nombre Tarea: Implementar la base de datos en un servidor Elephantsql.	
Fecha de Inicio: 11/11/2021	Fecha Fin: 16/11/2021
Descripción: Subir la base a un servidor gratuito que soporte Postgresql	

*Tabla N.- 33 Tarea 003*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T004	Código Historia: 004
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Conductor método PUT	
Fecha de Inicio: 16/11/2021	Fecha Fin: 17/11/2021
Descripción: creación del método PUT para la tabla Conductor	

*Tabla N.- 34 Tarea 004*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T005	Código Historia: 004
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Conductor método GET	
Fecha de Inicio: 17/11/2021	Fecha Fin: 18/11/2021
Descripción: Creación del método GET para la tabla Conductor	

*Tabla N.- 35 Tarea 005*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
--------------	---

N° Tareas: T006	Código Historia: 004
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Conductor método POST	
Fecha de Inicio: 18/11/2021	Fecha Fin: 19/11/2021
Descripción: Creación del método POST para la tabla Conductor	

*Tabla N.- 36 Tarea 006  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T007	Código Historia: 004
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Conductor método DELETE	
Fecha de Inicio: 22/11/2021	Fecha Fin: 23/11/2021
Descripción: Creación del método DELETE para la tabla Conductor	

*Tabla N.- 37 Tarea 007  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T008	Código Historia: 005
Tipo de Tarea: Análisis	Puntos estimados: 1
Nombre Tarea: Tabla Buses método GET	
Fecha de Inicio: 27/11/2021	Fecha Fin: 28/11/2021
Descripción: Creación del método GET para la tabla Buses	

*Tabla N.- 38 Tarea 008  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T009	Código Historia: 005
Tipo de Tarea: Análisis	Puntos estimados: 1
Nombre Tarea: Tabla Buses método POST	
Fecha de Inicio: 28/11/2021	Fecha Fin: 29/11/2021
Descripción: Creación del método POST para la tabla Buses	

*Tabla N.- 39 Tarea 009  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
--------------	---

N° Tareas: T010	Código Historia: 005
Tipo de Tarea: Análisis	Puntos estimados: 1
Nombre Tarea: Tabla Buses método PUT	
Fecha de Inicio: 29/11/2021	Fecha Fin: 30/11/2021
Descripción: Creación del método PUT para la tabla Buses	

*Tabla N.- 40 Tarea 010*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T011	Código Historia: 005
Tipo de Tarea: Análisis	Puntos estimados: 1
Nombre Tarea: Tabla Buses método DELETE	
Fecha de Inicio: 30/11/2021	Fecha Fin: 31/11/2021
Descripción: Creación del método DELETE para la tabla Buses	

*Tabla N.- 41 Tarea 011*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T012	Código Historia: 006
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla línea método GET	
Fecha de Inicio: 31/11/2021	Fecha Fin: 01/12/2021
Descripción: Creación del método GET para la tabla línea	

*Tabla N.- 42 Tarea 012*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T013	Código Historia: 006
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla línea método PUT	
Fecha de Inicio: 03/12/2021	Fecha Fin: 04/12/2021
Descripción: Creación del método PUT para la tabla línea	

*Tabla N.- 43 Tarea 013*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T014	Código Historia: 006
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla línea método POST	
Fecha de Inicio: 06/12/2021	Fecha Fin: 07/12/2021
Descripción: Creación del método POST para la tabla línea	

*Tabla N.- 44 Tarea 014  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T015	Código Historia: 006
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla línea método DELETE	
Fecha de Inicio: 07/12/2021	Fecha Fin: 08/12/2021
Descripción: Creación del método DELETE para la tabla línea	

*Tabla N.- 45 Tarea 015  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T016	Código Historia: 007
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Paradas método PUT	
Fecha de Inicio: 08/12/2021	Fecha Fin: 09/12/2021
Descripción: Creación del método PUT para la tabla Paradas	

*Tabla N.- 46 Tarea 016  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T017	Código Historia: 007
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Paradas método POST	
Fecha de Inicio: 09/12/2021	Fecha Fin: 10/12/2021
Descripción: Creación del método POST para la tabla Paradas	

*Tabla N.- 47 Tarea 017  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T018	Código Historia: 007
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Paradas método GET	
Fecha de Inicio: 13/12/2021	Fecha Fin: 14/12/2021
Descripción: Creación del método GET para la tabla Paradas	

*Tabla N.- 48 Tarea 018  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T019	Código Historia: 007
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Paradas método DELETE	
Fecha de Inicio: 14/12/2021	Fecha Fin: 15/12/2021
Descripción: Creación del método DELETE para la tabla Paradas	

*Tabla N.- 49 Tarea 019  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T020	Código Historia: 008
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Rutas método PUT	
Fecha de Inicio: 15/12/2021	Fecha Fin: 16/12/2021
Descripción: Creación del método PUT para la tabla Rutas	

*Tabla N.- 50 Tarea 020  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T021	Código Historia: 008
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Rutas método POST	
Fecha de Inicio: 16/12/2021	Fecha Fin: 17/12/2021
Descripción: Creación del método POST para la tabla Rutas	

*Tabla N.- 51 Tarea 021  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T022	Código Historia: 008
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Rutas método GET	
Fecha de Inicio: 20/12/2021	Fecha Fin: 21/12/2021
Descripción: Creación del método GET para la tabla Rutas	

*Tabla N.- 52 Tarea 022*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T023	Código Historia: 008
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Rutas método DELETE	
Fecha de Inicio: 21/12/2021	Fecha Fin: 22/12/2021
Descripción: Creación del método DELETE para la tabla Rutas	

*Tabla N.- 53 Tarea 023*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T024	Código Historia: 009
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Recorrido método PUT	
Fecha de Inicio: 21/12/2021	Fecha Fin: 22/12/2021
Descripción: Creación del método PUT para la tabla Recorrido	

*Tabla N.- 54 Tarea 024*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T025	Código Historia: 009
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Recorrido método POST	
Fecha de Inicio: 22/12/2021	Fecha Fin: 23/12/2021
Descripción: Creación del método POST para la tabla Recorrido	

*Tabla N.- 55 Tarea 025*  
**Elaborado por:** Franklin Lesano.

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T026	Código Historia: 009
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Recorrido método GET	
Fecha de Inicio: 28/12/2021	Fecha Fin: 29/12/2021
Descripción: Creación del método GET para la tabla Recorrido	

*Tabla N.- 56 Tarea 026  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T027	Código Historia: 009
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Nombre Tarea: Tabla Recorrido método DELETE	
Fecha de Inicio: 29/12/2021	Fecha Fin: 30/12/2021
Descripción: Creación del método DELETE para la tabla Recorrido	

*Tabla N.- 57 Tarea 027  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T028	Código Historia: 010
Tipo de Tarea: Desarrollo	Puntos estimados: 10
Nombre Tarea: APIs Notificaciones	
Fecha de Inicio: 03/01/2022	Fecha Fin: 14/01/2022
Descripción: Investigación he implementación para enviar notificaciones push a través de una web service.	

*Tabla N.- 58 Tarea 028  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T029	Código Historia: 011
Tipo de Tarea: Desarrollo	Puntos estimados: 3
Nombre Tarea: Subir un entorno de pruebas	



Fecha de Inicio: 17/01/2022	Fecha Fin: 19/01/2022
Descripción: Subir y comprobar que las Apis estén funcionales en el entorno de pruebas Somee	

*Tabla N.- 59 Tarea 029  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T030	Código Historia: 012
Tipo de Tarea: Desarrollo	Puntos estimados: 6
Nombre Tarea: Crear la pantalla de autenticación	
Fecha de Inicio: 19/01/2022	Fecha Fin: 27/01/2022
Descripción: Crear la lógica para autenticarse en la base de datos y realizar el diseño de Frond-End.	

*Tabla N.- 60 Tarea 030  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T031	Código Historia: 013
Tipo de Tarea: Desarrollo	Puntos estimados: 6
Nombre Tarea: Pantalla Datos para iniciar la ruta	
Fecha de Inicio: 28/01/2022	Fecha Fin: 04/02/2022
Descripción: Crear la lógica para traer los datos de las líneas, buses, horarios para que pueda iniciar una nueva ruta.	

*Tabla N.- 61 Tarea 031  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T032	Código Historia: 014
Tipo de Tarea: Desarrollo	Puntos estimados: 5
Nombre Tarea: Pantalla de guía de navegación del conductor	
Fecha de Inicio: 07/02/2022	Fecha Fin: 11/02/2022
Descripción: Diseño de la pantalla de recorrido del bus, se debe aplicar las notificaciones de las paradas siguientes y de las paradas actuales.	

*Tabla N.- 62 Tarea 032  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T033	Código Historia: 014
Tipo de Tarea: Desarrollo	Puntos estimados: 5
Nombre Tarea: Aplicación de geocercas para el control deparadas	
Fecha de Inicio: 14/02/2022	Fecha Fin: 18/02/2022
Descripción: Se debe aplicar un nuget o librería para formar geocercas y tener control de las paradas que visita.	

*Tabla N.- 63 Tarea 033  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T034	Código Historia: 014
Tipo de Tarea: Desarrollo	Puntos estimados: 5
Nombre Tarea: Enviar notificaciones push	
Fecha de Inicio: 14/02/2022	Fecha Fin: 18/02/2022
Descripción: Se debe construir la lógica en el envío de notificaciones push para el envío de incidentes a los usuarios del sistema	

*Tabla N.- 64 Tarea 034  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T035	Código Historia: 015
Tipo de Tarea: Desarrollo	Puntos estimados: 15
Nombre Tarea: Pantalla seguimiento de ruta cliente	
Fecha de Inicio: 21/02/2022	Fecha Fin: 11/03/2022
Descripción: La pantalla de seguimiento debe reflejar por dónde va el recorrido de la ruta elegida en tiempo real tomando las coordenadas de la base de datos para que sea visible en el mapa de Google.	

*Tabla N.- 65 Tarea 035  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T036	Código Historia: 016
Tipo de Tarea: Desarrollo	Puntos estimados: 4
Nombre Tarea: CRUD Conductor	
Fecha de Inicio: 14/03/2022	Fecha Fin: 17/03/2022
Descripción: Se debe crear una pantalla para el ingreso de los datos del conductor, debe tener la opción de actualizar cualquiera de sus atributos, también debe mostrar un listado de los conductores activos.	

*Tabla N.- 66 Tarea 036  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T037	Código Historia: 017
Tipo de Tarea: Desarrollo	Puntos estimados: 4
Nombre Tarea: CRUD Buses	
Fecha de Inicio: 18/03/2022	Fecha Fin: 23/03/2022
Descripción: Se debe crear una pantalla para el ingreso de los datos del Bus, debe tener la opción de actualizar cualquiera de sus atributos, también debe mostrar un listado de los Buses activos.	

*Tabla N.- 67 Tarea 037  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T038	Código Historia: 018
Tipo de Tarea: Desarrollo	Puntos estimados: 4
Nombre Tarea: CRUD líneas	
Fecha de Inicio: 24/03/2022	Fecha Fin: 29/03/2022
Descripción: Se debe crear una pantalla para el ingreso de los datos de las líneas, debe tener la opción de actualizar cualquiera de sus atributos, también debe mostrar un listado de las líneas activas.	

*Tabla N.- 68 Tarea 038  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T039	Código Historia: 019
Tipo de Tarea: Desarrollo	Puntos estimados: 4
Nombre Tarea: CRUD Paradas	
Fecha de Inicio: 30/03/2022	Fecha Fin: 04/04/2022
Descripción: Se debe crear una pantalla para el ingreso de los datos de las Paradas se debe tomar las coordenadas geográficas de un mapa, debe tener la opción de actualizar cualquiera de sus atributos, también debe mostrar un listado de las Paradas activas.	

*Tabla N.- 69 Tarea 039  
Elaborado por: Franklin Lesano.*

<b>Tarea</b>	Responsable: Franklin Vicente Lesano Pérez.
N° Tareas: T040	Código Historia: 020
Tipo de Tarea: Desarrollo	Puntos estimados: 12
Nombre Tarea: Reportes recorridos.	
Fecha de Inicio: 05/04/2022	Fecha Fin: 20/04/2022
Descripción: Se debe presentar un reporte de acuerdo con la fecha establecida y la ruta, una vez que se muestro las rutas se seleccionara para dar paso a la lista de las coordenadas geográficas, se dibujara en el mapa las líneas por donde se realizó el trayecto del bus de acuerdo con la fecha.	

*Tabla N.- 70 Tarea 040  
Elaborado por: Franklin Lesano.*

### **3.3.1.5. Valoración de Historias de Usuario**

Una vez de establecidos las historias de usuario se realizó la estimación de esfuerzo en 4 horas por día laborable.

### **3.3.1.6. Estimación de las Historias de Usuario**

La estimación del proyecto se basó en 4 iteraciones distribuidas en las historias de usuarios segmentadas por tipo de desarrollo, de esta forma tener una estimación aproximada de duración del proyecto.

Iteración	Numero	Historia de Usuario	Tiempo estimado	
			Días	Horas
1	001	Análisis de la base de datos	1	4
1	002	Modelamiento y desarrollo de la base de datos	2	8
1	003	Implementación en un servidor de pruebas	3	12
2	004	Desarrollo del API para tabla Conductor	4	16
2	005	Desarrollo del API para tabla Buses	4	16
2	006	Desarrollo del API para tabla Línea	4	16
2	007	Desarrollo del API para tabla Paradas	5	20
2	008	Desarrollo del API para tabla Ruta	4	16
2	009	Desarrollo del API para tabla Recorrido	4	16
2	010	Desarrollo del API Notificaciones	10	40
2	011	Subir las APIS a un entorno de pruebas	3	12
3	012	Pantalla Login Conductor	6	24
3	013	Pantalla Datos para iniciar la ruta	6	24
3	014	Pantalla de guía de navegación del conductor	15	60
3	015	Pantalla seguimiento de ruta cliente	15	60
4	016	CRUD Conductor	4	16
4	017	CRUD Buses	4	16
4	018	CRUD líneas	4	16
4	019	CRUD Paradas	4	16
4	020	Reportes recorridos.	12	48
Total			114	456

*Tabla N.- 71 Historias de Usuario  
Elaborado por: Franklin Lesano.*

### Iteración 1

Iteración	Número	Historia de Usuario	Tiempo estimado	
			Días	Horas
1	001	Análisis de la base de datos	1	4
1	002	Modelamiento y desarrollo de la base de datos	2	8

1	003	Implementación en un servidor de pruebas	3	12
Total			6	24

*Tabla N.- 72 Historias de Usuario Iteración 1  
Elaborado por: Franklin Lesano.*

### Iteración 2

Iteración	Número	Historia de Usuario	Tiempo estimado	
			Días	Horas
2	004	Desarrollo del API para tabla Conductor	4	16
2	005	Desarrollo del API para tabla Buses	4	16
2	006	Desarrollo del API para tabla Línea	4	16
2	007	Desarrollo del API para tabla Paradas	5	20
2	008	Desarrollo del API para tabla Ruta	4	16
2	009	Desarrollo del API para tabla Recorrido	4	16
2	010	Desarrollo del API Notificaciones	10	40
2	011	Subir las APIS a un entorno de pruebas	3	12
Total			38	152

*Tabla N.- 73 Historias de Usuario Iteración 2  
Elaborado por: Franklin Lesano.*

### Iteración 3

Iteración	Numero	Historia de Usuario	Tiempo estimado	
			Días	Horas
3	012	Pantalla Login Conductor	6	24
3	013	Pantalla Datos para iniciar la ruta	6	24
3	014	Pantalla de guía de navegación del conductor	15	60
3	015	Pantalla seguimiento de ruta cliente	15	60
Total			42	168

*Tabla N.- 74 Historias de Usuario Iteración 3  
Elaborado por: Franklin Lesano.*

#### Iteración 4

Iteración	Numero	Historia de Usuario	Tiempo estimado	
			Días	Horas
4	016	CRUD Conductor	4	16
4	017	CRUD Buses	4	16
4	018	CRUD líneas	4	16
4	019	CRUD Paradas	4	16
4	020	Reportes recorridos.	12	48
Total			28	112

*Tabla N.- 75 Historias de Usuario Iteración 4  
Elaborado por: Franklin Lesano.*

### 3.3.1.7. Plan de Entrega

N°	Historia de Usuario	Tiempo Estimado		Iteración Asignada				Entrega Asignada			
		Días	Horas	1	2	3	4	1	2	3	4
001	Análisis de la base de datos	1	4	x				x			
002	Modelamiento y desarrollo de la base de datos	2	8	x				x			
003	Implementación en un servidor de pruebas	3	12	x				x			
004	Desarrollo del API para tabla Conductor	4	16		x				x		
005	Desarrollo del API para tabla Buses	4	16		x				x		
006	Desarrollo del API para tabla Línea	4	16		x				x		
007	Desarrollo del API para tabla Paradas	5	20		x				x		
008	Desarrollo del API para tabla Ruta	4	16		x				x		
009	Desarrollo del API para tabla Recorrido	4	40		x				x		
010	Desarrollo del API Notificaciones	10	12		x				x		
011	Subir las APIS a un entorno de pruebas	3	24		x				x		
012	Pantalla Login Conductor	6	24			x				x	
013	Pantalla Datos para iniciar la ruta	6	60			x				x	
014	Pantalla de guía de navegación del conductor	15	60			x				x	
015	Pantalla seguimiento de ruta cliente	15	16			x				x	
016	CRUD Conductor	4	16				x				x



017	CRUD Buses	4	16				x				x
018	CRUD líneas	4	16				x				x
019	CRUD Paradas	4	48				x				x
020	Reportes recorridos.	12	456				x				x

*Tabla N.- 76 Plan de Entrega  
Elaborado por: Franklin Lesano.*

### 3.3.2. Fase 2: Diseño

#### 3.3.2.1. Metáfora

La aplicación GPS para los conductores ayudara a llevar la ruta correcta asignada a cada conductor de esta forma los administradores del sistema tendrán conocimiento de la ubicación en tiempo real, así como tener un reporte para reclamos en el departamento de servicios laborales.

<b>Iteración</b>	<b>N°</b>	<b>Historia</b>	<b>Prioridad</b>	<b>Riesgo</b>	<b>Estado de desarrollo</b>	<b>Pruebas</b>
1	1	Análisis de la base de datos	1	Medio	Completo	Aprobado
	2	Modelamiento y desarrollo de la base de datos	1	Alto	Completo	Aprobado
	3	Implementación en un servidor de pruebas	1	Alto	Completo	Aprobado
2	4	Desarrollo del API para tabla Conductor	1	Alto	Completo	Aprobado
	5	Desarrollo del API para tabla Buses	1	Alto	Completo	Aprobado

	6	Desarrollo del API para tabla Línea	1	Alto	Completo	Aprobado
	7	Desarrollo del API para tabla Paradas	1	Alto	Completo	Aprobado
	8	Desarrollo del API para tabla Ruta	1	Alto	Completo	Aprobado
	9	Desarrollo del API para tabla Recorrido	1	Alto	Completo	Aprobado
	10	Desarrollo del API Notificaciones	1	Medio	Completo	Aprobado
	11	Subir las APIS a un entorno de pruebas	1	Alto	Completo	Aprobado
3	12	Pantalla Login Conductor	1	Alto	Completo	Aprobado
	13	Pantalla Datos para iniciar la ruta	1	Alto	Completo	Aprobado
	14	Pantalla de guía de navegación del conductor	1	Alto	Completo	Aprobado
	15	Pantalla seguimiento de ruta cliente	1	Alto	Completo	Aprobado
4	16	CRUD Conductor	1	Alto	Completo	Aprobado
	17	CRUD Buses	1	Alto	Completo	Aprobado
	18	CRUD líneas	1	Alto	Completo	Aprobado
	19	CRUD Paradas	1	Alto	Completo	Aprobado
	20	Reportes recorridos.	1	Medio	Completo	Aprobado

*Tabla N.- 77 Metáfora  
Elaborado por: Franklin Lesano.*

El Sistema consta de varios procesos entre ellos está la app del conductor, la app del pasajero y por último tendremos el gestor web para los administradores o usuarios para ello cuenta con los procesos siguientes:

- 1. Modulo Webs Services:** Responsable de la interacción entre los aplicativos móviles, el gestor web, así como de interactuar con la base de datos.
- 2. Modulo Aplicativo Móvil:** El aplicativo móvil necesario para el envío de las coordenadas geográficas hacia la base de datos, encargada de la orientación en los recorridos, así como el envío de notificaciones en caso de incidentes.
- 3. Modulo Aplicativo Conductor:** El aplicativo móvil para pasajeros traerá la información de las coordenadas geográficas de la base de datos cuando retorne la información comenzará a dibujar en el mapa.
- 4. Modulo Login Ingreso al Gestor:** El ingreso al sistema debe estar autenticado para acceder al sistema.
- 5. Modulo CRUD Conductor:** El ingreso, actualización de los registros para los conductores se realiza a través del sistema web, una vez realizado el proceso se reflejará en los aplicativos, así como en las tablas del gestor web.
- 6. Modulo CRUD Bus:** El ingreso, actualización de los registros para los buses se realiza a través del sistema web, una vez realizado el proceso se reflejará en los aplicativos, así como en las tablas del gestor web.
- 7. Modulo CRUD Líneas y Paradas:** El módulo de líneas y paradas es la parte central de las aplicaciones, permite el ingreso y actualización de los registros, al ser maestro detalle están en el mismo modulo y los registros de las paradas depende de la información de los registros de las líneas.
- 8. Modulo Reportes:** Muestra información de las rutas disponibles en una fecha buscada, cuando se seleccione una de estas rutas desplegara la lista de las coordenadas capturas por el dispositivo del conductor, el gestor tiene la opción de sacar el recorrido en forma de puntos o en forma de líneas.

### 3.3.2.2. Tarjetas CRC

<b>Ingreso al aplicativo conductor</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Ingreso al aplicativo mediante clave y usuario, verificar los datos en la base de datos.	Capa de acceso de datos
Observaciones:	

*Tabla N.- 78 Tarjetas CRC Ingreso al aplicativo conductor  
Elaborado por: Franklin Lesano.*

<b>Iniciar ruta conductor</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Validar los parámetros de ingreso placa, línea, horario y orientación para iniciar ruta.	Capa de acceso de datos
Observaciones:	

*Tabla N.- 79 Tarjeta CRC Iniciar ruta conductor  
Elaborado por: Franklin Lesano.*

<b>Navegación recorrido conductor</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
información del conductor, placa del bus, línea, horario y orientación.	Capa de acceso de datos Administrador Firebase
envió de notificación push	
Observaciones:	

*Tabla N.- 80 Tarjetas CRC Navegación recorrido conductor  
Elaborado por: Franklin Lesano.*

<b>Visualizar línea pasajero</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Obtener la información si existe recorrido de la línea solicitada.	Capa de acceso de datos Administrador Firebase
Recibir notificaciones push	

Observaciones:

*Tabla N.- 81 Tarjeta CRC Visualizar línea pasajero  
Elaborado por: Franklin Lesano.*

<b>Autenticación gestor web</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Ingreso al sistema web mediante usuario y clave, validar si existe en la base de datos	Capa de acceso de datos
Observaciones:	

*Tabla N.- 82 Tarjeta CRC Autenticación gestor web  
Elaborado por: Franklin Lesano.*

<b>Crud conductor</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Ingreso, actualización y eliminación de los registros en la base de datos tabla conductor.	Capa de acceso de datos CRUD Conductor
Observaciones:	

*Tabla N.- 83 Tarjeta CRC Crud conductor  
Elaborado por: Franklin Lesano.*

<b>Crud bus</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Ingreso, actualización y eliminación de los registros en la base de datos tabla bus.	Capa de acceso de datos CRUD bus
Observaciones:	

*Tabla N.- 84 Tarjeta CRC Crud bus  
Elaborado por: Franklin Lesano.*

<b>Crud líneas y paradas</b>	
<b>Responsabilidades</b>	<b>Colaboradores</b>

Ingreso, actualización y eliminación de los registros en la base de datos tabla líneas.	Capa de acceso de datos CRUD líneas y paradas
Ingreso, actualización y eliminación de los registros en la base de datos tabla Paradas.	
Observaciones:	

*Tabla N.- 85 Tarjeta CRC Crud líneas y paradas  
Elaborado por: Franklin Lesano.*

<b>Reporte rutas</b>	
Responsabilidades	Colaboradores
Ingreso de parámetros de línea y fecha a buscar, segmentación del tiempo del recorrido, graficación mediante puntos o líneas	Capa de acceso de datos, Google Maps
Observaciones:	

*Tabla N.- 86 Tarjeta CRC Reporte rutas  
Elaborado por: Franklin Lesano.*

### 3.3.2.3. Estructura del proyecto

En el proyecto de investigación se tomó en cuenta herramientas necesarias para el desarrollo y ambiente de pruebas, de igual forma el funcionamiento en el app y sitio web.

#### **Hardware**

1 Laptop HP NoteBox core I7, 16 GB Ram.

1 Samsun 20S 4 GB Ram, 32GB Almacenamiento interno.

1 Samsun A12 4 GB Ram, 16GB Almacenamiento interno.

#### **Software**

Tecnología aplicada para el desarrollo, empezando con la web APIs, Asp.Net API Controller con Net Framework 4.5, PostgreSQL 11, Xamarin en Visual estudio 2019,

complementos adicionales como Shiny para las cercas virtuales, Google maps para la app de los pasajeros y reportes en el sitio Web, por último, la configuración para las notificaciones en Firebase

### Arquitectura del proyecto

En el desarrollo del proyecto está basada la funcionalidad tanto para la App como para el CRUD en la web tomando en cuenta la arquitectura del proyecto, de esta forma se tomó el modelo cliente servidor donde la base de datos y las APIS estarán almacenadas en un servidor y los clientes accederán a los datos mediante los webs services, de la misma forma los datos de los buses serán enviados a través de las webs service hacia la base de datos en el servidor.

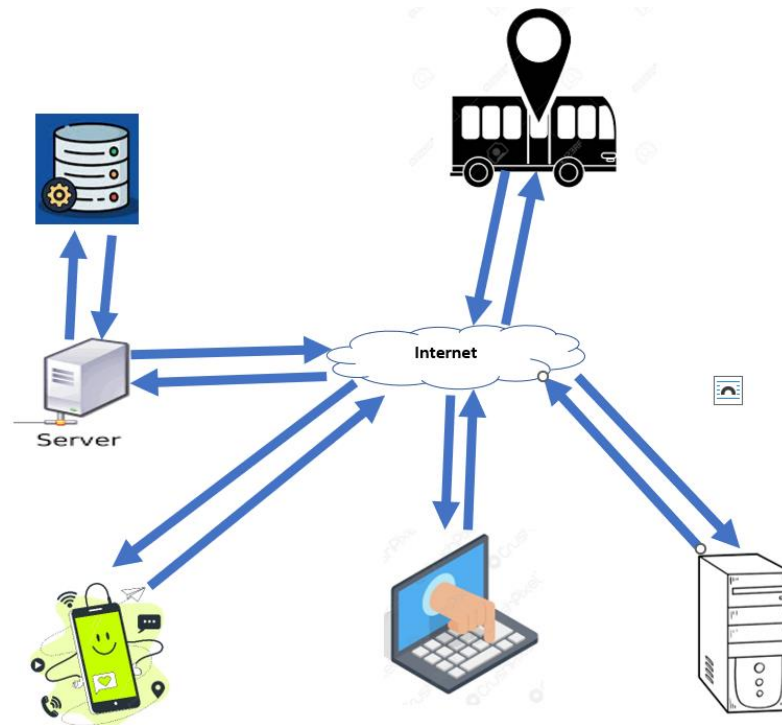


Figura 14 Arquitectura del sistema  
Elaborado por: Franklin Lesano.

#### 3.3.2.4. Diseño de la base de datos

Para el desarrollo del aplicativo se propuso utilizar la base de datos de la compañía, pero la empresa no poseía más licencias en el gestor de la base de datos Oracle, por lo que se optó por implementar en un gestor de base de datos en PostgreSQL el cual tiene licencia abierta.

En el modelamiento de la base de datos se tomó en cuenta la estructura de los recorridos que realiza los buses en sus diferentes horarios, así de esta forma se pudo sacar los actores y sus atributos principales para la gestión del proyecto, con ello se pudo sacar 6 tablas necesarias y básicas para el funcionamiento del sistema.

**Las principales tablas para utilizarse son:**

**Bus.** - Los buses son elementos de transporte de pasajeros donde estará la información de bus asignado en las rutas, los administradores serán encargados de cargar esta información y mantener actualizada de ser necesario, para los reportes es necesarios ya que sabremos en qué fecha y que ruta inicio el recorrido.

**Conductor.** - Los conductores son recursos humanos necesarios para la conducción de los buses, esta tabla almacenara los datos de conductores registrados, a de más de ello se les asignara un usuario que es su número de cedula y una clave para autenticarse en el aplicativo de seguimiento de rutas, en el sistema el administrador ingresara los datos y de ser necesario se actualizara, en los reportes es necesario ya que aparecerá como responsable del inicio de la ruta en el recorrido.

**Línea.** - Las líneas representan el número y el sector por donde debería ir el bus asignado, esta tabla almacenara información como el nombre y el sentido de la ruta esta tabla diferenciara de otros sectores por donde debería iniciar el recorrido.

**Parada.** - La parada son representaciones de un lugar en la ruta que va a hacer el bus, la tabla almacenara el nombre de la parada, las coordenadas geográficas y un orden que deberá seguir en bus, el administrador puede ingresar nuevas paradas a las líneas, así como actualizarlas si fuera necesario, en los reportes son necesarios para segregar las rutas a escoger en base a la fecha que deseen buscar.

**Ruta.** - La ruta representa el lugar y la fecha que inicio el recorrido en base a los parámetros como quien estaba conduciendo, bus que inicio la ruta y la línea que cubría, esta tabla recoge los datos cada vez que un conductor inicia una ruta y son diferentes para cada inicio, en los reportes son necesarios para buscar la fecha de inicio de las líneas que cargaron los conductores.

**Recorrido.** - El recorrido representa las coordenadas de cada lugar que esta el bus en diferente intervalo de tiempo, estos datos son recogidos por el GPS cada intervalo y se



almacena en esta tabla, en los reportes son los más importantes porque a partir de esta información se dibujara en puntos o en línea para mostrar el recorrido

### 3.3.2.5. Diseño de Interfaces Aplicación Móvil

Los diseños de interfaces visuales para el usuario final, ayuda a cumplir con los requerimientos y dar una perspectiva de la funcionalidad y los campos requeridos para la interacción entre el sistema y el usuario, tomando en cuenta que los diseños deben ser tanto como para el aplicativo como para el gestor web, deben de ir con la mano con las historias de usuario antes levantadas.

#### Diseño de Loguin Aplicativo Conductor

Para el login del aplicativo se establece el usuario que será la cedula y la clave, estos datos están en la base de datos, y son autenticados a través de una web service tipo POST.



Figura 15 Diseño de pantalla login conductor  
Elaborado por: Franklin Lesano.

#### Diseño Ingreso Datos de la Ruta

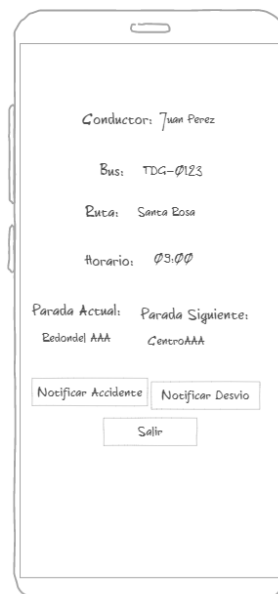
Para el registro de los datos de la ruta es necesario tener datos precargados como la línea que va a recorrer, la unidad asignada y la hora de recorrido, una vez autenticado esta información en pantalla se debe mostrar la siguiente página caso contrario deberá alertar que no son datos correctos, todos estos datos provienen de la base conectados por la API Rest.



*Figura 16* Diseño de pantalla de ingreso de la ruta  
**Elaborado por:** Franklin Lesano.

### **Diseño Pantalla de Recorrido**

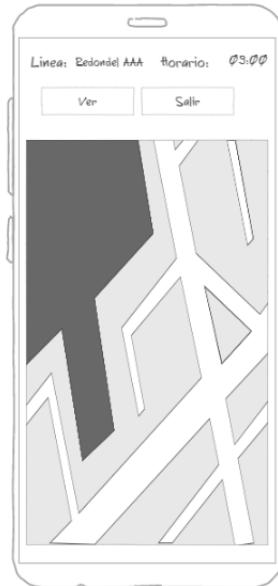
La pantalla de Recorrido debe mostrar la información de la ruta, Bus, Conductor, Línea y horario, entre las funcionalidades adicionales esta enviar una alerta sonora de la parada actual y mostrar cual es la parada actual y la siguiente parada que debe continuar, las paradas son datos que debe traer de la base de datos a través de las Webs Service.



*Figura 17* Diseño de pantalla del recorrido  
**Elaborado por:** Franklin Lesano.

### **Diseño Pantalla de Clientes**

La pantalla de Clientes debe mostrar el recorrido de la línea que desea ver, para ello el usuario debe seleccionar la línea y el horario una vez que se autentifica en la base de datos y existe deberá a empezar a dibujar y mostrar donde está el bus.



*Figura 18 Diseño de pantalla de pasajeros  
Elaborado por: Franklin Lesano.*

### **3.3.2.6. Diseño de Interfaces Gestor Web**

#### **Diseño de Pantalla Conductores**

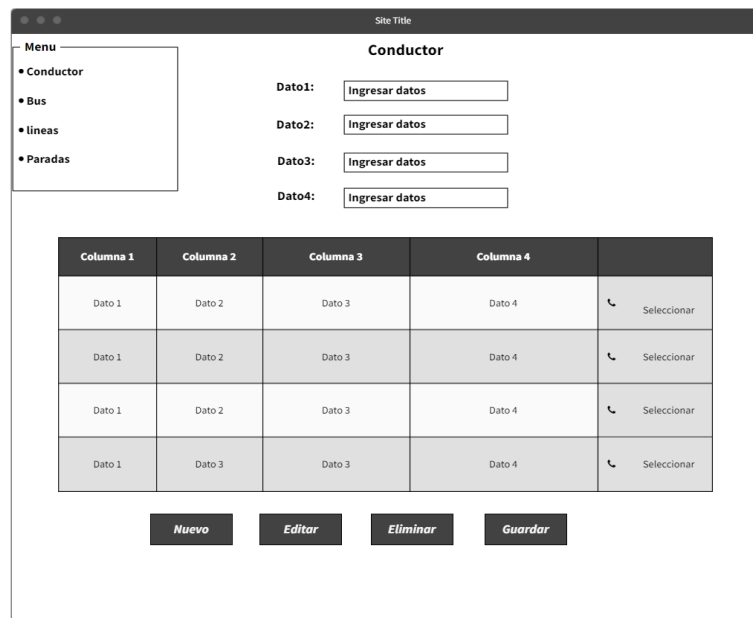
La pantalla de login debe autenticarse el usuario y la clave del gestor del sistema para ello debe ingresar una clave y un usuario, y a través de la web service se autenticará en el sistema para dar paso al menú.



*Figura 19 Diseño de pantalla login administrativo  
Elaborado por: Franklin Lesano.*

## Diseño de Pantalla Conductores

El diseño de pantalla de ingreso de la información de los conductores debe contener los campos requeridos para ingresar un registro nuevo, editar un registro y eliminar, una vez ingresado se debe guardar la información, con cada registro ingresado se debe reflejar en una tabla para visualizar.



*Figura 20 Diseño de pantalla de conductores  
Elaborado por: Franklin Lesano.*

## Diseño de Pantalla Buses

El diseño de pantalla de ingreso de la información de los Buses debe contener los campos requeridos para ingresar un registro nuevo, editar un registro y eliminar, una vez ingresado se debe guardar la información, con cada registro ingresado se debe reflejar en una tabla para visualizar.

Columna 1	Columna 2	Columna 3	Columna 4	
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 3	Dato 3	Dato 4	Seleccionar

*Figura 21 Diseño de pantalla de buses  
Elaborado por: Franklin Lesano.*

## Diseño de Pantalla Líneas

El diseño de pantalla de ingreso de la información de las Líneas debe contener los campos requeridos para ingresar un registro nuevo, editar un registro y eliminar, una vez ingresado se debe guardar la información, con cada registro ingresado se debe reflejar en una tabla para visualizar.

Columna 1	Columna 2	Columna 3	Columna 4	
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 3	Dato 3	Dato 4	Seleccionar

*Figura 22 Diseño de pantallas de líneas  
Elaborado por: Franklin Lesano.*

## Diseño de Pantalla Paradas

El diseño de pantalla de ingreso de la información de los conductores debe contener los campos requeridos para ingresar un registro nuevo, editar un registro y eliminar, una vez ingresado se debe guardar la información, con cada registro ingresado se debe reflejar en una tabla para visualizar, adicional a ello se debe abrir un mapa para sacar las coordenadas geográficas.

Columna 1	Columna 2	Columna 3	Columna 4	
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 2	Dato 3	Dato 4	Seleccionar
Dato 1	Dato 3	Dato 3	Dato 4	Seleccionar

Figura 23 Diseño de pantalla de paradas  
Elaborado por: Franklin Lesano.

Latitud: -7,27366

Longitud: 0,56323

Figura 24 Diseño de pantallas obtención de coordenadas geográficas  
Elaborado por: Franklin Lesano.

### 3.3.3. Fase 3 Codificación

#### 3.3.3.1. Bases de datos

En la creación de la base de datos se utilizó la versión 11.2, el cual se descargó de la página oficial de PostgreSQL el cual tiene la opción de descargar un archivo .exe para Windows Server 2012, una vez instalado el motor de la base de datos se procedió a crear las tablas con sus atributos.

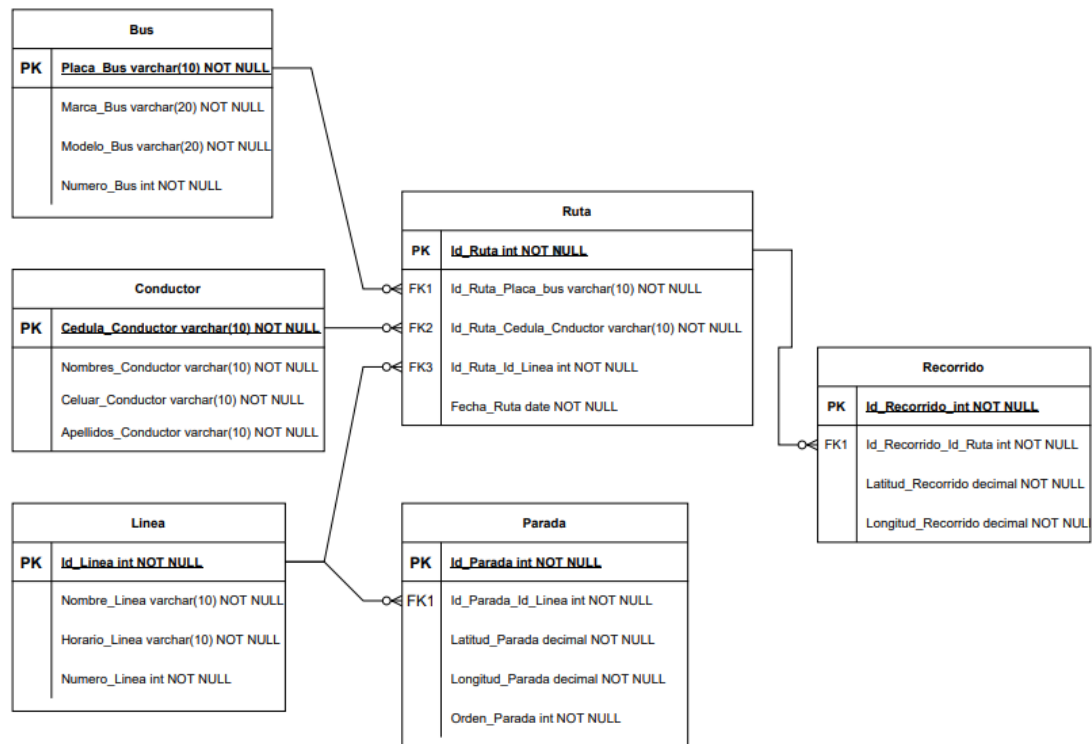


Figura 25 Base de datos modelo, entidad, relación  
Elaborado por: Franklin Lesano.

#### Tabla Buses

La tabla buses permite el registro de los datos además de ello se puso una bandera de control para no eliminar el registro solo para que no sea visible ya que por integridad de la información no se debe borrar los datos.

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [Bus](
```

```

        [Placa_Bus] [varchar](8) NOT NULL,
        [Marca_Bus] [varchar](15) NULL,
        [Modelo_Bus] [varchar](15) NULL,
        [Estado_Bus] [varchar](10) NULL,
CONSTRAINT [PK_Bus] PRIMARY KEY CLUSTERED
(
        [Placa_Bus] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

### **Tabla de Conductor**

La tabla de conductor permite el registro de los datos además de ello se puso una bandera de control para no eliminar el registro solo para que no sea visible ya que por integridad de la información no se debe borrar los datos.

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [Conductor](
        [Cedula_Condu] [varchar](10) NOT NULL,
        [Nombre_Condu] [varchar](20) NULL,
        [Apellido_Condu] [varchar](20) NULL,
        [Clave_Condu] [varchar](15) NULL,
        [Estado_Condu] [varchar](10) NULL,
CONSTRAINT [PK_Conductor] PRIMARY KEY CLUSTERED
(
        [Cedula_Condu] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```



### **Tabla Línea**

La tabla línea permite el registro de los datos necesarios para tener información acerca de las líneas disponibles para que los conductores puedan hacer sus recorridos.

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [Linea](
    [Id_Linea] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Linea] [varchar](20) NULL,
    [Sentido_Linea] [varchar](20) NULL,
    [Horario_Linea] [varchar](5) NULL,
    CONSTRAINT [PK_Linea] PRIMARY KEY CLUSTERED
(
    [Id_Linea] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

### **Tabla Parada**

La tabla parada es la medula central para determinar la ruta que seguirá los buses ya que posee la información de las coordenadas geográficas y en el sistema móvil determina el orden que debe seguir el bus.

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [Parada](
    [Id_Parada] [int] IDENTITY(1,1) NOT NULL,
    [Nombre_Parada] [varchar](30) NULL,
    [Latitud_Parada] [float] NULL,
```

```

        [Longitud_Parada] [float] NULL,
        [Orden_Parada] [int] NULL,
        [Id_Linea_Parada] [int] NULL,
    CONSTRAINT [PK_Parada] PRIMARY KEY CLUSTERED
    (
        [Id_Parada] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

### **Tabla Recorrido**

La tabla recorrido permite el registro de las coordenadas geográficas esta tablas es central y de gran importancia en los reportes así como para los clientes de la aplicación, ya que de ella depende la graficación de los puntos en los mapas y ver como se mueve o se movió el bus en el recorrido.

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [Recorrido](
    [Id_Recorrido] [int] NULL,
    [Hora_Recorrido] [varchar](8) NULL,
    [Longitud_Recorrido] [float] NULL,
    [Latitud_Recorrido] [float] NULL,
    [Id_Ruta_Recorrido] [int] NULL
) ON [PRIMARY]

```

### **Tabla Ruta**

La tabla ruta se registra los datos de la provenientes del registro de los conductores para inicio de la ruta ya que es necesario como tabla padre para la recolección de las coordenadas.

```

SET QUOTED_IDENTIFIER ON

```

```

GO
SET ANSI_PADDING ON
GO
CREATE TABLE [Ruta](
    [Id_Ruta] [int] IDENTITY(1,1) NOT NULL,
    [Fecha_Inicio_Ruta] [datetime] NULL,
    [Cedula_Conductor_Ruta] [varchar](10) NULL,
    [Placa_Bus_Ruta] [varchar](8) NULL,
    [Id_Linea_Ruta] [int] NULL,
    CONSTRAINT [PK_Ruta] PRIMARY KEY CLUSTERED
(
    [Id_Ruta] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

### 3.3.3.2. Webs APIs

#### 3.3.3.2.1. Conexión a la base de datos

En la conexión a la base de datos se utilizó un nuget Npgsql con ello se consiguió acceder a la base de datos y hacer consultas.

```

public string CadenaC()
{
    String cadena = " User Id = "+
        " Host = elephantsql.com;" +
        " Port = 5432; Database = ";
    return cadena;
}

```

#### 3.3.3.2.2. Creación APIS REST Conductores

**PUT api/PGConductor.**

El método Put de conductor recibe como parámetro un objeto conductor y crea un registro a través de un insert en lenguaje SQL, si el registro del conductor es correcto retornara 1 si es correcto y 0 si ocurrido un error.

#### **GET api/PGConductor**

El método Get de conductor no requiere ningún dato adicional y a través de una instrucción SQL retorna una lista de clases de tipo conductor, si no hay registros retornara un null.

#### **GET api/PGConductor?cedulaC={cedulaC}**

El método del conductor recibe un string de 10 dígitos realiza una consulta SQL y retorna un registro de tipo conductor si no existe retornara un null.

#### **POST api/PGConductor**

El método post de conductor recibe como parámetro un objeto de tipo conductor, realiza una actualización de tipo SQL, si el registro se actualiza correctamente retornara un 1 caso contrario retornara un 0.

#### **PUT api/PGConductor?cedulaD={cedulaD}**

El método put de conductor recibe como parámetro un string de 10 dígitos, realiza una consulta SQL y realiza una actualización de la bandera EstaActivo, y lo cambia por DESACTIVADO, para evitar eliminar el registro, si el registro se actualiza correctamente retornara un 1 caso contrario retornara un 0.

#### **PUT api/PGConductor?cedulaA={cedulaA}**

El método put de conductor recibe como parámetro un string de 10 dígitos, realiza una consulta SQL y realiza una actualización de la bandera esta activo, y lo cambia por ACTIVADO, para evitar eliminar el registro, si el registro se actualiza correctamente retornara un 1 caso contrario retornara un 0

#### **GET api/PGConductor?cedulaV={cedulaV}&claveV={claveV}**

El método get de conductor recibe dos parámetros un string de 10 dígitos y el otro string de máximo 12 dígitos, realiza una consulta SQL y si existe retornara un 1 caso contrario de no existir retornara un 0.

## PGConductor

API	Description
<a href="#">PUT api/PGConductor</a>	No documentation available.
<a href="#">GET api/PGConductor</a>	No documentation available.
<a href="#">GET api/PGConductor?cedulaC={cedulaC}</a>	No documentation available.
<a href="#">POST api/PGConductor</a>	No documentation available.
<a href="#">PUT api/PGConductor?cedulaD={cedulaD}</a>	No documentation available.
<a href="#">PUT api/PGConductor?cedulaA={cedulaA}</a>	No documentation available.
<a href="#">GET api/PGConductor?cedulaV={cedulaV}&amp;claveV={claveV}</a>	No documentation available.
<a href="#">POST api/PGConductor?CedulaC={CedulaC}&amp;ClaveC={ClaveC}</a>	No documentation available.

*Figura 26 Webs Apis Conductor  
Elaborado por: Franklin Lesano.*

### 3.3.3.2.3. Creación APIS REST Bus

#### **GET api/PGBus**

EL método get de bus no recibe ningún parámetro, realiza una consulta de todos los registros y retorna una lista de objetos tipo buses, si no existe ningún registro retornara un null.

#### **GET api/PGBus?gplaca={gplaca}**

EL método get de bus toma un parámetro de tipo string de 8 digitos realiza una consulta de tipo SQL, si existe retornara un objeto de tipo bus si no existe retornara un null.

#### **GET api/PGBus?placaV={placaV}**

EL método get de bus toma un parámetro de tipo string de 8 digitos realiza una consulta de tipo SQL, si existe retornara un 1 si no existe retorna un 0.

#### **POST api/PGBus**

El método post de buses toma como parámetro un objeto de tipo bus y con una instrucción SQL inserta un registro, si es correcto retorna un 1 si no es correcto retornara un 0.

#### **PUT api/PGBus**

El método put de buses toma como parámetro un objeto de tipo bus y con una instrucción SQL actualiza un registro, si es correcto retorna un 1 si no es correcto retornara un 0.

#### **PUT api/PGBus?placaD={placaD}**

El método put de Buses recibe como parámetro un string de 8 dígitos, realiza una consulta SQL y realiza una actualización de la bandera EstaActivo, y lo cambia por DESACTIVADO, para evitar eliminar el registro, si el registro se actualiza correctamente retornara un 1 caso contrario retornara un 0.

#### **PUT api/PGBus?PlacaA={PlacaA}**

El método put de Buses recibe como parámetro un string de 8 dígitos, realiza una consulta SQL y realiza una actualización de la bandera EstaActivo, y lo cambia por ACTIVADO, para evitar eliminar el registro, si el registro se actualiza correctamente retornara un 1 caso contrario retornara un 0.

PGBus	
API	Description
<a href="#">GET api/PGBus</a>	No documentation available.
<a href="#">GET api/PGBus?gplaca={gplaca}</a>	No documentation available.
<a href="#">GET api/PGBus?placaV={placaV}</a>	No documentation available.
<a href="#">POST api/PGBus</a>	No documentation available.
<a href="#">PUT api/PGBus</a>	No documentation available.
<a href="#">PUT api/PGBus?placaD={placaD}</a>	No documentation available.
<a href="#">PUT api/PGBus?PlacaA={PlacaA}</a>	No documentation available.

*Figura 27 Webs Apis bus  
Elaborado por: Franklin Lesano.*

### **3.3.3.2.4. Creación APIS REST Línea**

#### **GET api/PGLinea?idlineab={idlineab}**

EL método get de líneas toma un parámetro de tipo int realiza una consulta de tipo SQL, si existe retornara un objeto de tipo bus si no existe retornara un null.

### **PUT api/PGLinea**

EL método put de líneas toma un parámetro de objeto de tipo línea y realiza una consulta SQL y actualiza el registro, si es correcto retornara un 1 caso contrario si existe alguna falla o error retornara un 0.

### **POST api/PGLinea**

EL método post de líneas toma un parámetro de objeto de tipo línea y realiza una instrucción SQL y realiza una inserción de un registro, si es correcto retornara un 1 caso contrario si existe alguna falla o error retornara un 0.

### **GET api/PGLinea**

EL método get de líneas no recibe ningún parámetro, realiza una consulta de todos los registros y retorna una lista de objetos tipo líneas, si no existe ningún registro retornara un null.

### **GET**

**api/PGLinea?Numero\_LineaV={Numero\_LineaV}&Sentido\_LineaV={Sentido\_LineaV}**

EL método get de líneas recibe 2 parámetros, el primero recibe un entero, el segundo recibe un string para ver el sentido de la línea realiza una consulta SQL, si el registro existe retornara un entero, si no existe retornara un -1.

PGLinea	
API	Description
<a href="#">GET api/PGLinea?idlineab={idlineab}</a>	No documentation available.
<a href="#">PUT api/PGLinea</a>	No documentation available.
<a href="#">POST api/PGLinea</a>	No documentation available.
<a href="#">GET api/PGLinea</a>	No documentation available.
<a href="#">GET api/PGLinea?Numero_LineaV={Numero_LineaV}&amp;Sentido_LineaV={Sentido_LineaV}</a>	No documentation available.
<a href="#">DELETE api/PGLinea?lineE={lineE}</a>	No documentation available.

*Figura 28 Webs apis línea  
Elaborado por: Franklin Lesano.*

### 3.3.3.2.5. Creación APIS REST Parada

#### **GET api/PGParada?lineaB={lineaB}**

El método get de líneas toma como parámetro un entero realizara una instrucción SQL si existe retornara un objeto de tipo línea caso contrario retornara un null.

#### **DELETE api/PGParada?idParada={idParada}**

El método delete de líneas toma un entero realiza una instrucción SQL y realizara la eliminación de la parada, si se realiza correctamente retornara un 1 caso contrario retornara un 0.

#### **PUT api/PGParada**

EL método put de paradas toma un parámetro de objeto de tipo parada y realiza una instrucción SQL y actualiza el registro, si es correcto retornara un 1 caso contrario si existe alguna falla o error retornara un 0.

#### **POST api/PGParada**

EL método post de paradas toma un parámetro de objeto de tipo parada y realiza una instrucción SQL e inserta un registro, si es correcto retornara un 1 caso contrario si existe alguna falla o error retornara un 0.

#### PGParada

API	Description
<a href="#">GET api/PGParada?lineaB={lineaB}</a>	No documentation available.
<a href="#">DELETE api/PGParada?idParada={idParada}</a>	No documentation available.
<a href="#">PUT api/PGParada</a>	No documentation available.
<a href="#">POST api/PGParada</a>	No documentation available.

*Figura 29 Webs apis parada  
Elaborado por: Franklin Lesano.*



### 3.3.3.2.6. Creación APIS REST Ruta

#### **DELETE api/PGRuta?rutaE={rutaE}**

El método delete de rutas toma por parámetro un entero y realiza una operación SQL y elimina un registro de la tabla ruta, si la operación es exitosa retornara un 1 caso contrario retorna un 0.

#### **POST api/PGRuta**

El método post de rutas toma por parámetro un objeto tipo ruta y realiza una operación SQL he insertara un registro en la tabla ruta, si la operación es exitosa retornara un entero identificador del registro insertado caso contrario retorna un 0.

#### **GET api/PGRuta?fechaR={fechaR}&lineaR={lineaR}&Hora={Hora}**

El método get de rutas toma 3 parámetros el primero toma una fecha con formato yyyy-MM-dd hh:mm:ss, el segundo un entero identificador que de una línea existente y el horario tipo string de 5 caracteres mediante una consulta SQL, si existe el registro retornara un entero identificador del registro caso contrario retornara un 0.

#### **GET api/PGRuta?fechaB={fechaB}&lineaB={lineaB}&HoraB={HoraB}**

El método get de rutas toma 3 parámetros el primero toma una fecha con formato yyyy-MM-dd hh:mm:ss, el segundo un entero identificador que de una línea existente y el horario tipo string de 5 caracteres mediante una consulta SQL, si existe el registro retornara un entero identificador del registro con el número máximo para identificar el ultimo registro insertado aso contrario retornara un 0.

### PGRuta

API	Description
<a href="#">DELETE api/PGRuta?rutaE={rutaE}</a>	No documentation available.
<a href="#">POST api/PGRuta</a>	No documentation available.
<a href="#">GET api/PGRuta?fechaR={fechaR}&amp;lineaR={lineaR}</a>	No documentation available.
<a href="#">GET api/PGRuta?fechaB={fechaB}&amp;lineaB={lineaB}&amp;HoraB={HoraB}</a>	No documentation available.

*Figura 30 Webs Apis Ruta  
Elaborado por: Franklin Lesano.*

### 3.3.3.2.7. Creación APIS REST Recorrido

#### POST api/PGRecorido

El método post de recorrido toma por parámetro un objeto tipo recorrido y realiza una operación SQL he insertara un registro en la tabla recorrido, si la operación es exitosa retornara un 1 caso contrario retorna un 0.

#### GET api/PGRecorido?IdRuta={IdRuta}

EL método get de recorrido recibe un entero de parámetro entero que representa el id de la ruta y realizara una operación SQL y retornara un listado de objeto recorrido, si no existe la id de la ruta retornara un null.

#### GET api/PGRecorido?IdRutaB={IdRutaB}

EL método get de recorrido recibe un entero de parámetro entero que representa el id de la ruta y realizara una operación SQL y retornara un objeto tipo recorrido, si no existe retornara un null.

## PGRecorido

API	Description
<a href="#">POST api/PGRecorido</a>	No documentation available.
<a href="#">GET api/PGRecorido?IdRuta={IdRuta}</a>	No documentation available.
<a href="#">GET api/PGRecorido?IdRutaB={IdRutaB}</a>	No documentation available.

*Figura 31 Webs Apis recorrido  
Elaborado por: Franklin Lesano.*

### 3.3.3.2.8. Creación APIS REST Notificaciones

#### GET api/BENotificaciones?token={token}

El método get de notificaciones recibe como parámetro un string de hasta 128 caracteres realiza una operación de registro del token de seguridad en la base de firebase y envía una notificación tipo push de acuerdo con el token ingresado, registra el token en un tópico para recibir notificaciones posteriores, si es exitoso retornara una cadena de 16 caracteres, caso contrario una cadena con el mensaje FAIL.

## GET api/BENotificaciones?titulo={titulo}&mensaje={mensaje}

El método get de notificaciones recibe dos parámetros el primero un string de hasta 32 caracteres y el segundo un string de hasta 68 caracteres, realiza un envío de notificaciones a todos los suscritos de acuerdo con el tópico, no una cadena de 16 caracteres caso contrario una cadena con el mensaje FAIL.

## BENotificaciones

API	Description
<a href="#">GET api/BENotificaciones?token={token}</a>	No documentation available.
<a href="#">GET api/BENotificaciones?titulo={titulo}&amp;mensaje={mensaje}</a>	No documentation available.

*Figura 32 Webs Apis notificaciones  
Elaborado por: Franklin Lesano.*

### 3.3.3.3. Aplicación móvil

#### 3.3.3.3.1. Construcción del Servicio para consumo de las APIs

En la construcción de los métodos HTTPClient se requiere un nuget llamada Microsoft.HttpClient, que permite la comunicación http, y otro nuget llamado Newtonsoft el cual permitirá la serialización y desrealización de las clases que se pase por parámetro y que entrega como lista o como objeto.

```
private const string TransporteUrl = "http://somee.com/";
private static HttpClient clienteApi = CrearCliente(TransporteUrl);
public static async Task<List<T>> ObtenerListas<T>(string controlador)
    where T : new()
{
    List<T> lista = new List<T>();
    try
    {
        if (Connectivity.NetworkAccess == NetworkAccess.Internet)
        {
            var respuesta = await clienteApi.GetAsync(controlador);
            if (respuesta.IsSuccessStatusCode)
            {
```

```

        var contenido = await
            respuesta.Content.ReadAsStringAsync();
        lista =
            JsonConvert.DeserializeObject<List<T>>(contenido);
    }
}
}
catch (WebException)
{
    return lista = null;
}
catch (Exception)
{
    return lista = null;
}
return lista;
}
public static async Task<T> ObtenerDato<T>(string controlador)
    where T : new()
{
    T item = new T();
    try
    {
        if (Connectivity.NetworkAccess == NetworkAccess.Internet)
        {
            var respuesta = await clienteApi.GetAsync(controlador);
            if (respuesta.IsSuccessStatusCode)
            {
                var contenido = await
                    respuesta.Content.ReadAsStringAsync();
                item = JsonConvert.DeserializeObject<T>(contenido);
            }
        }
    }
}

```

```

    }
    catch (WebException)
    {
        return item;
    }
    catch (Exception)
    {
        return item;
    }
    return item;
}
public static async Task<int> ValidarDatos(string controlador)
{
    int dato = 0;
    try
    {
        if (Connectivity.NetworkAccess == NetworkAccess.Internet)
        {
            var respuesta = await clienteApi.GetAsync(controlador);
            if (respuesta.IsSuccessStatusCode)
            {
                var contenido = await
                    respuesta.Content.ReadAsStringAsync();
                dato = JsonConvert.DeserializeObject<int>(contenido);
            }
        }
    }
    catch (WebException ex)
    {
        return dato = -1;
    }
    catch (Exception)
    {

```

```

        return dato = -1;
    }
    return dato;
}
public static async Task<int> AgregarItem<T>(string controlador, T item)
where T : new()
{
    int dato = 0;
    try
    {
        if (Connectivity.NetworkAccess == NetworkAccess.Internet)
        {
            var body = JsonConvert.SerializeObject(item);
            var content = new StringContent(body, Encoding.UTF8,
                "application/json");

            var respuesta = await clienteApi.PostAsync(controlador,
                content);
            if (respuesta.IsSuccessStatusCode)
            {
                var contenido = await
                    respuesta.Content.ReadAsStringAsync();
                dato = JsonConvert.DeserializeObject<int>(contenido);
            }
        }
    }
    catch (WebException)
    {
        return dato = -1;
    }
    catch (Exception)
    {
        return dato = -1;
    }
}

```

```

    }
    return dato;
}
public static async Task<bool> ActualizarItem<T>(string controlador,
    T item) where T : new()
{
    if (Connectivity.NetworkAccess == NetworkAccess.Internet)
    {
        var body = JsonConvert.SerializeObject(item);
        var content = new StringContent(body, Encoding.UTF8,
            "application/json");
        var respuesta = await clienteApi.PutAsync(controlador,
            content);
        return respuesta.IsSuccessStatusCode;
    }
    return false;
}
public static async Task<bool> EliminarItem(string controlador)
{
    if (Connectivity.NetworkAccess == NetworkAccess.Internet)
    {
        var respuesta = await clienteApi.DeleteAsync(controlador);
        return respuesta.IsSuccessStatusCode;
    }
    return false;
}

```

### 3.3.3.3.2. Construcción del Servicio de Geocercas

En construcción del servicio se utilizó un nuget externo llamado Shiny el cual va a permitir hacer cercas virtuales para el control de las paradas, la construcción de los servicios es complejo si no se tiene los conocimientos básicos para este tipo de tecnología las limitantes son que no se puede registrar más allá de los 100 puntos geográficos.

```

public async Task OnStatusChanged(GeofenceState newState, GeofenceRegion
region)
{
    if (newState == GeofenceState.Entered)
    {
        var locales = await TextToSpeech.GetLocalesAsync();
        var localCol = locales.FirstOrDefault
            (x => x.Language == "spa" || x.Language=="es");
        if (localCol != null)
        {
            var opciones = new SpeechOptions()
            {
                Volume = 0.9f,
                Pitch = 1f,
                Locale = localCol
            };
            await TextToSpeech.SpeakAsync($"Parada actual
                {region.Identifier}", opciones);
            CTempParada actualP= new CTempParada();
            actualP.paradaActual= region.Identifier;
            await App.Context.DeleteAllItemAsync<CTempParada>();
            int guar = await
                App.Context.SaveItemAsync<CTempParada>(actualP, true);
        }
    }
    else
    {
        await this.notifications.Send(new Notification()
        {
            Title = "¡Hasta la próxima!",
            Message = $"Te esperamos de vuelta en {region.Identifier}"
        });
    }
}

```



```

    }
public override void ConFiguraServices(IServiceCollection services, IPlatform
platform)
    {
        services.UseGeofencing<MapasGeofenceDelegate>();
        services.UseNotifications();
        services.UseGps();
    }

```

En caso de tener más de un inicio de los Applications se debe implementar el inicio apuntando al servicio, esto en el proyecto de Android, tener en cuenta los permisos de acceso a la locación.

```

public MainApplication(IntPtr handle, JniHandleOwnership transfer) : base(handle,
transfer) { }
public override void OnCreate()
    {
        this.ShinyOnCreate(new ShinyElements.MapasShinyStartup());
        global::Xamarin.Essentials.Platform.Init(this);
        base.OnCreate();
    }

```

Registro de las paradas de acuerdo con la lista de extraída de la consulta en la tabla paradas.

```

public async Task RegistrarGeocerca(CParada parada)
{
    var gpsRequest = new GpsRequest()
    {
        UseBackground = true,
        Priority = GpsPriority.Highest,
        Interval = TimeSpan.FromSeconds(3),
        ThrottledInterval = TimeSpan.FromSeconds(1)
    };
    var geofenceManager = ShinyHost.Resolve<IGeofenceManager>();
    var geofenceAccess = await geofenceManager.RequestAccess();
}

```

```

if (geofenceAccess == AccessState.Available)
{
    var gpsManager = ShinyHost.Resolve<IGpsManager>();
    var gpsAccess = await gpsManager.RequestAccess(gpsRequest);
    if (gpsAccess == AccessState.Available)
    {
        await gpsManager.StartListener(gpsRequest);
        var notificationsManager =
            ShinyHost.Resolve<INotificationManager>();
        var notificationsAccess = await
            notificationsManager.RequestAccess();
        if (notificationsAccess == AccessState.Available)
        {
            var posicion = new
                Position(parada.Latitud_Parada1,
                    parada.Longitud_Parada1);
            var radio = Shiny.Distance.FromMeters(100);
            var geocerca = new
                GeofenceRegion(parada.Nombre_Parada1, posicion, radio)
            {
                NotifyOnEntry = true,
                NotifyOnExit = true,
                SingleUse = false
            };
            Await
                geofenceManager.StartMonitoring(geocerca);
        }
    }
}
}
}

```

### 3.3.3.3.3. Desarrollo Back y Frond de los aplicativos

#### Pantalla Login

La pantalla de login o autenticación toma por parámetros un usuario y una clave, estos parámetros se enviarán a través del de la web service y devolverá un valor de 1 si existe en la base caso contrario un cero, so es uno dará paso a la siguiente pantalla con un objeto tipo usuario para las consultas posteriores.

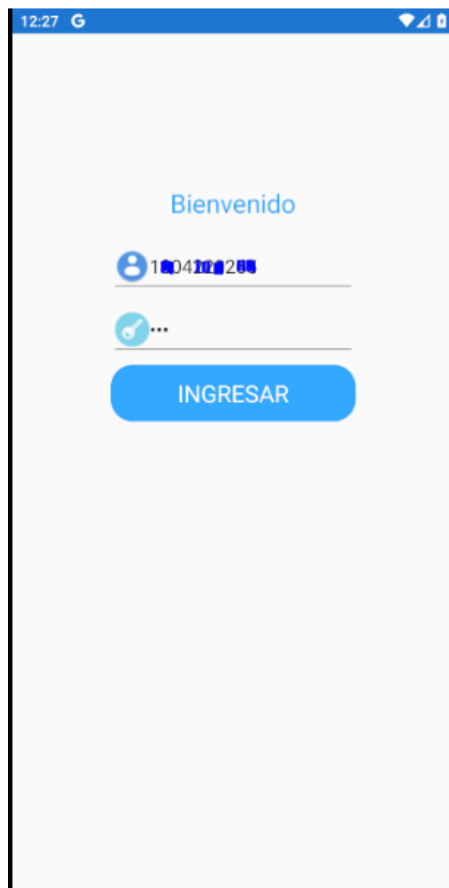
Back-End

```
public async void IngresarRuta()
{
    if (!string.IsNullOrEmpty(this.Usuario))
    {
        if (!string.IsNullOrEmpty(this.Clave))
        {
            IsBusy = true;
            var ResultadoConductor = await
                ServicioTransporte.ValidarDatos(parameCondu
                    ctor);
            if (ResultadoConductor == 1)
            {
                Application.Current.MainPage = new
                    RutaView(Usuario);
            }
            else
            {
                Await
                Application.Current.MainPage.DisplayAlert("Er
                    ror", "No hemos podido autenticar por favor
                    intenta de nuevo ", "ok");
            }
        }
    }
    else
    {
```

```

        Await
        Application.Current.MainPage.DisplayAlert("error",
        "Ingrese el clave", "ok");
    }
}
else
{
    await Application.Current.MainPage.DisplayAlert("Error",
    "Ingrese el usuario", "ok");
}
IsBusy = false;
}
}
FronD-End

```



*Figura 33 Pantalla login conductor  
Elaborado por: Franklin Lesano.*

## Pantalla Inicio Ruta

La pantalla de la ruta requiere parámetros de registro para los reportes como el ingreso de la placa del bus, la línea por la cual va a manejar el horario y el sentido o dirección de la ruta.

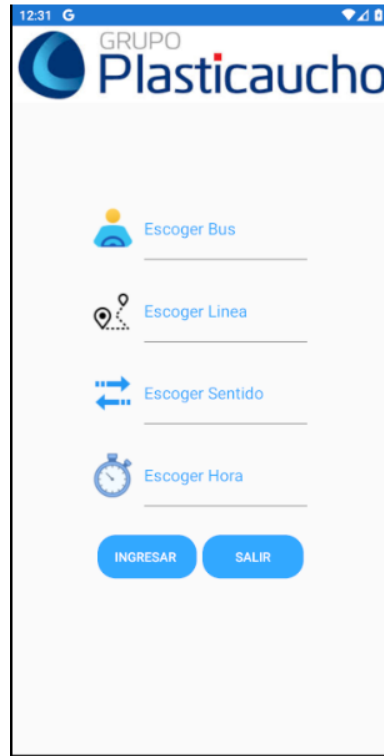
Front-End

```
private async void Ingresar()
{
    bool enBlanco;
    enBlanco = await controlEnBlanco();
    if (enBlanco == false)
    {
        try
        {
            CRuta ruta = new CRuta();
            DateTime FechaHoy = DateTime.Now;
            ruta.Fecha_Inicio_Ruta1 = FechaHoy;
            ruta.Cedula_Conductor_Ruta1 = Usuario;
            ruta.Placa_Bus_Ruta1 = seleccionBu.Placa_Bus1;
            ruta.Id_Linea_Ruta1 = seleccionLi.Id_Linea1;
            ruta.Horario_ruta = SeleccionHora.Trim();
            ChequeoPermisos permiso = new ChequeoPermisos();
            bool ubicacion1 = await
                permiso.CheckAndRequestLocationPermission();
            if (ubicacion1 == true)
            {
                Application.Current.MainPage = new RecorridoView(ruta,
                    Nombre, SeleccionLi);
            }
        }
        catch (System.Net.WebException)
        {
            await Application.Current.MainPage.DisplayAlert("Error",
```

```

        "Chequea tu servicio de internet", "ok");
    }
}
}
Back-End

```



*Figura 34 Pantalla inicio de ruta  
Elaborado por: Franklin Lesano.*

### **Pantalla del recorrido**

La pantalla presentara información como cuál es el conductor, la placa del bus que realiza el recorrido, el horario asignado al conductor y la salida o llegada dependiendo para el recorrido

Back-End

```

public async void ControlTiempo()
{
    while (true)
    {
        await RegistroPuntosGeograficos();
        await Task.Delay(1000);
    }
}

```

```

    }
}
public async void PantallaInicial(CRuta _Ruta, CLinea _Lin1)
{
    Conductor = Conductor;
    Bus = _Ruta.Placa_Bus_Ruta1;
    LineaR = _Lin1.Nombre_Linea1;
    Hora= _Ruta.Horario_ruta;
    Direccion = _Lin1.Sentido_Linea1;
    IdRuta = await ServicioTransporte.AgregarItem
        ("/api/PGRuta",_Ruta);
    ControlTiempo();
    try
    {
        ListaParada = new ObservableCollection<CParada>();
        int c = 0;
        string paradas = $"api/PGParada?lineaB={_Lin1.Id_Linea1}";
        var listaParadas = await
            ServicioTransporte.ObtenerListas<CParada>(paradas);
        foreach (var item in listaParadas)
        {
            if (c==0)
            {
                CTemParada actualP = new CTemParada();
                actualP.paradaActual = item.Nombre_Parada1;
                await App.Context.DeleteAllItemAsync<CTemParada>();
                int guar = await
                    App.Context.SaveItemAsync<CTemParada>(actualP, true);
                c = 2;
            }
            CParada pr = new CParada();
            pr.Id_Parada1 = item.Id_Parada1;
            pr.Nombre_Parada1 = item.Nombre_Parada1;

```

```

        pr.Latitud_Parada1 = item.Latitud_Parada1;
        pr.Longitud_Parada1 = item.Longitud_Parada1;
        pr.Orden_Parada1 = item.Orden_Parada1;
        pr.Id_Linea_Parada1 = item.Id_Linea_Parada1;
        ListaParada.Add(pr);
        await RegistrarGeocerca(pr);
    }
}
catch
{
}
}

public async void enviarNotificacionAccidente()
{
    string NotiMS = $" {LineaR}, Sentido: {Direccion}, Parada: {Entrada}";
    int resulNot= await ServicioTransporte.ObtenerDato<int>(NotiMS);
    if (resulNot==1)
        await Application.Current.MainPage.DisplayAlert("Notificacion",
            "Se a notificado a todos los usuarios", "ok");
    else
        await Application.Current.MainPage.DisplayAlert("Error", "Intente
            mas tarde por favor", "ok");
}

public async void enviarNotificacionDesvio()
{
    string NotiMS = $" {LineaR}, Sentido: {Direccion}, Parada:
        {Entrada}";
    int resulNot = await ServicioTransporte.ObtenerDato<int>(NotiMS);
    if (resulNot == 1)
        await Application.Current.MainPage.DisplayAlert("Notificacion",
            "Se a notificado a todos los usuarios", "ok");
    else

```



```
await Application.Current.MainPage.DisplayAlert("Error", "Intente  
mas tarde por favor", "ok");
```

```
}
```

Frond-End



Figura 35 Pantalla navegación ruta  
Elaborado por: Franklin Lesano.

### 3.3.3.4. Gestor Web

#### Máster Pague

La master page servirá como base para el diseño de pantalla y tener un menú donde pueda gestionar las direcciones a las paginas para las operaciones sw ingreso actualización y listas de las diferentes tablas

Back-End

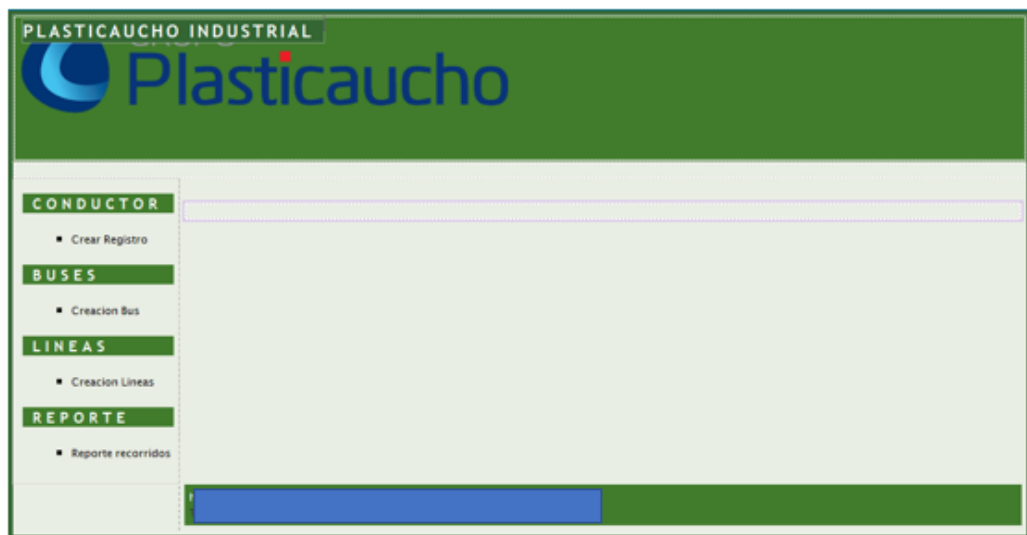
```
<div id="sidebar">  
<h2>Conductor</h2>  
<div class="sidemenu">  
<ul>  
<li><a href="CreacionConductor.aspx">Crear Registro</a></li>  
</ul>  
</div>
```

```

<h2>Buses</h2>
<div class="sidemenu">
  <ul>
    <li><a href="CreacionBus.aspx">Creacion Bus</a></li>
  </ul>
</div>
<h2>Lineas</h2>
<div class="sidemenu">
  <ul>
    <li><a href="CreacionLinea.aspx">Creacion Lineas</a></li>
  </ul>
</div>
<h2>Reporte</h2>
<div class="sidemenu">
  <ul>
    <li><a href="MenuReporte.aspx">Reporte recorridos</a></li>
  </ul>
</div>
</div>

```

FronD-End



*Figura 36 Pantalla master page  
Elaborado por: Franklin Lesano.*

## CRUD Conductor

La página del conductor permite hacer el registro actualización y eliminación de los registros del conductor

Back-End

```
private async void nuevoConductor()
{
    try
    {
        CConductor listaCon = new CConductor();
        listaCon.Cedula_Condu1 = txtCedula.Text.ToString().Trim();
        listaCon.Nombre_Condu1 = txtNombre.Text.ToString().Trim();
        listaCon.Apellido_Condu1 = txtApellido.Text.ToString().Trim();
        listaCon.Celular_Condu1 = txtCelular.Text.ToString().Trim();
        listaCon.Correo_Condu1 = txtCorreo.Text.ToString().Trim();
        listaCon.Clave_Condu1 = txtClave.Text.ToString().Trim();
        int c = await ServiciosAPI.AgregarItem<>(" ", listaCon);
        if (c == 1)
            this.Page.ClientScript.RegisterStartupScript
            (this.GetType(),
            "alert", "alert('Los datos se insertaron correctamente')", true);
        else
            this.Page.ClientScript.RegisterStartupScript(this.GetType(),
            "alert", "alert('Los datos no se insertaron correctamente')",
            true);
    }
    catch (Exception)
    {
        this.Page.ClientScript.RegisterStartupScript(this.GetType(),
            "alert", "alert('Algo ocurrio y no se insertaron los datos')",
            true);
    }
}
```

```

private async void editarConductor()
{
    ServiciosAPI.ObtenerListas<>(" ");
    try
    {
        CConductor listaCon = new CConductor();
        listaCon.Cedula_Condu1 = txtCedula.Text.ToString().Trim();
        listaCon.Nombre_Condu1 = txtNombre.Text.ToString().Trim();
        listaCon.Apellido_Condu1 = txtApellido.Text.ToString().Trim();
        listaCon.Celular_Condu1 = txtCelular.Text.ToString().Trim();
        listaCon.Correo_Condu1 = txtCorreo.Text.ToString().Trim();
        listaCon.Clave_Condu1 = txtClave.Text.ToString().Trim();
        int c = await ServiciosAPI.ActualizarItem<>(" ", listaCon);
        if (c == 1)
            this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
                "alert('Los datos se actualizaron correctamente')", true);
        else
            this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
                "alert('Los datos no se actualizaron correctamente')", true);
    }
    catch (Exception)
    {
        this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
            "alert('Algo ocurrio y no se actualizo los datos')", true);
    }
}

private async void BindData()
{
    dt.Columns.AddRange(new DataColumn[7]
    {
        new DataColumn("Cedula", typeof(string)),
        new DataColumn("Nombre", typeof(string)),
        new DataColumn("Apellido", typeof(string)),

```

```

        new DataColumn("Celular", typeof(string)),
        new DataColumn("Clave", typeof(string))
    });
var GridConductor = await ServiciosAPI.ObtenerListas<>(" ");
foreach (var listaItem in GridConductor)
{
    dt.Rows.Add(listaItem.Cedula_Condu1.ToString(),
        listaItem.Nombre_Condu1.ToString(),
        listaItem.Apellido_Condu1.ToString(),
        listaItem.Celular_Condu1.ToString(),
        listaItem.Clave_Condu1.ToString());
}
gvConductor.DataSource = dt;
gvConductor.DataBind();
}
private async void EliminarRegistro()
{
    if (!String.IsNullOrEmpty(lblCedula.Text))
    {
        int c = await ServiciosAPI.ActualizarItem<Object>(" =" +
            lblCedula.Text.Trim() + "", null);
        if (c == 1)
            this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
                "alert('Se Elimino correctamente')", true);
        else
            this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
                "alert('no se elimino')", true);
    }
}
}

```

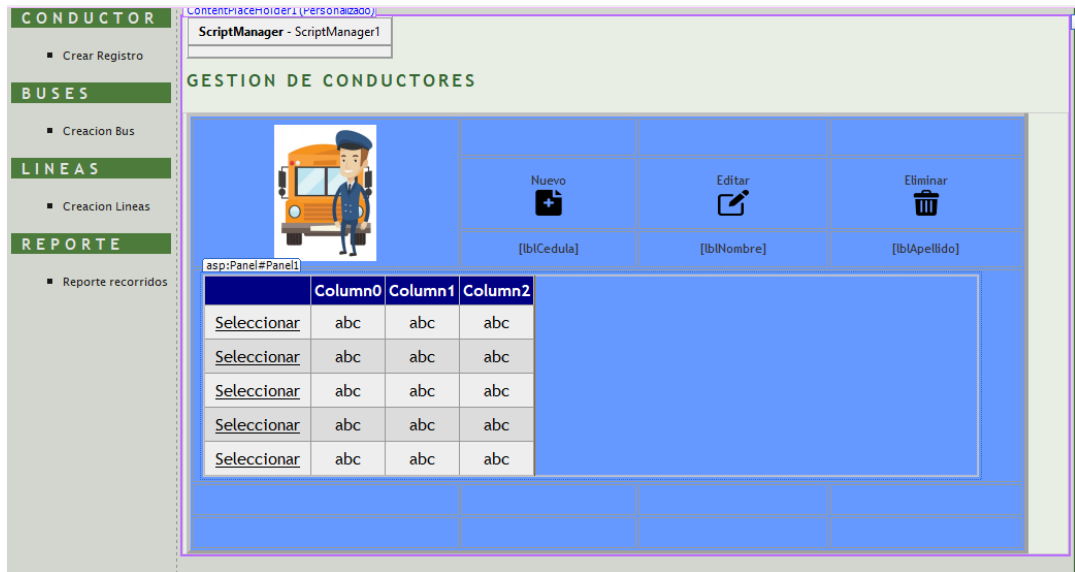


Figura 37 Pantalla crud conductor  
Elaborado por: Franklin Lesano.



Figura 38 Pantalla crud conductor ingreso de datos  
Elaborado por: Franklin Lesano.

## CRUD Buses

El crud de los buses consiste en la actualización de los registros, la inserción y la eliminación de los registros, además de ello se refleja una lista de todos los registros activos.

```
private async void guardarBus(_unBus)
```

```
{
    try
```

```

    {
        string script="";
        int c = await ServiciosAPI.AgregarItem<CBus>(" ", _unBus);
        if (c == 1)
        {
            script = @"<script type='text/javascript'>
                alert('Los datos se insertaron correctamente');
            </script>";
        }
        else
        {
            script = @"<script type='text/javascript'>
                alert('Los datos no se insertaron correctamente');
            </script>";
        }
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
            script, false);
    }
    catch
    {
    }
}

private async void editarBus(_unBus)
{
    try
    {
        string script = "";
        int c = await ServiciosAPI.ActualizarItem<CBus>(" ", _unBus);
        if (c == 1)
        {
            script = @"<script type='text/javascript'>
                alert('Los datos se actualizaron correctamente');
            </script>";
        }
    }
}

```

```

        BindData();
    }
    else
        script = @"<script type='text/javascript'>
            alert('Los datos no se editaron correctamente');
        </script>";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
            script, false);
    }
    catch
    {
    }
}

private async void eliminarBus(string placa)
{
    try
    {
        int c = await ServiciosAPI.ActualizarItem<object>($" null);
        if (c == 1)
        {
            this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
                "alert('Los datos se editaron correctamente')", true);
            BindData();
        }
    }
    else
        this.Page.ClientScript.RegisterStartupScript(this.GetType(), "alert",
            "alert('Los datos no se editaron correctamente')", true);
}
catch
{
}
}

private async void BindData()

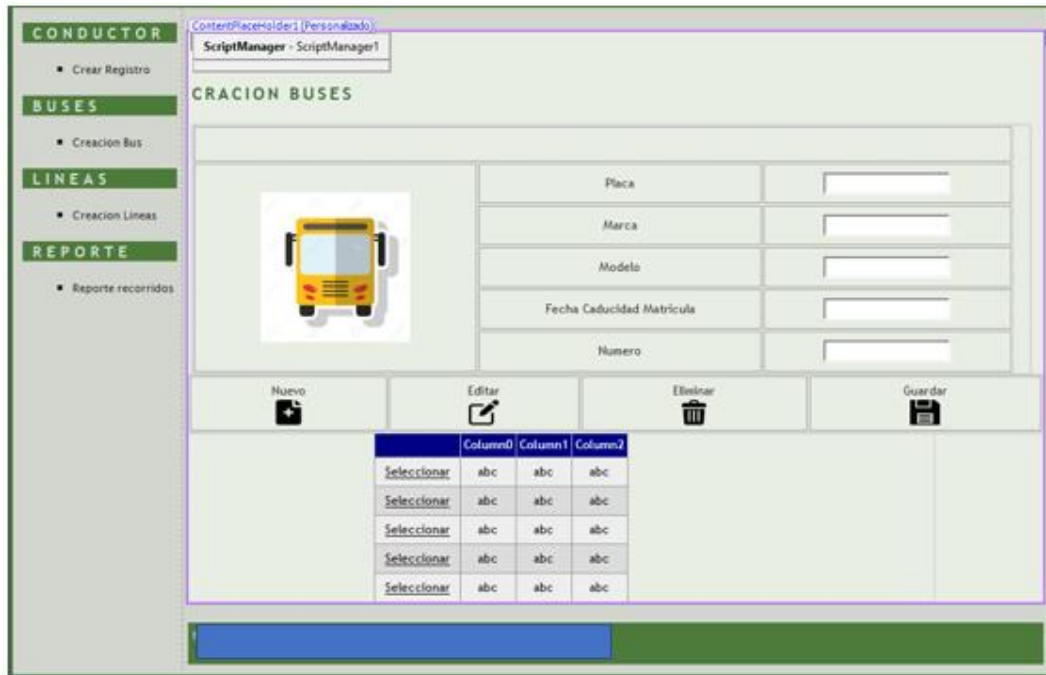
```



```

{
    DataTable dt = new DataTable();
    dt.Columns.AddRange(new DataColumn[5]
        {
            new DataColumn("Placa", typeof(string)),
            new DataColumn("Marca", typeof(string)),
            new DataColumn("Modelo", typeof(string)),
            new DataColumn("Numero", typeof(string))
        });
    var GridConductor = await ServiciosAPI.ObtenerListas<>(" ");
    foreach (var listaItem in GridConductor)
    {
        dt.Rows.Add(listaItem.Placa_Bus1.ToString(),
            listaItem.Marca_Bus1.ToString(),
            listaItem.Modelo_Bus1.ToString(),
            listaItem.Numero_Bus1.ToString());
    }
    gvBus.DataSource = dt;
    gvBus.DataBind();
}
FronD-End

```



*Figura 39 Pantalla crud bus  
Elaborado por: Franklin Lesano.*

## CRUD Líneas

El crud de las líneas consiste en la actualización, inserción y la eliminación de los registros, otra funcionalidad refleja una lista de todos los registros activos, la tabla líneas al ser una tabla maestro detalle con las paradas se realizó en una sola pantalla, se utilizó JavaScript para traer los puntos geográficos Google maps.

Back-End

```
private async void BindDataLinea()
{
    DataTable dt = new DataTable();
    dt.Columns.AddRange(new DataColumn[4] {
        new DataColumn("Id Linea", typeof(string)),
        new DataColumn("Nombre", typeof(string)),
        new DataColumn("Sentido", typeof(string)),
        new DataColumn("Numero", typeof(string))});
    var GridConductor = await ServiciosAPI.ObtenerListas<>(" ");
    foreach (var listaItem in GridConductor)
    {
```

```

        dt.Rows.Add(listaItem.Id_Linea1.ToString(),
                    listaItem.Nombre_Linea1.ToString(),
                    listaItem.Sentido_Linea1.ToString(),
                    listaItem.Numero_Linea1.ToString()
                    );
    }
    gvLinea.DataSource = dt;
    gvLinea.DataBind();
}
private async void BindDataParada(lineaB)
{
    DataTable dt = new DataTable();
    dt.Columns.AddRange(new DataColumn[5] {
        new DataColumn("Nombre", typeof(string)),
        new DataColumn("Latitud", typeof(string)),
        new DataColumn("Longitud", typeof(string)),
        new DataColumn("Orden", typeof(string)),
        new DataColumn("Id Linea", typeof(string))});
    var GridConductor = await ServiciosAPI.ObtenerListas<>();
    foreach (var listaItem in GridConductor)
    {
        dt.Rows.Add(listaItem.Nombre_Parada1.ToString(),
                    listaItem.Latitud_Parada1.ToString(),
                    listaItem.Longitud_Parada1.ToString(),
                    listaItem.Orden_Parada1.ToString(),
                    listaItem.Id_Parada1.ToString());
    }
    gvParadas.DataSource = dt;
    gvParadas.DataBind();
}
private async void guardarLinea(_linea)
{
    try

```

```

    {
        int c = await ServiciosAPI.AgregarItem(" ", );
        if (c >= 1)
        {
            string script = @"<script type='text/javascript'>
                alert('Los datos se insertaron correctamente');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
            script, false);
        }
        else
        {
            string script = @"<script type='text/javascript'>
                alert('Los datos no se insertaron correctamente');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
            script, false);
        }
    }
    catch
    {
    }
}

private async void editarLinea()
{
    try
    {
        int c = await ServiciosAPI.ActualizarItem("api/PGLinea", _linea);
        if (c == 1)
        {
            string script = @"<script type='text/javascript'>
                alert('Los datos se editaron correctamente');
            </script>";

```

```

        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta", script,
        false);
        BindDataLinea();
    }
    else
    {
        string script = @"<script type='text/javascript'>
            alert('Los datos no se editaron correctamente');
        </script>";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
        script, false);
    }
}
catch
{
    string script = @"<script type='text/javascript'>
        alert('Perdida de comunicacion con el servidor');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
    "alerta", script, false);
}
}
private async void eliminarLinea()
{
    try
    {
        int c = await ServiciosAPI.EliminarItem()
        if (c == 1)
        {
            string script = @"<script type='text/javascript'>
                alert('Registro Eliminado');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page),

```

```

        "alerta", script, false);
    BindDataLinea();
}
else
{
    string script = @"<script type='text/javascript'>
        alert('Registro No Eliminado');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, false);
}
}
catch
{
    string script = @"<script type='text/javascript'>
        alert('Error, intenta mas tarde por favor');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, false);
}
}
private async void guardarParada()
{
    try
    {
        int c = await ServiciosAPI.AgregarItem();
        if (c >= 1)
        {
            string script = @"<script type='text/javascript'>
                alert('Los datos se insertaron correctamente');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, false);
        }
    }
}

```

```

    }
else
{
    string script = @"<script type='text/javascript'>
        alert('Los datos no se insertaron correctamente');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
        "alerta", script, false);
}
}
catch
{
}
}
private async void editarParada(CParada _parada)
{
    try
    {
        int c = await ServiciosAPI.ActualizarItem();
        if (c == 1)
        {
            string script = @"<script type='text/javascript'>
                alert('Los datos se editaron correctamente');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
                "alerta", script, false);
            BindDataLinea();
        }
    }
else
{
    string script = @"<script type='text/javascript'>
        alert('Los datos no se editaron correctamente');
    </script>";
}
}

```

```

        ScriptManager.RegisterStartupScript(this, typeof(Page),
        "alerta", script, false);
    }
}
catch
{
    string script = @"<script type='text/javascript'>
        alert('Perdida de comunicacion con el servidor');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
    "alerta", script, false);
}
}
private async void eliminarParada(string idParada)
{
    try
    {
        int c = await ServiciosAPI.EliminarItem()
        if (c == 1)
        {
            string script = @"<script type='text/javascript'>
                alert('Registro Eliminado');
            </script>";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
            "alerta", script, false);
            BindDataLinea();
        }
    }
    else
    {
        string script = @"<script type='text/javascript'>
            alert('Registro No Eliminado');
        </script>";
        ScriptManager.RegisterStartupScript(this, typeof(Page),

```



```

"alerta", script, false);
    }
}
catch
{
    string script = @"<script type='text/javascript'>
        alert('Error, intenta mas tarde por favor');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
    "alerta", script, false);
}
}
}

```

Back-End



Figura 40 Pantalla crud lineas- paradas  
Elaborado por: Franklin Lesano.

Captura de las coordenadas en el mapa de Google utilizando JavaScript, luego de capturar marca un punto como referencia en el mapa.

```

<script type="text/javascript">
    function initMap() {

```

```

// The location of Uluru
const uluru = { lat: - 1.19711, lng: -78.5979977 };
// The map, centered at Uluru
const map = new google.maps.Map(document.getElementById("map3"), {
  zoom: 20,
  center: uluru
});
map.addListener("click", (e) => {
  placeMarkerAndPanTo(e.latLng, map);
});
}
function placeMarkerAndPanTo(latLng, map) {
  var latitudob = latLng.lat();
  var longitudob = latLng.lng();
  document.getElementById('<%= txtLat.ClientID%>').value = latitudob+"";
  document.getElementById('<%= txtLog.ClientID%>').value = longitudob +
"";

  new google.maps.Marker({
    position: latLng,
    map: map,
  });
  map.panTo(latLng);
}
window.initMap = initMap;
</script>

```

FronD-End

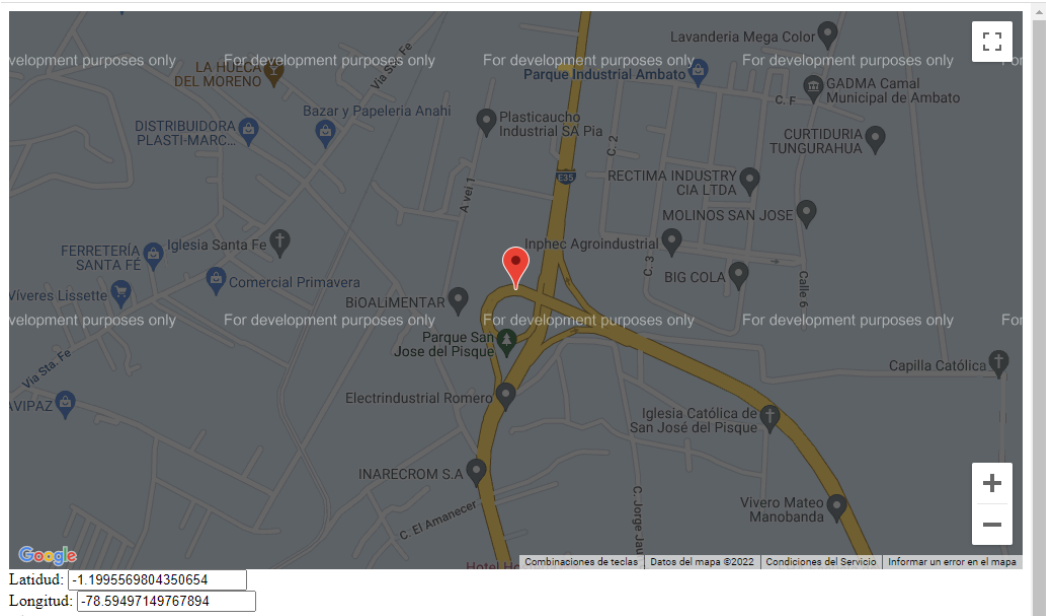


Figura 41 Pantalla obtener coordenadas  
Elaborado por: Franklin Lesano.

## Reportes

La pantalla de reportes toma como parámetros la fecha y la línea que desea ver, a través de los webs services realiza la consulta de cuantas veces recorrido ese bus la línea, si no hay datos se retornara un mensaje que no hay datos, si se selecciona uno de los registros de ruta abre los puntos por el cual recorrido, cuando selecciona uno de estos registros se marcara en el mapa el punto, hay dos opciones mas que permite graficas o por puntos o líneas el recorrido del bus.

## Rutas por Fechas

Los parámetros de entrada iniciales se basan en la fecha y la línea para filtrar las rutas iniciadas en la fecha seleccionada, si existen registros traerá los datos y se mostrará en una tabla con todos los registros.

## Back-End

```
public async void TraerLinea(String fecha, int IdLinea)
```

```
{
    try
    {
        string paramRuta = ""
        var ResultadoRuta = await ServiciosAPI.ObtenerListas();
```

```

CRuta miruta = new CRuta();
List<CRuta> ListaMiruta = new List<CRuta>();
foreach (CRuta item in ResultadoRuta)
{
    miruta = item;
    ListaMiruta.Add(miruta);
    miruta = new CRuta();
}
BindDataLinea(ListaMiruta);
}
catch
{
}
}

```

FronD-End



Figura 42 Pantalla selección de línea por fecha  
Elaborado por: Franklin Lesano.

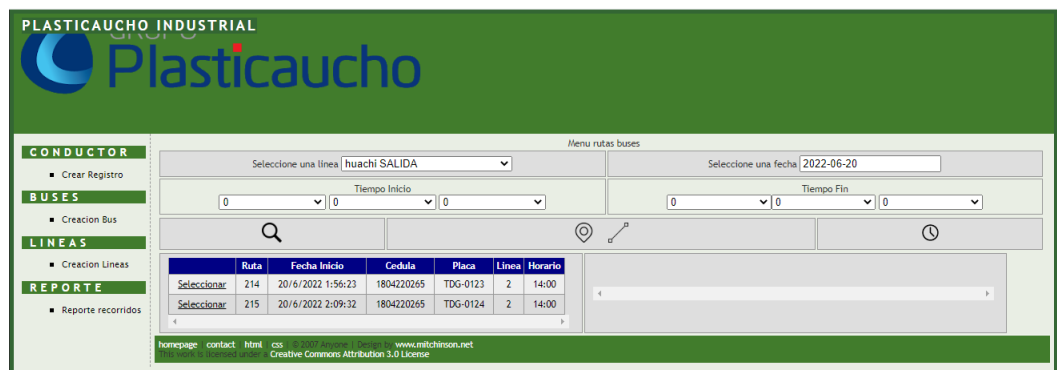


Figura 43 Pantalla visualización de rutas  
Elaborado por: Franklin Lesano.

## Visualización de los Puntos

Al seleccionar un registro del listado de las rutas se va a mostrar todos los puntos registrados por el bus en su trayecto.

Back-End

```
private async void BindDataPuntos(int idRuta)
{
    List<CRecorrido> ListaMiRecorrido = new List<CRecorrido>();
    DataTable dt = new DataTable();
    dt.Columns.AddRange(new DataColumn[3] {
        new DataColumn("Hora ", typeof(string)),
        new DataColumn("Longitud ", typeof(string)),
        new DataColumn("Latitud ", typeof(string))});
    ListaMiRecorrido = await ServiciosAPI.ObtenerListas
    Session["ListaMiRecorrido"] = ListaMiRecorrido;
    string Cadena = "";
    int c = 0;
    foreach (CRecorrido listaItem in ListaMiRecorrido)
    {
        dt.Rows.Add(listaItem.Hora_Recorrido1.ToString(),
            listaItem.Latitud_Recorrido1.ToString(),
            listaItem.Longitud_Recorrido1.ToString()
        );
        if (c == 0)
        {
            Cadena = Cadena + listaItem.Latitud_Recorrido1.ToString() + "|" +
            listaItem.Longitud_Recorrido1.ToString() + "|" +
            listaItem.Hora_Recorrido1.ToString();
            Cadena = Cadena + "|";
        }
        c++;
        if (c == 5)
            c = 0;
    }
}
```

```

}
Session["ListaPuntosGeograficos"] = Cadena;
gvPuntos.DataSource = dt;
gvPuntos.DataBind();
}

```

Front-End

	Ruta	Fecha Inicio	Cedula	Placa	Linea	Horario
<a href="#">Seleccionar</a>	214	20/6/2022 1:56:23	1804220265	TDG-0123	2	14:00
<a href="#">Seleccionar</a>	215	20/6/2022 2:09:32	1804220265	TDG-0124	2	14:00

Figura 44 Selección de ruta  
Elaborado por: Franklin Lesano.

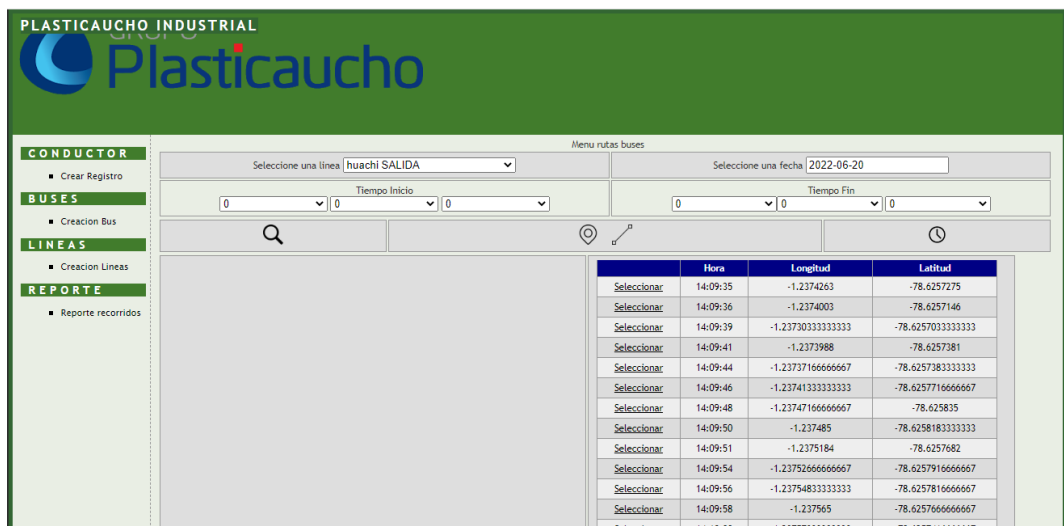


Figura 45 Pantalla de lista de coordenadas  
Elaborado por: Franklin Lesano.

## Segmentación de Tiempo

La segmentación de datos existe 5 parámetros, hora inicial, minutos iniciales, segundos iniciales, hora final, minutos finales, segundos finales, se considera que el tiempo inicial debe ser menor al tiempo final.

Back-End

```

protected void segmentacionTiempo()
{
    int HoraI = int.Parse(dlHoraInicio.Selected.Value.ToString());

```

```

int MinutoI = int.Parse(dlMinutosInicio.SelectedValue.ToString());
int segundoI =
int.Parse(dlSegundosInicio.SelectedValue.ToString());
string _unTiempoInicio = HoraI+":" + MinutoI + ":" + segundoI;
DateTime UnTiempoInicio = DateTime.Parse(_unTiempoInicio);
int HoraF = int.Parse(dlHoraFin.SelectedValue.ToString());
int MinutoF = int.Parse(dlMinutosFin.SelectedValue.ToString());
int segundoF= int.Parse(dlsegundoFin.SelectedValue.ToString());
string _unTiempoFin = HoraF + ":" + MinutoF + ":" + segundoF;
DateTime UnTimpofin = DateTime.Parse(_unTiempoFin);
if(UnTiempoInicio>= UnTimpofin)
{
    string script = @"<script type='text/javascript'>
        alert('La hora de inicio no puede ser mayor que la hora de fin');
    </script>";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
    "alerta", script, false);
}
else
{
    List< CRecorrido >nuevaSecmentacion=Session["ListaMiRecorrido"]
    as List<CRecorrido>;
    var nuevaSecmentacion1 = nuevaSecmentacion.Where(valor1=>
    DateTime.Parse(valor1.Hora_Recorrido1)>= UnTiempoInicio &&
    DateTime.Parse(valor1.Hora_Recorrido1) <= UnTimpofin );
    string Cadena = "";
    DataTable dt = new DataTable();
    dt.Columns.AddRange(new DataColumn[3] {
        new DataColumn("Hora ", typeof(string)),
        new DataColumn("Longitud ", typeof(string)),
        new DataColumn("Latitud ", typeof(string))});
    foreach (CRecorrido listaItem in nuevaSecmentacion1)
    {

```

```

dt.Rows.Add(
    listaItem.Hora_Recorrido1.ToString(),
    listaItem.Latitud_Recorrido1.ToString(),
    listaItem.Longitud_Recorrido1.ToString()
);
Cadena = Cadena + listaItem.Latitud_Recorrido1.ToString() + "|" +
listaItem.Longitud_Recorrido1.ToString() + "|" +
listaItem.Hora_Recorrido1.ToString();
Cadena = Cadena + "|";
}
Session["ListaPuntosGeograficos"] = Cadena;
gvPuntos.DataSource = dt;
gvPuntos.DataBind();
}
}
}

```

FronD-End

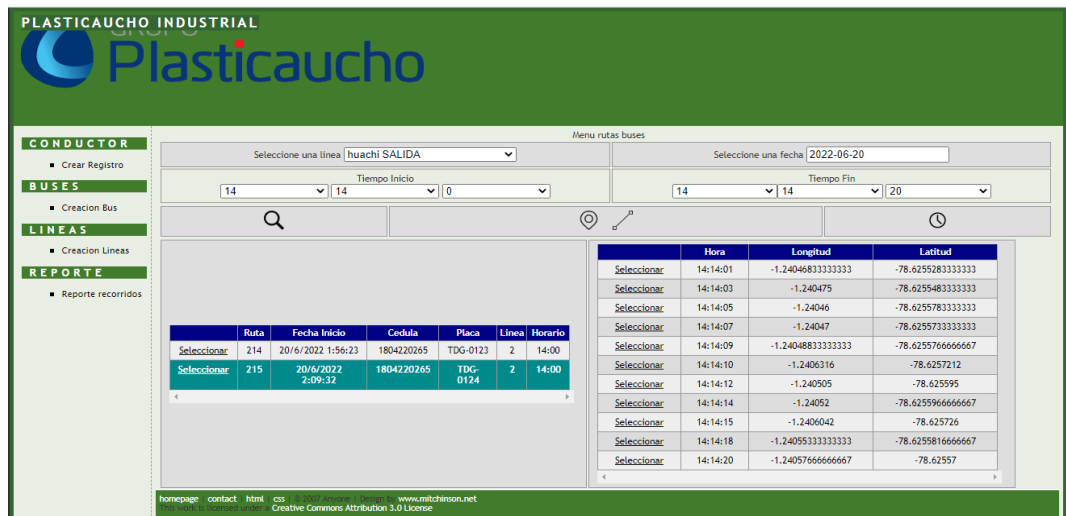


Figura 46 Pantalla de coordenadas segmentadas  
Elaborado por: Franklin Lesano.

## Graficación por Puntos

Grafica en el mapa una lista de puntos de acuerdo con la ruta seleccionada.

Back-End

```
<script type="text/javascript">
```



```

function initMap()
{
    var TipoVista = "";
    var soloPunto = "";
    var ListaPunto = "";
    if (TipoVista == "SI") {
        const flightPlanCoordinates = [];
        if (ListaPunto.length > 0)
        {
            var filaUno = ListaPunto.split('|');
            var InicialCentroPunto = filaUno[0].split('|');
            var Latitude1 = parseFloat(InicialCentroPunto[0]);
            var longitude1 = parseFloat(InicialCentroPunto[1]);
            const uluru = { lat: Latitude1, lng: longitude1 };
            const map = new
                google.maps.Map(document.getElementById("map3"),
                    {
                        zoom: 15,
                        center: uluru
                    });
            var filasPuntos = ListaPunto.split('|');
            for (i = 0; i < filasPuntos.length-1; i++)
            {
                var puntoGraficar = filasPuntos[i].split('|');
                var LatitudeG = parseFloat(puntoGraficar[0]);
                var longitudeG = parseFloat(puntoGraficar[1]);
                flightPlanCoordinates.push(new google.maps.LatLng(LatitudeG,
                    longitudeG));
            }
            const flightPath = new google.maps.Polyline({
                path: flightPlanCoordinates,
                geodesic: true,
                strokeColor: "#FF0000",

```

```

        strokeOpacity: 1.0,
        strokeWeight: 8,
    });
    flightPath.setMap(map);
}
}
else
{
    if (soloPunto.length > 0) {
        var name = "";
        var latitudevar = name.split('|');
        var LatitudeG = parseFloat(latitudevar[0]);
        var longitudeG = parseFloat(latitudevar[1]);
        const uluru = { lat: LatitudeG, lng: longitudeG };
        const map = new
        google.maps.Map(document.getElementById("map3"),
        {
            zoom: 15,
            center: uluru
        });
        const punto = new google.maps.Marker({
            position: uluru,
            map: map,
        });
    }
    else {
        if (ListaPunto.length > 0)
        {
            var filaUno = ListaPunto.split('|');
            var InicialCentroPunto = filaUno[0].split('|');
            var Latitude1 = parseFloat(InicialCentroPunto[0]);
            var longitude1 = parseFloat(InicialCentroPunto[1]);
            const uluru = { lat: Latitude1, lng: longitude1 };

```

```

const map = new
google.maps.Map(document.getElementById("map3"),
  {
    zoom: 15,
    center: uluru
  });
var filasPuntos = ListaPunto.split(',');
for (i = 0; i < filasPuntos.length; i++)
{
  var puntoGraficar = filasPuntos[i].split('|');
  var LatitudeG = parseFloat(puntoGraficar[0]);
  var longitudeG = parseFloat(puntoGraficar[1]);
  const uluru = { lat: LatitudeG, lng: longitudeG };
  // The map, centered at Uluru
  const punto = new google.maps.Marker({
    position: uluru,
    map: map,
  });
}
}
}
}
}
window.initMap = initMap;
</script>

```

FronD-End

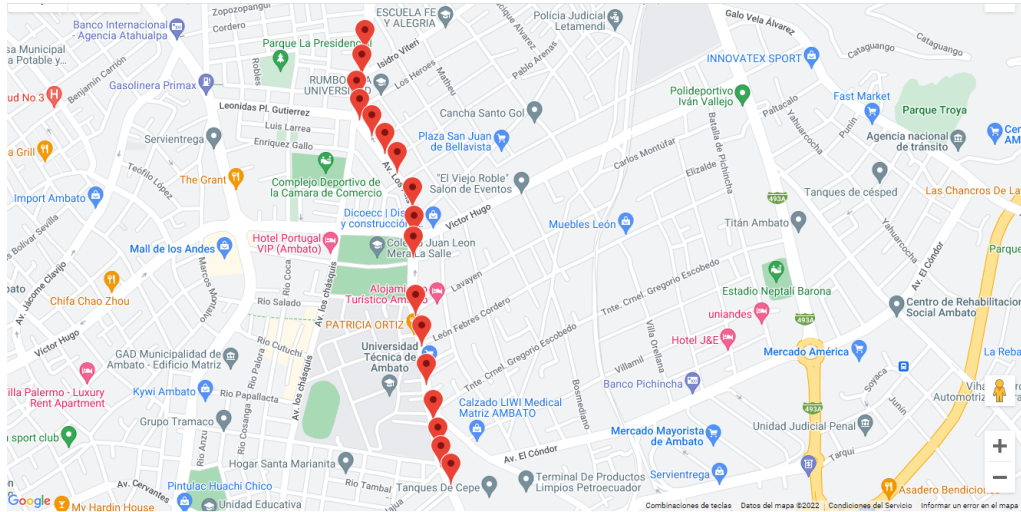


Figura 47 Pantalla de lista de coordenadas graficadas en puntos  
Elaborado por: Franklin Lesano.

### Graficación por líneas

Grafica en el mapa una línea de acuerdo con la ruta seleccionada.

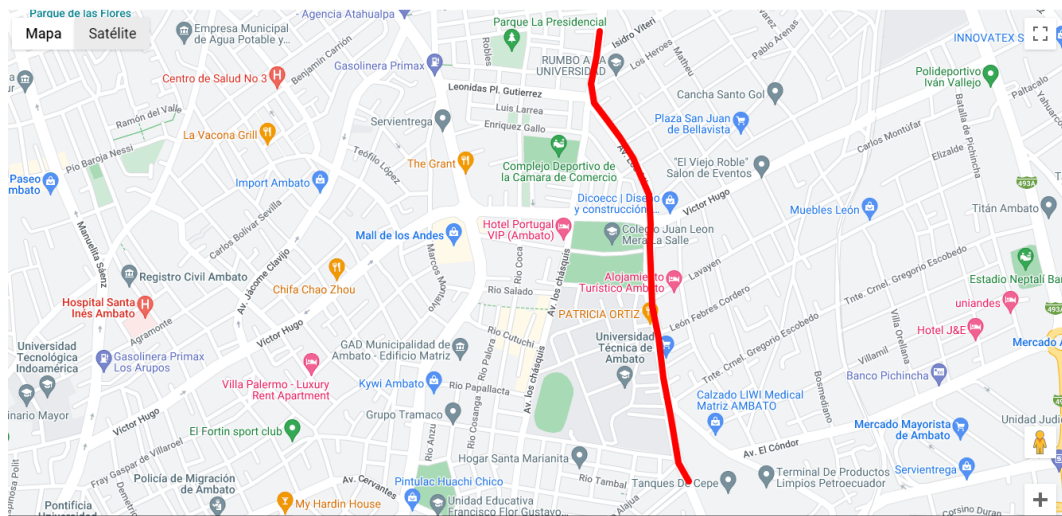
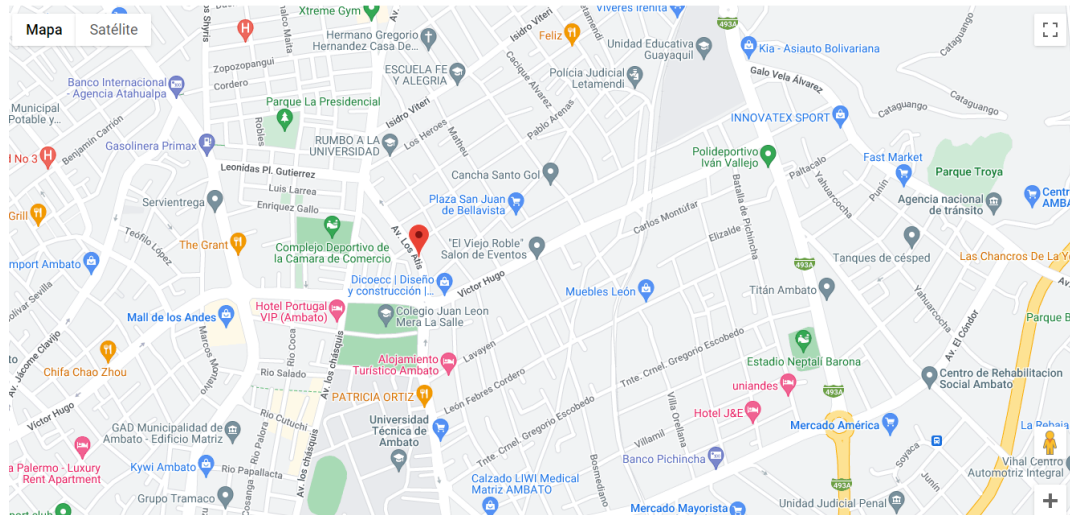


Figura 48 Pantalla de lista de coordenadas graficadas en línea  
Elaborado por: Franklin Lesano.

### Graficación punto

Grafica en el mapa un solo punto seleccionado de la lista del recorrido realizado en la ruta.



*Figura 49 Pantalla graficación punto único  
Elaborado por: Franklin Lesano.*

### **3.3.4. Fase 3 Pruebas**

El objetivo de las pruebas es validar el correcto funcionamiento del sistema en conjunto, esperando que los resultados deseados sean los más esperados, basados en las historias de usuarios los requerimientos deben ir de la mano con el sistema, para ello las pruebas de aceptación se subirán al entorno de pruebas donde serán verificadas por el cliente para luego dar el paso a producción.

#### **3.3.4.1. Configuración del entorno de pruebas**

##### **Alojamiento APIs**

En la configuración del entorno de pruebas se utilizó un hosting que permite el alojamiento de las Webs APIs desarrollado en Net Framework de forma gratuita

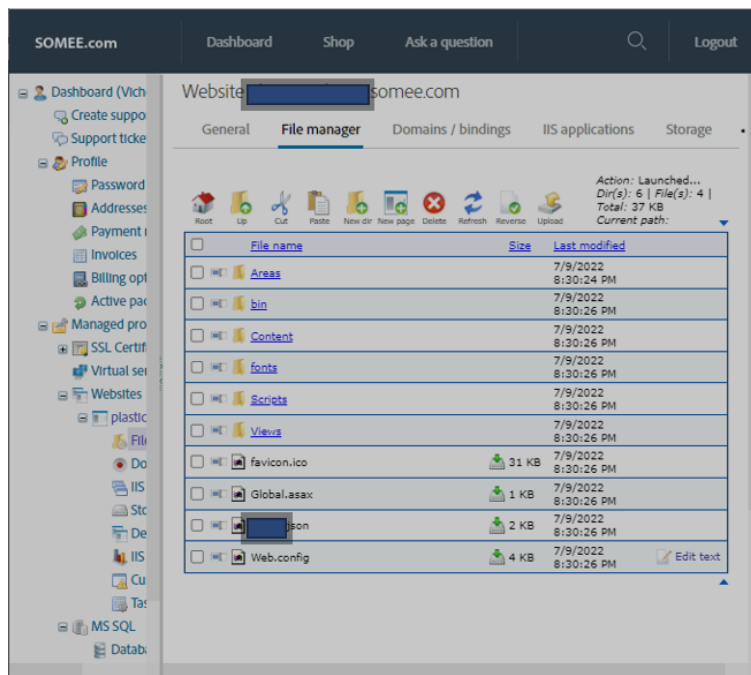


Figura 50 Ambiente de pruebas apis  
Elaborado por: Franklin Lesano.

## Alojamiento Base de Datos

En el alojamiento de la base de datos se utilizó un hosting gratuito que permite el alojamiento y gestión de bases de datos PostgreSQL.

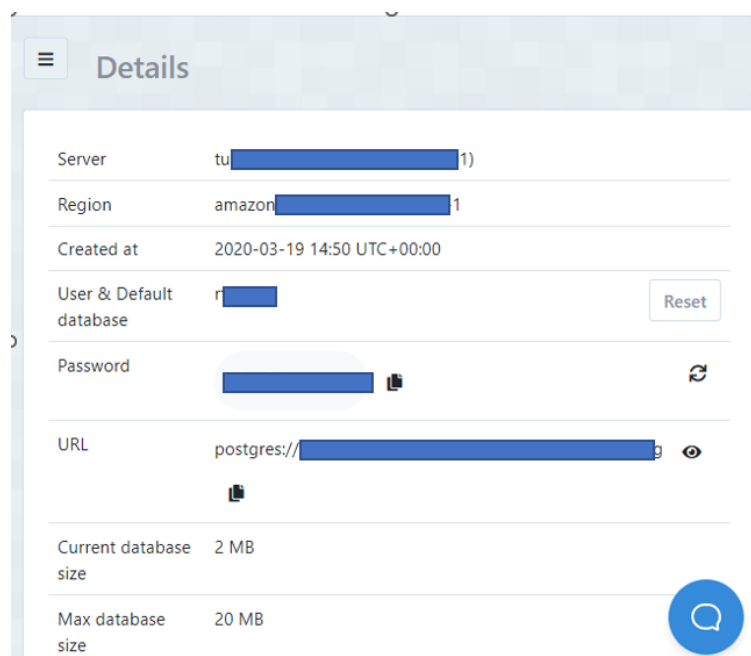
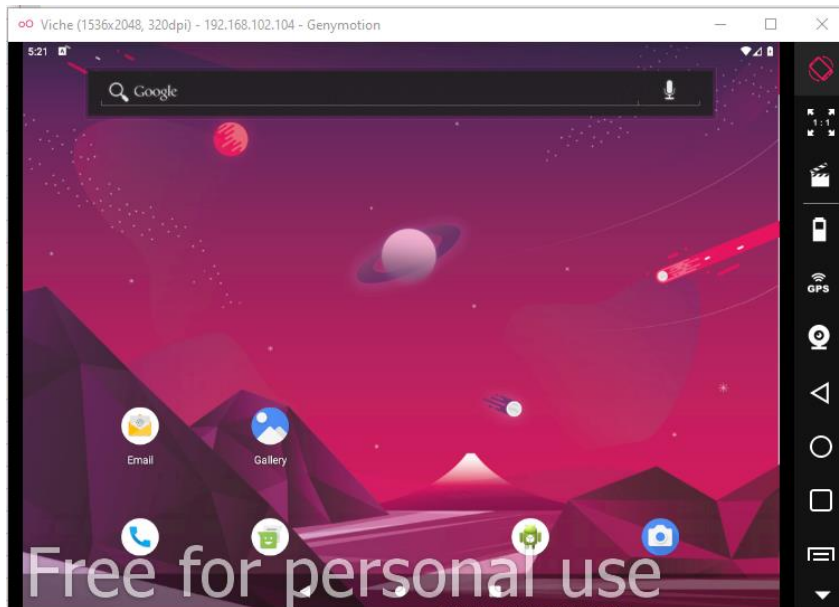


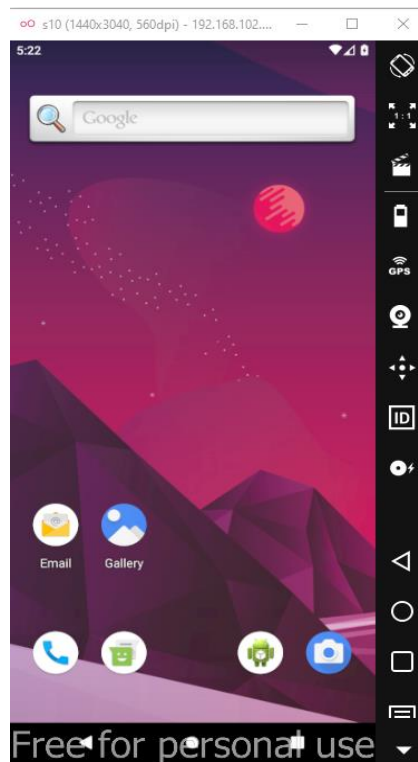
Figura 51 Ambiente de pruebas base de datos  
Elaborado por: Franklin Lesano.

## Emuladores Virtuales

Para las aplicaciones se utilizó emuladores virtuales donde se alojó las aplicaciones utilizando Genymotion.



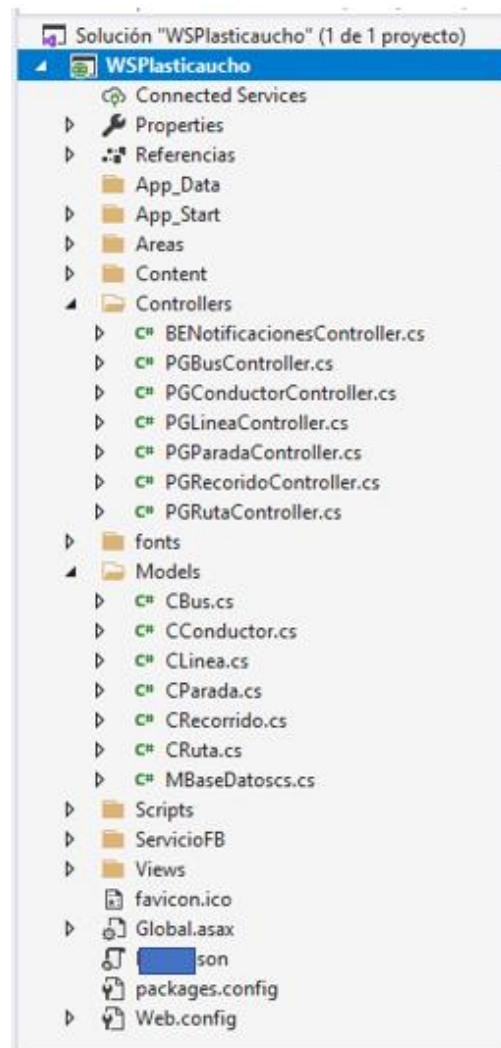
*Figura 52 Emulador tipo Tablet  
Elaborado por: Franklin Lesano.*



*Figura 53 Emulador tipo celular  
Elaborado por: Franklin Lesano.*

## APIs web services

La arquitectura para la construcción de los servicios API Rest, están basados en los controladores quienes exponen los métodos a la hacia la web, separados la conexión a la base de datos, modelo de las clases y controller, la principal función es tener ordenada y segmentado las diferentes funcionalidades.

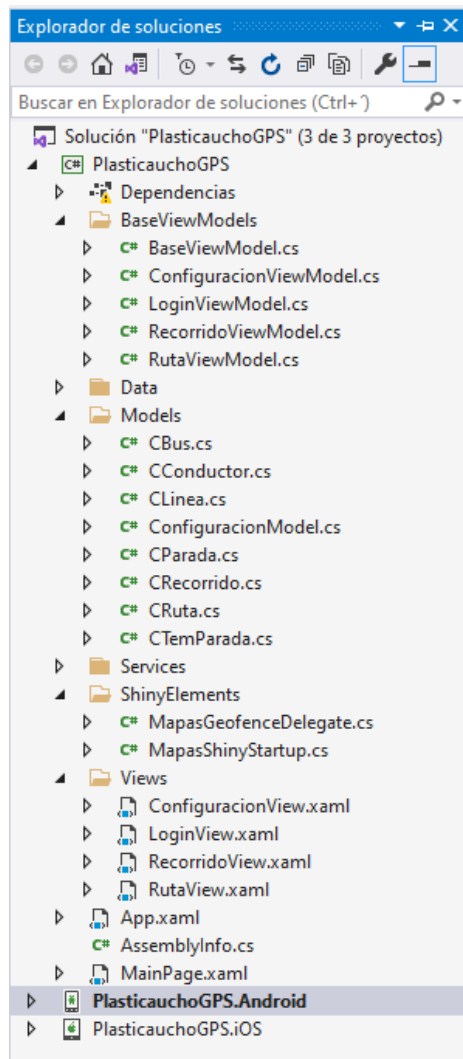


*Figura 54 Apis web services  
Elaborado por: Franklin Lesano.*

## App movil conductor

El proyecto desarrollado para el aplicativo del conductor se base con el patrón de diseño MVVM(Modelo Vista Vista Modelo) con la finalidad de blindar los controles y acceder de forma más rápida y optimizando tiempos y código, se segmento en carpetas las vistas, modelos, clases y viewmodels.





*Figura 55 Estructura del proyecto app móvil conductor  
Elaborado por: Franklin Lesano.*

## **Sistema de gestión**

El proyecto está definido por defecto por Visual estudio 2019 para sitios desarrollados en As.Net, los accesos a los datos y las vistas ese encuentran separadas por carpetas de acuerdo con cada una de las funcionalidades que realiza.

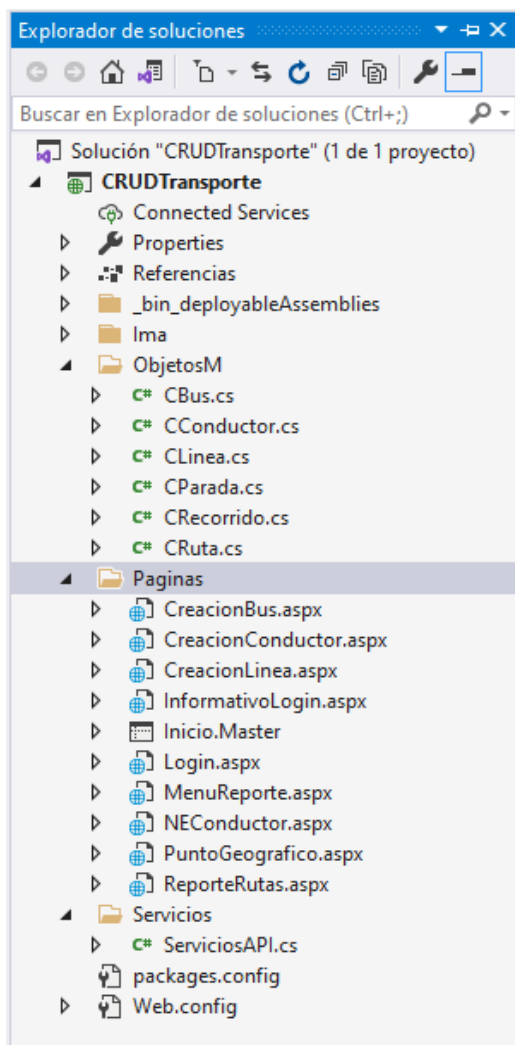


Figura 56 Estructura de proyecto gestión administrativa  
 Elaborado por: Franklin Lesano.

### 3.3.4.2. Pruebas de aceptación

En la fase de pruebas utilizara el siguiente formato.

<b>Pruebas de Aceptación</b>	
Número:	Historia de Usuario:
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Pasos:	
Resultado Esperado:	
Evaluación de la Prueba:	

Tabla N.- 87 Modelo de prueba de aceptación  
 Elaborado por: Franklin Lesano.

**Número:** Número de asignación a cada prueba.

**Historia de usuario:** Número de la historia de usuario correspondiente a la historia de usuario.

**Nombre:** Etiqueta asignada a la prueba de aceptación.

**Descripción:** Describe que debe cumplir la prueba de aceptación.

**Condiciones de Ejecución: Especificación** o conjunto de especificaciones para que la prueba de aceptación sea correcta.

**Pasos:** Conjunto de procedimientos para que se cumpla la prueba.

**Resultado Esperado:** Información obtenida del usuario al realizar la prueba.

**Evaluación de la prueba:** Resultado (Satisfactorio - No Satisfactorio)

<b>Pruebas de Aceptación</b>	
Número: 1	Historia de Usuario: 012
Nombre: Login Conductor	
Descripción: Autenticación conductor	
Condiciones de Ejecución: El conductor debe ingresar su cedula como usuario y una clave.	
Pasos: Primero ingresar el usuario, luego ingresar una clave, presionar el botón Ingresar.	
Resultado Esperado: Si existe el conductor en la base dará paso a la siguiente pantalla, caso contrario emite un mensaje que no existe o si existe algún fallo emite una alertade error.	
Evaluación de la Prueba: Satisfactorio	

*Tabla N.- 88 Prueba de aceptación 01  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 2	Historia de Usuario: 013
Nombre: Inicio Ruta	

Descripción: Ingreso datos para la ruta
Condiciones de Ejecución: El conductor debe elegir, una placa de bus, una línea, un sentido y por último el horario, debe presionar aceptar o cancelar, se debe mostrar que es necesario activar la ubicación del dispositivo.
Pasos: Primero se debe elegir de la lista un bus, luego de ello se debe escoger el nombre de la línea a recorrer, el sentido de la ruta y por último el horario, presionar aceptar.
Resultado Esperado: El sistema autentificara si existe y son coherentes los datos ingresados, si es verdaderos ingresara un registro y devolverá el identificador único para esa ruta nueva, caso contrario generara un error.
Evaluación de la Prueba: Satisfactorio

*Tabla N.- 89 Prueba de aceptación 02  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 3	Historia de Usuario: 014
Nombre: Navegación conductor	
Descripción: Navegación de la ruta conductor	
Condiciones de Ejecución: La pantalla debe tener un id de ruta previamente para traer las paradas, la navegación de las paradas depende del orden asignado en el crud de la parada, para las notificaciones simplemente se enviarán de acuerdo con la parada actual y al tipo de incidente, Se debe configurar la pantalla para que no se bloquee.	
Pasos: Presionar los botones de incidentes o la de salir si termino la ruta.	
Resultado Esperado: Si el bus se encuentra en la parada asignada se activará una notificación auditiva para el conductor indicando que ha llegado a la parada, y una notificación local de la siguiente parada, en las notificaciones de incidente, si hay un resultado de envío retornara un true caso contrario notificara que existió un	

error, la opción de salir al presionar debe parar el servicio de la geocercas y salir de la aplicación.
Evaluación de la Prueba: Satisfactorio

*Tabla N.- 90 Prueba de aceptación 03  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 4	Historia de Usuario: 015
Nombre: Seguimiento ruta pasajeros	
Descripción: Seguimiento del vehículo en Google Maps	
Condiciones de Ejecución: El pasajero debe elegir el nombre de la línea, el sentido de la ruta y por último el horario, al presionar en el botón ver se iniciará la graficación Google maps.	
Pasos: Primero se debe elegir una línea, segundo un sentido y por último el horario.	
Resultado Esperado: Si los datos ingresados existen en l base de datos devolverá el id de la ruta buscada, el cual ara una búsqueda el ultimo registro de esa ruta, caso contrario mostrará que no existe datos que mostrar.	
Evaluación de la Prueba: Satisfactorio	

*Tabla N.- 91 Prueba de aceptación 04  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 5	Historia de Usuario: 016
Nombre: Pantalla Conductor	
Descripción: gestión de la tabla conductor.	
Condiciones de Ejecución: El sistema mostrara una pantalla donde el gestor ingresara la información en los campos necesarios para crear un nuevo registro o actualizar, si se desea eliminar un registro se pedirá confirmación, la lista de conductores mostrara solo los registros activos en la base de datos.	

Pasos: Presionar el botón nuevo para el ingreso de nuevos registros, el botón de actualizar en caso de actualizar y eliminar si desea inactivar un registro, los campos se debe ingresar previamente.
Resultado Esperado: La interfaz muestra un mensaje de confirmación si se ingresaron correctamente los datos del registro caso contrario retornará un mensaje de negación del proceso, una vez ingresado el listado de los conductores se actualizará correctamente.
Evaluación de la Prueba: Satisfactorio

*Tabla N.- 92 Prueba de aceptación 05*

*Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 6	Historia de Usuario: 017
Nombre: Pantalla Bus	
Descripción: gestión de la tabla Bus.	
Condiciones de Ejecución: El sistema mostrara una pantalla donde el gestor ingresara la información en los campos necesarios para crear un nuevo registro o actualizar, si se desea eliminar un registro se pedirá confirmación, la lista de buses mostrara solo los registros activos en la base de datos.	
Pasos: Presionar el botón nuevo para el ingreso de nuevos registros, el botón de actualizar en caso de actualizar y eliminar si desea inactivar un registro, los campos se debe ingresar previamente.	
Resultado Esperado: La interfaz muestra un mensaje de confirmación si se ingresaron correctamente los datos del registro caso contrario retornará un mensaje de negación del proceso, una vez ingresado el listado de los buses se actualizará correctamente.	
Evaluación de la Prueba: Satisfactorio	

*Tabla N.- 93 Prueba de aceptación 06*

*Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 7	Historia de Usuario: 018
Nombre: Pantalla Líneas	

Descripción: gestión de la tabla línea.
Condiciones de Ejecución: El sistema mostrará una pantalla donde el gestor ingresará la información en los campos necesarios para crear un nuevo registro o actualizar, si se desea eliminar un registro se pedirá confirmación, si existen datos no se podrán eliminar por integridad de la información, una vez terminado se mostrará la lista de las líneas.
Pasos: Presionar el botón nuevo para el ingreso de nuevos registros, el botón de actualizar en caso de actualizar y eliminar si desea inactivar un registro, los campos se debe ingresar previamente.
Resultado Esperado: La interfaz muestra un mensaje de confirmación si se ingresaron correctamente los datos del registro caso contrario retornará un mensaje de negación del proceso, el listado de la línea se actualizará si los resultados son correctos.
Evaluación de la Prueba: Satisfactorio

*Tabla N.- 94 Prueba de aceptación 07  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 8	Historia de Usuario: 019
Nombre: Pantalla Paradas	
Descripción: gestión de la tabla parada.	
Condiciones de Ejecución: El sistema mostrara una pantalla donde el gestor ingresara la información en los campos necesarios para crear un nuevo registro o actualizar, si se desea eliminar un registro se pedir confirmación, si existen datos no se podrán eliminar por integridad de la información, una vez terminado se mostrará la lista de las Paradas.	
Pasos: Presionar el botón nuevo para el ingreso de nuevos registros, el botón de actualizar en caso de actualizar y eliminar si desea inactivar un registro, los campos se debe ingresar previamente.	
Resultado Esperado: La interfaz muestra un mensaje de confirmación si se ingresaron correctamente los datos del registro	

caso contrario retornará un mensaje de negación del proceso, el listado de las paradas se actualizará si los resultados son correctos.
Evaluación de la Prueba: Satisfactorio

*Tabla N.- 95 Prueba de aceptación 08  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 9	Historia de Usuario:020
Nombre: Reportes Datos	
Descripción: Reporte rutas iniciadas	
Condiciones de Ejecución: Seleccionar la línea y la fecha, presionar el icono de la lupa	
Pasos: Del listado de las líneas seleccionar una línea, seleccionar del calendario la fecha buscada, presionar el icono de la lupa	
Resultado Esperado: Si existen datos de las líneas iniciadas devolverá un listado de las rutas con la línea y la fecha seleccionados previamente, caso contrario mostrara un mensaje que no existen datos.	
Evaluación de la Prueba: Satisfactorio	

*Tabla N.- 96 Prueba de aceptación 09  
Elaborado por: Franklin Lesano.*

<b>Pruebas de Aceptación</b>	
Número: 10	Historia de Usuario: 020
Nombre: Reportes Graficación	
Descripción: Graficación de los puntos geográficos	
Condiciones de Ejecución: Seleccionar el icono de línea para graficar el listado de coordenadas geográficas, presionar el icono de puntos, para graficar en forma de puntos en Google maps, si no existe datos se mostrará una pantalla en blanco.	
Pasos: Seleccionar el tipo de grafico que desea mostrar en Google maps entre las opciones de puntos, líneas o punto individual.	



Resultado Esperado: Si existe datos se mostrará el tipo de graficación de acuerdo con lo seleccionado, caso contrario mostrará un mapa en blanco.
---

Evaluación de la Prueba: Satisfactorio
--

*Tabla N.- 97 Prueba de aceptación 10*  
***Elaborado por: Franklin Lesano.***

## **CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

Con las entrevistas realizadas a la parte administrativa de servicios laborales, las encuestas realizadas a los conductores se pudo obtener la información necesaria para armar los procesos que luego se transformaron en funcionalidades para el sistema, la observación y el análisis ayudo a el desarrollo de los requerimientos iniciales y los ajustes en el transcurso del desarrollo del aplicativo.

La arquitectura del sistema se estableció como tradicional donde tenemos un cliente servidor, del lado del cliente tenemos el aplicativo móvil, el gestor del sistema del lado del servidor tendremos almacenados los servicios web y la base de datos, en el aplicativo se pudo hacer el seguimiento a través de obtener las coordenadas y enviar la información cada 5 segundos.

El despliegue del sistema GPS se generó un archivo Apk que corre sobre un sistema operativo Android, el aplicativo registra de manera exitosa y con una precisión alta, mejorando en la parte operativa que puedan gestionar y anticipar su línea, reducir los retrasos y desde la aparte corporativa reducir costos de retribución o justificación por atrasos.

El desarrollo del aplicativo móvil recoge los puntos geográficos a través del GPS del dispositivo móvil, los envía a una base de datos y después es recuperado para visualizar en un mapa en tiempo real el bus, con esto la administración puede sacar reportes de los recorridos y adicionalmente los pasajeros pueden ver la ubicación de su línea para salida del trabajo o llegada a la compañía Plasticaucho Industrial S.A Ambato.

## 4.2. Recomendaciones

En cuanto al complemento de Shiny se recomienda que no se registre más de 100 coordenadas geográficas, en este caso del aplicativo no se aceptaría más del límite de paradas, ya que después de los 100 registros el complemento comienza a fallar y a consumir recursos del dispositivo móvil haciendo que se caliente y el sistema no correrá de manera eficaz.

Se recomienda que las notificaciones no excedan de las 100 diarias ya que después de este límite Google notificara que se alcanza el límite diario, y se procederá a bloquear y que ingrese una tarjeta para el cobro para seguir utilizando.

Firestore no garantiza el 100% de las entregas de las notificaciones, por lo que se recomienda que si no llegan todas las notificaciones se revise la consola de Firestore sección de Cloud Analytics para revisar el evento que genere que no se entrega, Firestore garantiza el 95% de efectividad en la entrega de sus notificaciones Push.

Las claves generadas para la utilización de los mapas tienen un costo gratuito y un costo pagado, dependiendo si se lanza a producción y el número de visitas, por lo que se recomienda que no se tengan más de 100 mapas abiertos para evitar el costo por utilizar ya que después de superar el límite diario no podrán visualizar los mapas hasta el inicio de un nuevo día.

Se recomienda en la parte de los dispositivos móviles tener mínimo Android 7 y que soporte GPS tener una carga mínima de uno 80% de batería ya que el consumo del gps consume energía por la alta precisión que necesita al obtener las coordenadas geográficas.

## **BIBLIOGRAFÍA:**

- [1] M. L. Chaparro Ariza y N. R. Guzman Diaz, «Desarrollo de aplicación móvil de transporte entre la comunidad universitaria con capacidad de geolocalización para el proyecto ud sobre ruedas,» Universidad Distrital Francisco José de Caldas, Bogotá, 2017.
- [2] J. Camargo, L. Gonzalez, D. Segura, F. Garay y N. Rincon, «Orientacion de pasajeros con discapacidad visual dentro del sistema de transporte masivo Transmilenio, mediante geolocalizacion satelital,» *Revista Ingenieria*, pp. 283-297, 2017.
- [3] R. P. Asencio Vargas, «Desarrollo de un Sistema Web y Aplicativo Android Para el Control y Monitoreo por Geolocalización de Unidades de Transportación de Carga Pesada para la Empresa Transporte y Logística Honores Méndez,» Universidad de Guayaquil, Ecuador - Guayaquil, 2017.
- [4] E. D. Jaramillo Zambrano, «Desarrollo de Aplicación Móvil, con Geolocalización de Líneas de Autobuses y sus paradas para el Gobierno Autónomo Descentralizado Municipalidad de Ambato.,» Universidad Técnica de Ambato, Ambato, 2018.
- [5] E. F. Vega Obando, «Desarrollo de un sistema de geilocalización para monitorear los vehículos de transporte de la Cooperativa Andina,» Universidad de las Fuerzas Armadas ESPE, Sangolqui, 2017.
- [6] A. Castro Alvares, «Repositorio de la Univer,» Universidad de Guayaquil, Guayaquil, Desarrollo De Sistema Gps Para Geolocalización Y Control De Ruta De La Flota Perteneciente A Una Línea De Transporte Público Vía Internet.
- [7] C. A. Elias, «Aplicacion movil de Geolocalización para el control y la gestión de la seguridad en conductores de la empresa de Taxi Exitodo E.I.R.L. Juliaca 2019,» Universidad Nacional de Altiplano, Puno – Perú, 2019.

- [8] G. Beltrán, Geolocalización Online. La importancia del dónde., Barcelona: Oberta UOC Publishing,SL, 2016.
- [9] G. Beltrán López, Geomarketing: geolocalización, redes sociales y turismo., Madrid: Bubok Publishing S.L., 2018.
- [10] R. M. Cabrera, «Geolocalización del día a día,» TecnoCapus Mataro-Maresme, Barcelona, 2018.
- [11] E. J. Bernedo Tito, «Eficacia de a Implementación de Geocercas en el Tramo de la Carretera desde la Cantera-Tajo G3 hasta la Planta - Pet Coke, para la Reducción de Accidentes en el Transporte Interno de Caliza en la Empresa Minera no Metálica Cal y Cemento Sur S.A.,» Universidad Tecnológica de Perú, Perú, 2021.
- [12] J. Hugon, C#7: Desarrolle aplicaciones Windows con Visual Studio 2017, EDICIONES ENI, 2018, p. 1066.
- [13] Microsoft, «Le damos la bienvenida al IDE de Visual Studio,» 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>. [Último acceso: 4 4 2022].
- [14] M. P. Zea Ordóñez, J. R. Molina Ríos y F. F. Redrován Castillo, Administración de base de datos con ProgresQL, España: Editorial Área de Innovación y Desarrollo, S.I., 2017.
- [15] microsoft.com, «¿Qué es PostgreSQL?,» [En línea]. Available: <https://azure.microsoft.com/es-mx/overview/what-is-postgresql/>. [Último acceso: 02 05 2022].
- [16] S. Sanderson, Pro ASP.NET MVC 2 Framework, New York: Editorial Board, 2010.

- [17] microsoft.com, «Información general de ASP.NET Core MVC,» [En línea]. Available: <https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-6.0>. [Último acceso: 10 04 2022].
- [18] G. Taskos, *Xamarin Cross-Platform Development Cookbook*, Reino Unido: Editorial Packt Publishing, 2016.
- [19] microsoft.com, «¿Qué es Xamarin.Forms?,» [En línea]. Available: <https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin-forms>. [Último acceso: 27 03 2022].
- [20] H. V. Cevallos, B. Molina Montero y J. Dávila Cuesta, «Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,» *Espirales*, p. 114, 2018.
- [21] J. Garcia, «Metodologías Ágiles en el Desarrollo,» *Universidad Politécnica de Valencia*, pp. 2,3.
- [22] C. Rodríguez y R. Dorad, «¿Por qué implementar Scrum?,» *Revista ONTARE*, vol. 3, n° 1, p. 125–144, 2015.
- [23] J. M. Fernandez y S. Cedelli, «Convivencia de metodologías: Scrum y Rup en un proyecto de gran escala,» Univeridad Nacional de la Plata, Argentina, 2014.
- [24] M. Trigas Gallego, «Metodología Scrum,» Universitat Oberta de Catalunya, Catalunya, 2012.
- [25] M. Bermejo, «El Kanban,» Universidad Oberta de Catalunya, España, 2011.
- [26] J. I. Gaete Rodriguez-Peña, «Propuesta para un nuevo enfoque de aplicación ágil con Scrum, LSD y Kanban,» Pontificia Universidad Católica de Valparaíso, Valparaíso, 2016.

- [27] B. Molina Montero, H. Vite Cevallos y J. Dávila Cuesta, «Metodologías ágiles frente a la tradicionales en el proceso de desarrollo de software,» *Espiraless revista multidisciplinaria de investigación*, pp. 113-121, 2018.
- [28] S. G. Rivadeneira Molina, «Metodología Ágiles enfocadas al modelado de requerimientos,» *Unidad Académica Rio Turbio*, vol. 1, nº 1, pp. 1-29, 2012.
- [29] E. Ruiz Larrocha, *Nuevas Tendencias en los Sistemas de Información.*, Madrid: Editorial Universitaria Ramón Areces, 2017.
- [30] Postgresql.org, «Postgresql.org,» 2022. [En línea]. Available: <https://www.postgresql.org/about/>. [Último acceso: 23 05 2022].
- [31] López Herrera Patricia, «Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL,» Universidad Autónoma del Estado de México, México, 2016.
- [32] J. J. León Soberón, «Análisis comparativo de Sistemas gestores de bases de datos PostgreSQL y MySQL en procesos CRUD,» Universidad Señor de Sipán, Perú, 2020.
- [33] J. Dominguez Chávez, «Trabajo con Vistas (views) en Maria DB y/o MySQL,» Universidad Politécnica Territorial del Estado Aragua, Venezuela, 2017.
- [34] microsoft.com, «Xamarin.Forms,» 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/xamarin/get-started/what-is-xamarin-forms>. [Último acceso: 20 05 2022].
- [35] flutter-dev, «Flutter,» 2022. [En línea]. Available: <https://esflutter.dev/docs/get-started/learn-more>. [Último acceso: 22 05 2022].
- [36] D. Villaón Pardo, «Crear y desarrollar una aplicación de alto rendimiento con bajo coste utilizando Flutter y Firebase.,» etsinf, Valencia, 2021.

- [37] reactnative.dev, «React Native,» Support Ukraine UA, 2022. [En línea]. Available: <https://reactnative.dev/docs/getting-started>. [Último acceso: 20 05 2022].
- [38] E. V. Macias Vera, «Estudio comparativo de los frameworks del desarrollo móvil nativo "Flutter" y "react Native" .,» Universidad Técnica de Babahoyo, Babahoyo, 2021.
- [39] C. Idalberto, Administración de Recursos Humanos, México: McGraw Hill / Internacional, 2011.



## Manual de usuario

Manual de instrucciones

2022



## Tabla de Contenidos

1. Aplicativo GPS conductor .....	168
1.1. Pantalla autentificacion .....	168
1.1.1. Pantalla autentificacion .....	168
1.2. Pantalla ruta.....	170
1.2.1. Área de clientes .....	170
1.2.2. Vista datos ingresados .....	172
1.3. Pantalla recorrido.....	173
1.3.1. Área de clientes .....	173
2. Aplicativo pasajeros .....	176
2.1. Pantalla del pasajero .....	176
3. Gestor Web .....	179
3.1. Pantalla de inicio.....	179
3.2. Pantalla gestion conductor .....	181
3.2.1. Pantalla ingreso informacion .....	182
3.3. Pantalla gestion buses .....	195
3.4. Pantalla gestion lineas y paradas .....	188
Document control.....	192
3.5. Pantalla reporteria .....	195
3.5.1. Graficacion por puntos .....	198
3.5.2. Graficacion por lineas .....	198

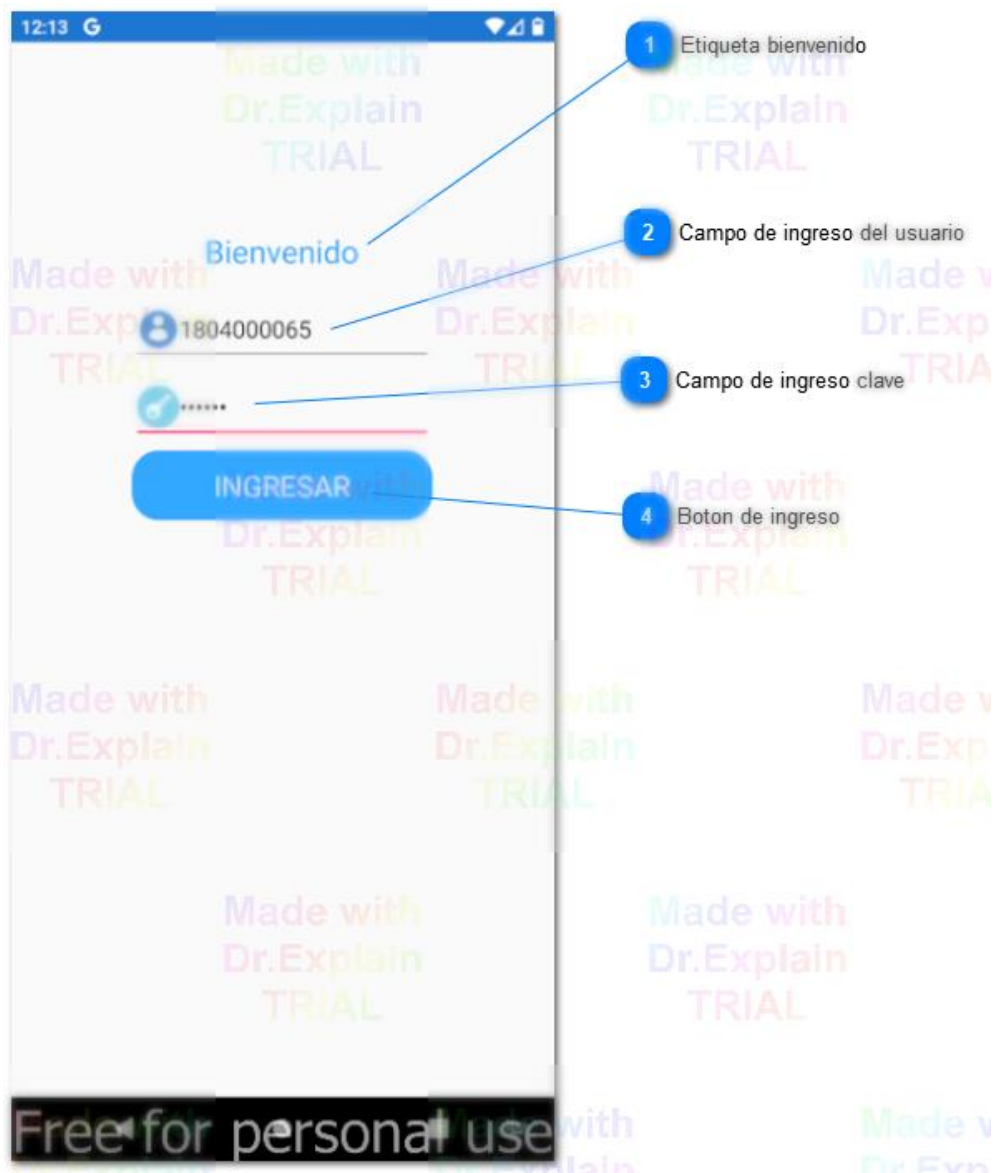
## 1. Aplicativo GPS conductor

El aplicativo móvil esta desarrollado para el seguimiento del vehículo, la app consta de funcionalidades como seguimiento de la ruta, advertencia de la parada actual y de la parada próxima, puede enviar notificaciones en caso de un incidente en su recorrido.

### 1.1 Pantalla autenticación

La primera pantalla del aplicativo móvil nos mostrara dos campos para autenticarse que es un conductor autorizado para empezar el recorrido.

#### 1.1.1 Pantalla autenticación



1

### Etiqueta bienvenida



Etiqueta de bienvenida al sistema para el conductor.

2

### Campo de ingreso del usuario



El campo de usuario ingresa la cedula del conductor, teniendo en cuenta que la longitud máxima permitida es de 0 caracteres numéricos.

3

### Campo de ingreso clave



El campo de clave permite el ingreso de caracteres números, alfabéticos y especiales, el formato está cifrado para evitar ver la clave

4

### Botón de ingreso



Al presionar el botón enviara una petición al servidor, i los datos son correctos mostrara la siguiente pantalla caso contrarió enviara un error como alerta al usuario.

## 1.2 Pantalla ruta

La pantalla de la ruta consta de cuatro campos necesarios, el primero se refiere a la placa del bus, el segundo campo se refiere al nombre de la línea, el tercero es el sentido de la trayectoria y por último el horario asignado.

### 1.2.1 Área de clientes



1

### Listado de placas de buses



Se desplegará un listado de placas de los buses activos en el sistema, luego de ello el conductor deberá escoger la placa de su unidad.

2

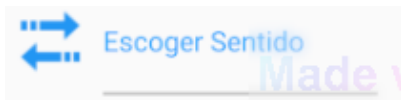
### Listado de líneas



Se desplegará un listado de las líneas activas en el sistema, el conductore deberá escoger la línea asignada para empezar el trayecto.

3

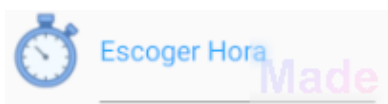
### Listado de orientación de la línea



Se desplegará un listado de la orientación, el conductore deberá escoger la orientación asignada para empezar el trayecto.

4

### Listado de horarios



Se desplegará un listado de los horarios, el conductore deberá escoger el horario asignada para empezar el trayecto.

5

### Botón salir



Al presionar el botón salir se cerrará el aplicativo.

6

### Botón ingresar



Al presionar el botón ingresará el sistema validará los datos y asignará una ruta para empezar el recorrido de la unidad, si existe un problema emitirá una alerta.

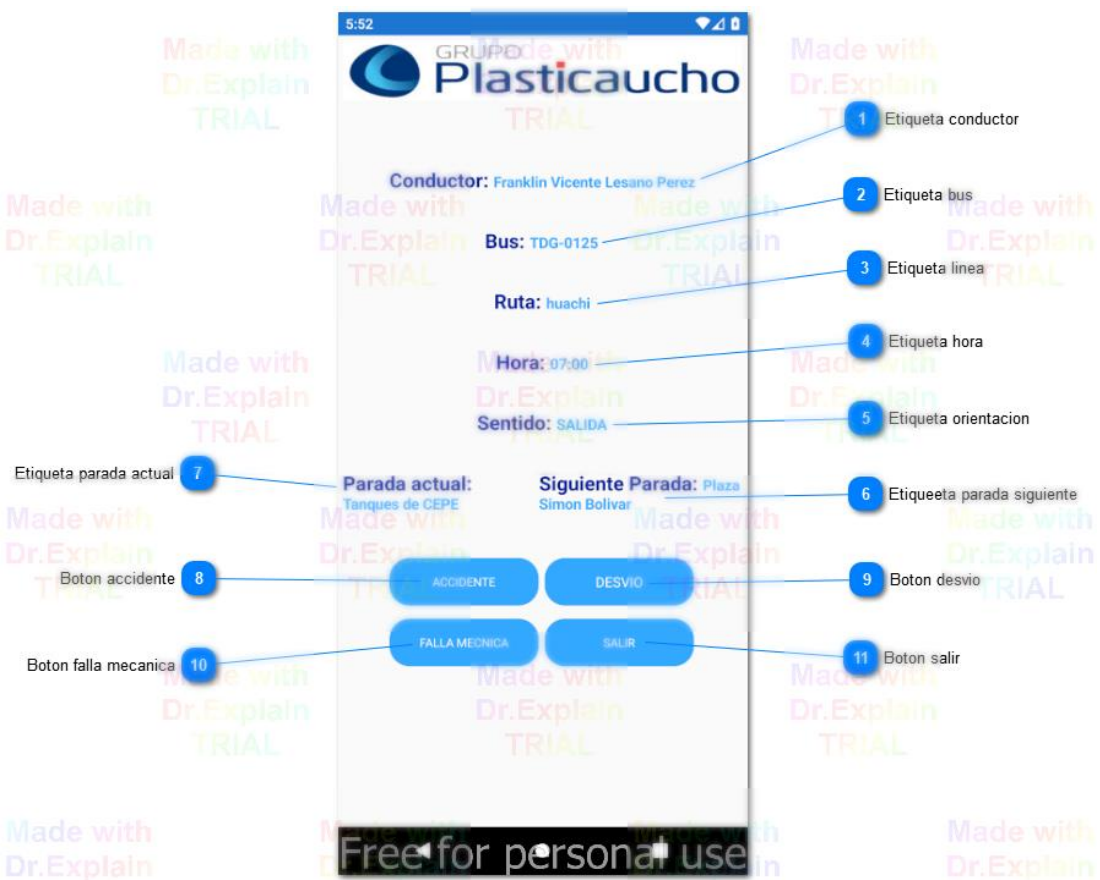
## 1.2.2. Vista datos ingresados



## Pantalla recorrida

La pantalla de visualización de la ruta para el recorrido es informativa ya que presenta los datos ingresados anteriormente, existen funcionalidades para el envío de notificaciones en caso de algún incidente en el transcurso del recorrido.

### 1.3.1. Área de clientes



1

#### Etiqueta conductor

**Conductor:** Franklin Vicente Lesano Perez

Etiqueta informativa del conductor.

2

#### Etiqueta bus



**Bus:** TDG-0125

Etiqueta informativa del bus.

3

### Etiqueta línea

**Ruta:** huachi

Etiqueta informativa de la línea.

4

### Etiqueta hora

**Hora:** 07:00

Etiqueta informativa del horario.

5

### Etiqueta orientación

**Sentido:** SALIDA

Etiqueta informativa de la información.

6

### Etiqueta parada siguiente

**Siguiente Parada:** Plaza  
Simon Bolivar

Etiqueta informativa de la parada siguiente

7

### Etiqueta parada actual

**Parada actual:**  
Tanques de CEPE

Etiqueta informativa de la parada siguiente

8

### **Botón accidente**



Botón de envío de notificaciones a los pasajeros en caso de accidente.

9

### **Botón desvío**



Botón de envío de notificaciones a los pasajeros en caso de desvío de la parada siguiente.

10

### **Botón falla mecánica**



Botón de envío de notificaciones a los pasajeros en caso de falla mecánica.

11

### **Botón salir**

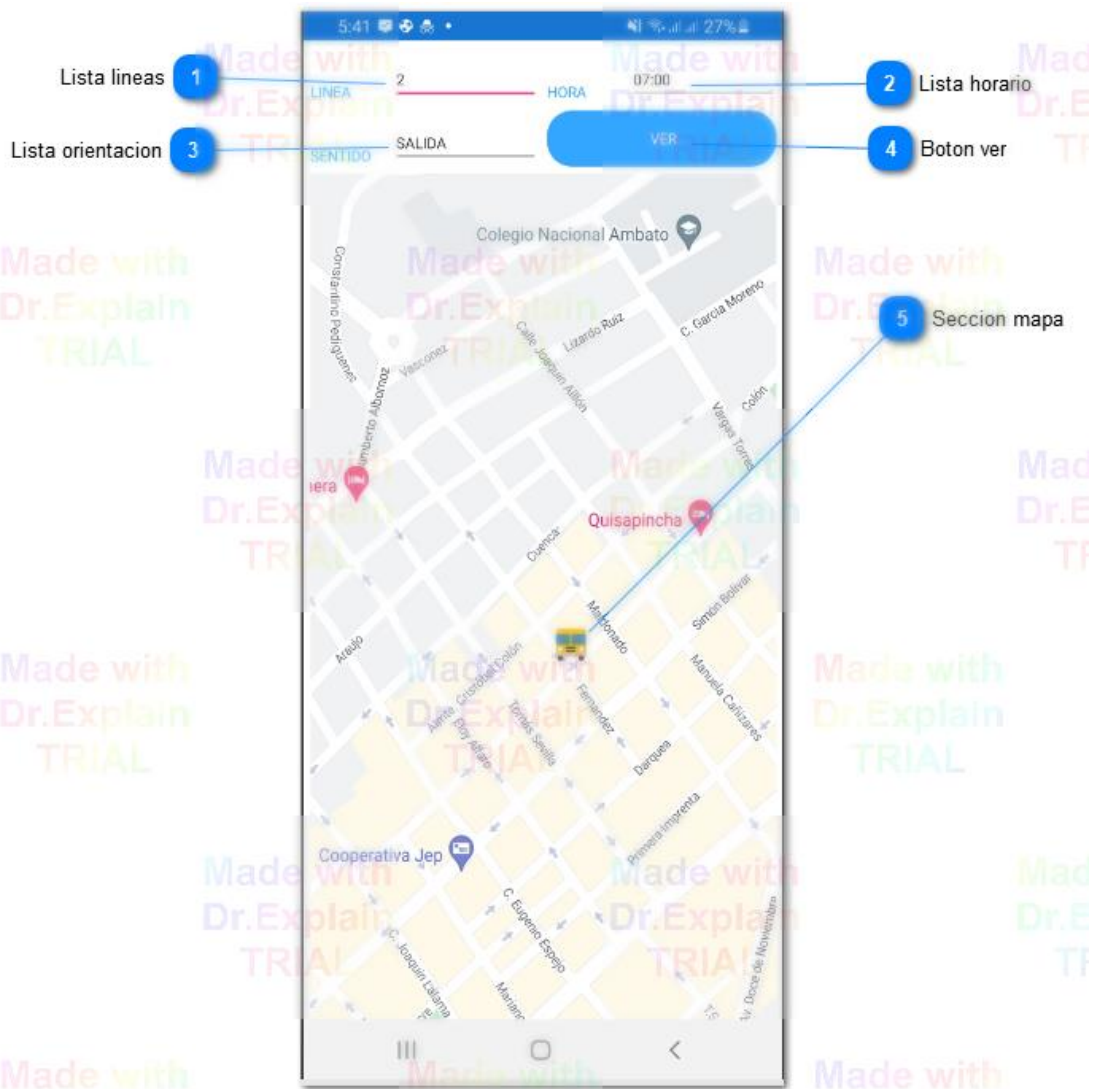


Botón para salir, terminar la ruta y salir del sistema

## 2 Aplicativo pasajeros

La pantalla de la ruta consta de cuatro campos necesarios, el primero se refiere a la placa del bus, el segundo campo se refiere al nombre de la línea, el tercero es el sentido de la trayectoria y por último el horario asignado.

### 2.1. Pantalla del pasajero



#### 1 Lista líneas

LINEA 2

Lista de rutas disponibles para mostrar, el pasajero debe seleccionar la ruta para ver en el mapa.

2

### Lista horaria

HORA 07:00

Lista de horarios para mostrar, el pasajero debe seleccionar el horario para ver en el mapa.

3

### Lista orientación

SENTIDO SALIDA

Lista de horarios para mostrar, el pasajero debe seleccionar el horario para ver en el mapa.

4

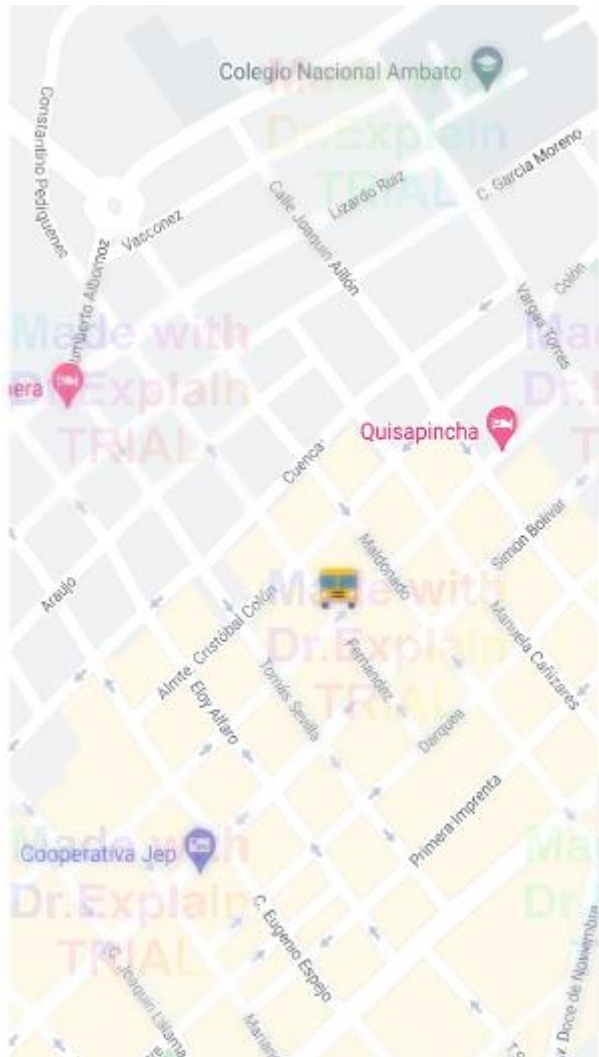
### Botón ver

VER

El botón ver, una vez ingresado la información previa, el pasajero debe presionar para empezar a trazar el trayecto de la línea deseada en tiempo real.

5

### Sección mapa

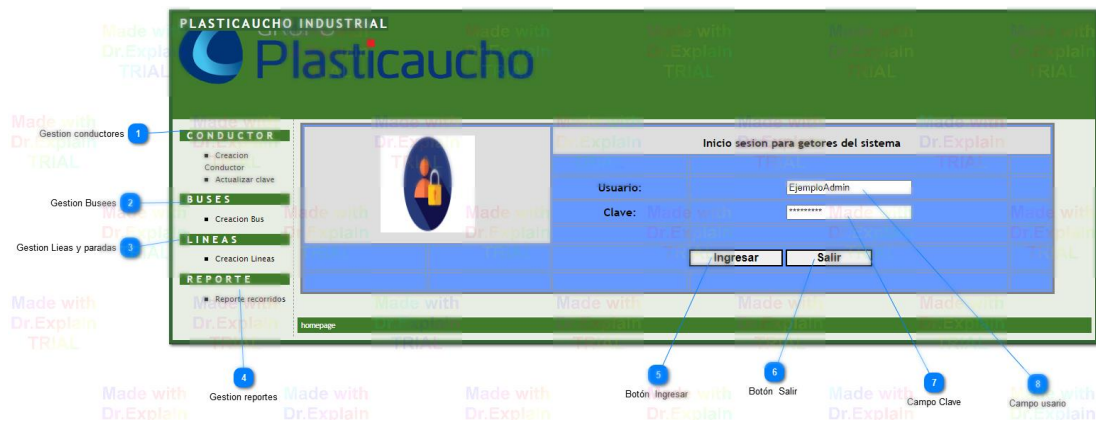


Sección para visualización del bus en el mapa

### 3. Gestor Web

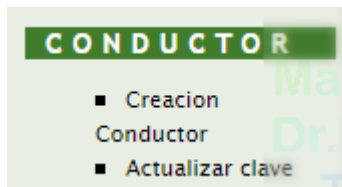
El sistema de gestión consta de pantallas para añadir, actualizar y consultar registros de conductores, buses, líneas y paradas, entre las funcionalidades están los reportes para visualizar los recorridos realizados en el mapa de Google maps.

#### 3.1. Pantalla de inicio



1

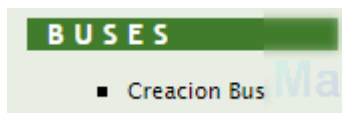
#### Gestión conductores



Menú para el ingreso a la gestión de conductores.

2

#### Gestión Buses



Menú para el ingreso a la gestión de buses.

3

#### Gestión líneas y paradas



Menú para el ingreso a la gestión de líneas y paradas

4

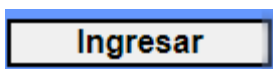
#### Gestión reportes



Menú para el ingreso a la gestión de reportes

5

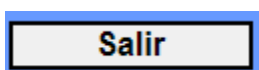
#### Botón Ingresar



Botón para el ingreso al sistema, una vez autenticado en el sistema se podrá utilizar el menú lateral.

6

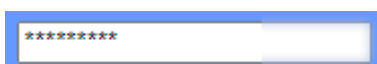
#### Botón Salir



Botón para salir del sistema una vez presionado el menú lateral quedara inactivo.

7

#### Campo Clave



Campo para el ingreso de la clave del gestor del sistema.

8

## Campo usuario

EjemploAdmin

Campo para el ingreso del usuario gestor del sistema.

### 3.2. Pantalla gestión conductor

1 Boton Nuevo

2 Boton edición

3 Boton eliminar

4 Tabla lista conductores

	Cedula	Nombre	Apellido	Telefono	Celular	Correo	Clave
<a href="#">Seleccionar</a>	1801321975	ejemplo 1	anonimo	00000000	0900000008	ejemplo1gmail.com	*****
<a href="#">Seleccionar</a>	1850285808	ejemplo2	anonimo	3030303030	0983150000	ejemplo2@gmail.com	*****
<a href="#">Seleccionar</a>	1804220265	Franklin Vicente	Lesano Perez	3030303030	0983065148	prueba@hotmail.com	*****

1

#### Botón Nuevo



Al presionar el botón nuevo se abrirá una vista para el ingreso de la información necesaria para el ingreso de un nuevo registro

2

#### Botón edición





Se debe previamente seleccionar un registro de la tabla para activar este botón caso contrario emitirá una alerta que no se ha seleccionado ningún registro

**3**

**Botón eliminar**



Antes de utilizar este botón se debe seleccionar el registro que se desea eliminar, antes de eliminar el registro mostrara una alerta de confirmación.

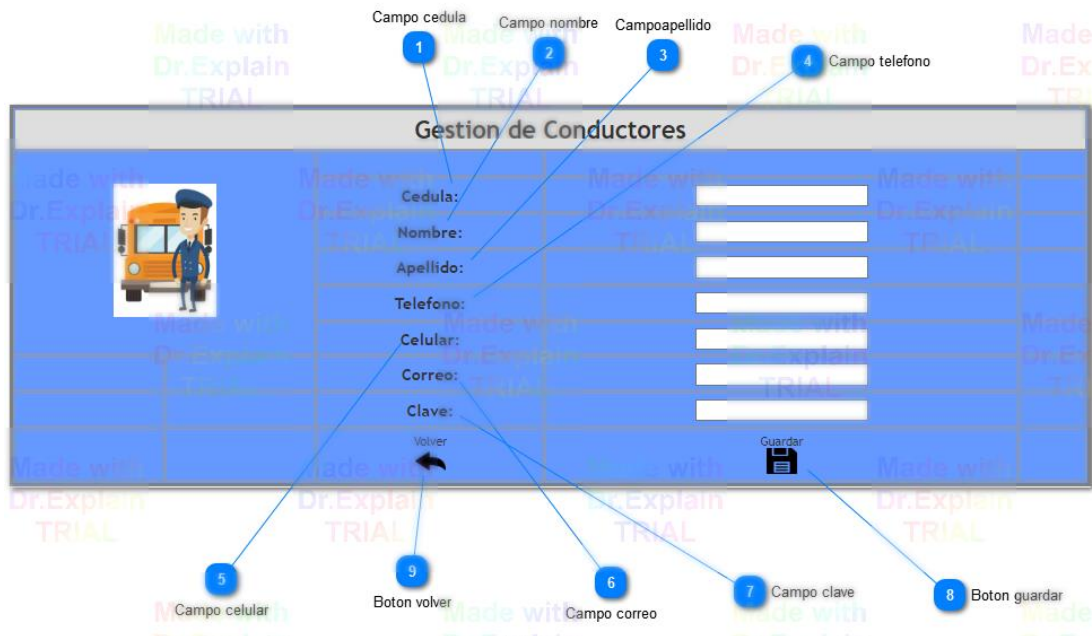
**4**

**Tabla lista conductores**

	Cedula	Nombre	Apellido	Telefono	Celular	Correo	Clave
<a href="#">Seleccionar</a>	1801321975	ejemplo 1	anonimo	00000000	0900000008	ejemplo1@gmail.com	*****


La tabla de conductores muestra un listado de los conductores activos.

**3.2.1. Pantalla ingreso información**



1

### Campo cedula



Cedula:

El campo cedula admite solo 10 caracteres numéricos para el registro del conductor.

2

### Campo nombre



Nombre:

El campo nombre admite una cadena para el registro del conductor.

3

### Campo apellido



Apellido:

El campo apellido admite una cadena para el registro del conductor.

4

### Campo teléfono



Telefono:

El campo teléfono admite solo 10 caracteres numéricos para el registro del conductor.

5

### Campo celular



Celular:

El campo celular admite solo 10 caracteres numéricos para el registro del conductor.

6

### Campo correo

Correo:

El campo correo admite una cadena para el registro del conductor.

7

### Campo clave

Clave:

El campo clave admite hasta 15 caracteres de tipo cadena para el registro del conductor

8

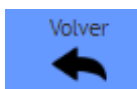
### Botón guardar



El botón guardar al presionar validará que todos los datos estén llenos luego de ello guardará un registro del conductor si algo ocurre emitirá una alerta informando.

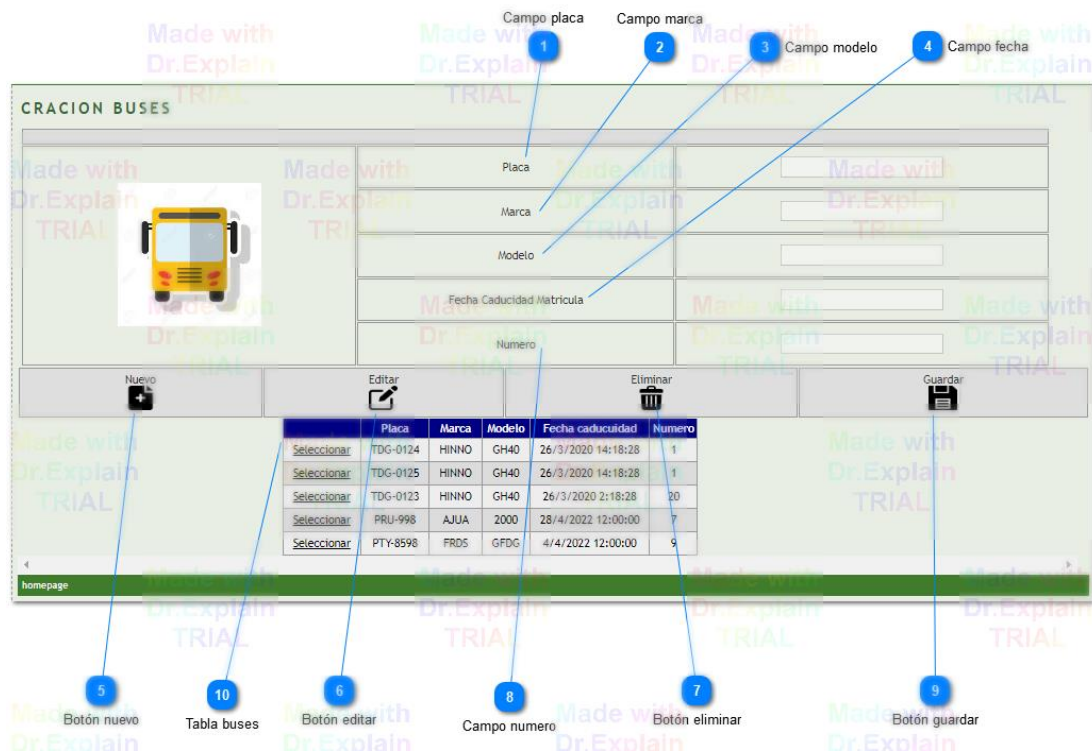
9

### Botón volver



El botón volver retornara a la pantalla principal del conductor.

### 3.3. Pantalla gestión buses



#### 1 Campo placa

Placa

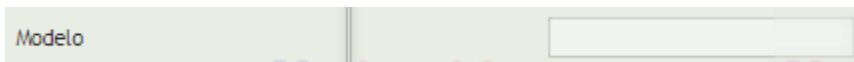
El campo placa admite una cadena para el registro o edición de un conductor.

#### 2 Campo marca

Marca

El campo marca admite una cadena para el registro o edición de un conductor.

#### 3 Campo modelo



El campo modelo admite una cadena para el registro o edición de un conductor.

4

#### **Campo fecha**



El campo fecha admite una cadena para el registro o edición de un conductor.

5

#### **Botón nuevo**



El botón nuevo permite que se habilite los campos para el ingreso de la información necesaria para añadir un nuevo registro tipo bus

6

#### **Botón editar**



El botón editar permite que se habilite los campos para la edición, previo a la edición se debe seleccionar un registro.

7

#### **Botón eliminar**



Previo a la eliminación se debe seleccionar un registro de la tabla buses, una vez que se presiona pedirá confirmación para eliminar el registro.

8

### Campo numero



A screenshot of a web form showing a label 'Numero' followed by a text input field.

El campo número admite datos numéricos para el registro o edición de un conductor.

9

### Botón guardar



Al presionar guardar dependiendo de la acción actualizara o añadirá un registro tipo bus, si algo ocurre retornara un mensaje de advertencia.

10

### Tabla buses

	Placa	Marca	Modelo	Fecha caducuidad	Numero
<a href="#">Seleccionar</a>	TDG-0124	HINNO	GH40	26/3/2020 14:18:28	1

La tabla buses muestra todos los buses activos en la base de datos.

### 3.4. Pantalla gestión líneas y paradas

The screenshot shows a web application interface for creating and managing bus lines and stops. The interface is titled 'CREACION DE LINEAS' and is divided into several sections:

- Line Creation Form:** Includes fields for 'Nombre Línea' (Santa Rosa), 'Sentido' (SALIDA), and 'Numero línea' (1). It has buttons for 'Nuevo', 'Editar', 'Eliminar', and 'Guardar'.
- Stop Creation Form:** Includes fields for 'Nombre Parada', 'Latitud' (0.0000), 'Longitud' (0.0000), and 'Orden'. It also has buttons for 'Nuevo', 'Editar', 'Eliminar', and 'Guardar'.
- Line Table:** A table with columns: Id Línea, Nombre, Sentido, Numero. It lists existing lines like 'Santa Rosa', 'huachi', 'PASO LATERAL', 'FICOA', 'EJEMPLO2', and 'EJEMPLO11'.
- Stop Table:** A table with columns: Nombre, Latitud, Longitud, Orden, Id Línea. It lists stops like 'PR1', 'EE', and 'PR2'.

Numbered callouts (1-17) identify the following elements:

- 1: Botón nuevo
- 2: Botón editar
- 3: Campo nombre línea
- 4: Campo sentido
- 5: Campo numero
- 6: Botón nuevo
- 7: Campo nombre parada
- 8: Campo latitud
- 9: Campo longitud
- 10: Campo orden
- 11: Botón eliminar
- 12: Botón guardar
- 13: Botón editar
- 14: Botón eliminar
- 15: Botón guardar
- 16: Tabla paradas
- 17: Tabla líneas

1

#### Botón nuevo



El botón nuevo habilita los campos necesarios para agregar la información para el nuevo registro tipo línea.

2

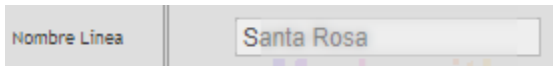
#### Botón editar



El botón editar habilita los campos necesarios para agregar la información para el nuevo registro tipo línea, previo a la edición se debe seleccionar el registro a editar.

3

#### Campo nombre línea

A screenshot of a form field labeled 'Nombre Línea'. The field contains the text 'Santa Rosa' and has a small dropdown arrow on the right side.

El campo nombre línea admite una cadena de texto necesarios para añadir o actualizar en registro tipo líneas.

4

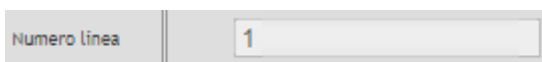
#### **Campo sentido**

A screenshot of a form field labeled 'Sentido'. The field contains a dropdown menu with the text 'SALIDA' and a small downward arrow.

El campo sentido despliega una lista de orientación necesarios para añadir o actualizar en registro tipo líneas.

5

#### **Campo numero**

A screenshot of a form field labeled 'Numero línea'. The field contains the number '1'.

El campo número admite datos tipos numéricos necesarios para añadir o actualizar en registro tipo líneas.

6

#### **Botón nuevo**



El botón nuevo habilita los campos necesarios para agregar la información para el nuevo registro tipo paradas, además de ello se desplegará un mapa para elegir la latitud y longitud de la parada.

7

#### **Campo nombre parada**

A screenshot of a form field labeled 'Nombre Parada'. The field is empty and has a small dropdown arrow on the right side.



El campo nombre parada admite una cadena de texto necesarios para añadir o actualizar en registro tipo parada.

8

### Campo latitud

Latitud	<input type="text" value="0.0000"/>
---------	-------------------------------------

El campo latitud admite valores de tipo decimal necesarios para añadir o actualizar en registro tipo parada, este dato se puede editar manualmente o ser seleccionado del mapa que se despliega.

9

### Campo longitud

Longitud	<input type="text" value="0.0000"/>
----------	-------------------------------------

El campo longitud admite valores de tipo decimal necesarios para añadir o actualizar en registro tipo parada, este dato se puede editar manualmente o ser seleccionado del mapa que se despliega.

10

### Campo orden

Orden	<input type="text"/>
-------	----------------------

El campo orden admite valores de tipo números necesarios para añadir o actualizar en registro tipo líneas.

11

### Botón eliminar



Previo a eliminar se debe seleccionar el registro de la tabla paradas, una vez seleccionado se presionará el botón donde pedirá

conformación para la acción, si algo ocurre emitirá un mensaje de advertencia.

12

### Botón guardar



El botón guardar dependiendo de la acción previa actualizará o añadirá un registro de tipo líneas, si algo ocurre emitirá una advertencia.

13

### Botón editar



El botón editar habilita los campos necesarios para agregar la información para el nuevo registro tipo línea, previo a la edición se debe seleccionar el registro a editar.

14

### Botón eliminar



Previo a eliminar se debe seleccionar el registro de la tabla líneas, una vez seleccionado se presionará el botón donde pedirá conformación para la acción, si algo ocurre emitirá un mensaje de advertencia.

15

### Botón guardar



El botón guardar dependiendo de la acción previa actualizará o añadirá un registro de tipo paradas, si algo ocurre emitirá una advertencia.

16

### Tabla paradas

	Nombre	Latitud	Longitud	Orden	Id Linea
<a href="#">Seleccionar</a>	PR1	-1.19763962060824	-78.5997585702183	1	11

La tabla parada visualiza los registros de las paradas

17

### Tabla líneas

	Id Linea	Nombre	Sentido	Numero
<a href="#">Seleccionar</a>	1	Santa Rosa	SALIDA	1

La tabla líneas visualiza los registros de las paradas.

### 3.3.1. Pantalla coordenadas geográficas

Mapa Google

Mapa Satélite

CC LABORATORIOS PHARMAVITAL CIA...

Via a Samanga Granja Don Bosco

Via a Fe

Parque Industrial Ambato

Lavandería Mega Color

GADMA Camal Municipal de Ambato

CURTIDURIA TUNGURAHUA

RECTIMA INDUSTRY CIA LTDA

MOLINOS SAN JOSE

EUROMAQUINAS

Capilla Católica

Iglesia Católica de San José del Pisque

Maquiladora de calzado Zoe

Electroindustrial Romero

INARECROM S.A

Bazar y Papelería Anahi

Plasticaucho industrial SA Pia

Inphec Agroindustrial

Comercial Primavera

Iglesia Santa Fe

FERRETERIA SANTA FE

Viveres Lisette

AVIPAZ

LA HUECA DEL MORENO

DISTRIBUIDORA PLASTI-MARC

Google

Latitud: -1.1980113233255323

Longitud: -78.59409955638857

Combinaciones de teclas Datos del mapa ©2022 Condiciones del Servicio Informar un error en el mapa

1 Botón volver

2 Campo latitud

3 Campo longitud

1

### Botón volver



El botón volver retornará a la página de gestión líneas y paradas, el botón no se activará si no se acoge un punto.

2

### Campo latitud

Latitud:

El campo latitud se llenar automáticamente cuando el gestor seleccione un punto en el mapa de Google.

3

### Campo longitud

Longitud:

El campo longitud se llenar automáticamente cuando el gestor seleccione un punto en el mapa de Google.

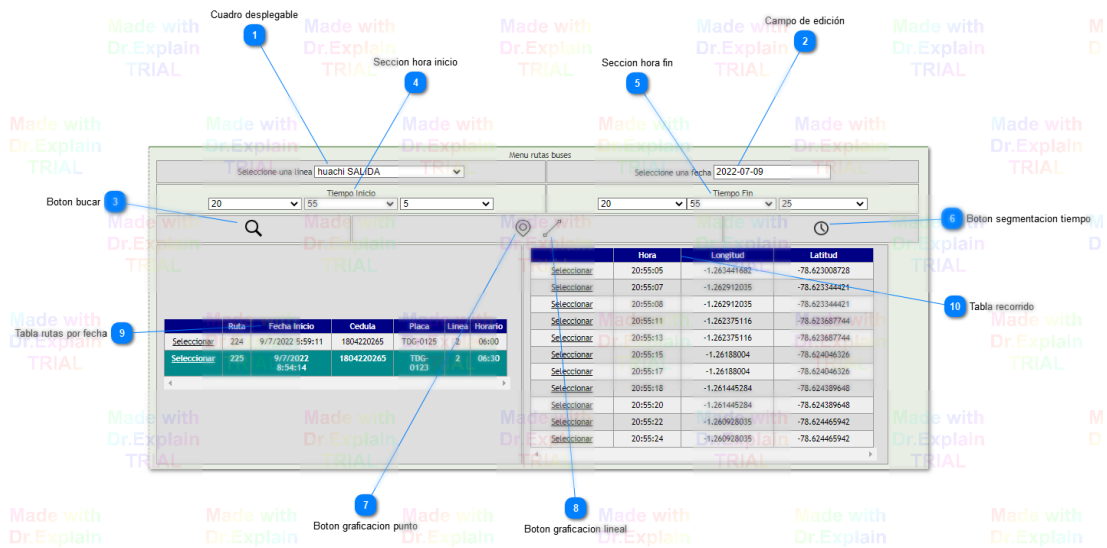
4

### Mapa Google



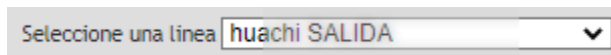
El Mapa está configurado para escuchar el evento de dar clic, al dar clic sobre un punto del mapa este llenara los campos latitud y longitud.

### 3.5. Pantalla reporteria



1

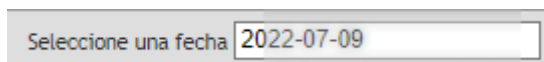
#### Cuadro desplegable



Listado de líneas activas, el gestor debe seleccionar la línea para validar si existe en el listado de rutas por fecha.

2

#### Campo de edición



El gestor debe elegir del calendario la fecha para traer el listado de rutas por la fecha buscada.

3

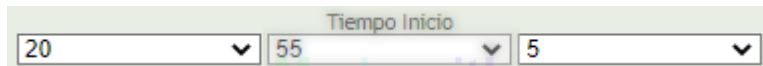
#### Botón buscar



Previo a buscar se debe elegir una fecha y una línea, con estos datos se llenará la tabla de rutas, si no existe datos no mostrará ningún listado.

4

#### Sección hora inicio



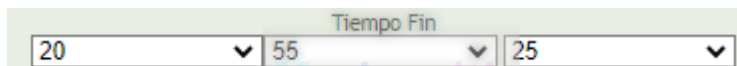
Tiempo Inicio

20	55	5
----	----	---

El gestor debe construir la hora completa, debe desplegar la hora y seleccionar, desplegar los minutos y seleccionar, debe desplegar y seleccionar, tomar en cuenta que la construcción es formato de 24 horas.

5

#### Sección hora fin



Tiempo Fin

20	55	25
----	----	----

El gestor debe construir la hora completa, debe desplegar la hora y seleccionar, desplegar los minutos y seleccionar, debe desplegar y seleccionar, tomar en cuenta que la construcción es formato de 24 horas.

6

#### Botón segmentación tiempo



botón de segmentación, previo a presionar el gestor debe construir la hora de inicio y la hora de fin, si se construye mal el sistema enviara un mensaje de alerta, si no existe datos mostrara un mensaje, caso contrario segmentara el listado del recorrido.

7

#### Botón graficación punto



Al presionar el botón se desplegará una nueva pestaña con los puntos en un mapa de Google maps disponibles, esta graficación depende del listado de la tabla recorrido.

8

### Botón graficación lineal



Al presionar el botón se desplegará una nueva pestaña graficando tipo líneas en un mapa de Google maps, esta graficación depende del listado de la tabla recorrido.

9

### Tabla rutas por fecha

	Ruta	Fecha Inicio	Cedula	Placa	Línea	Horario
<a href="#">Seleccionar</a>	224	9/7/2022 5:59:11	1804220265	TDG-0125	2	06:00

La tabla de rutas desplegará el listado de todas las rutas existentes en la fecha seleccionada.

10

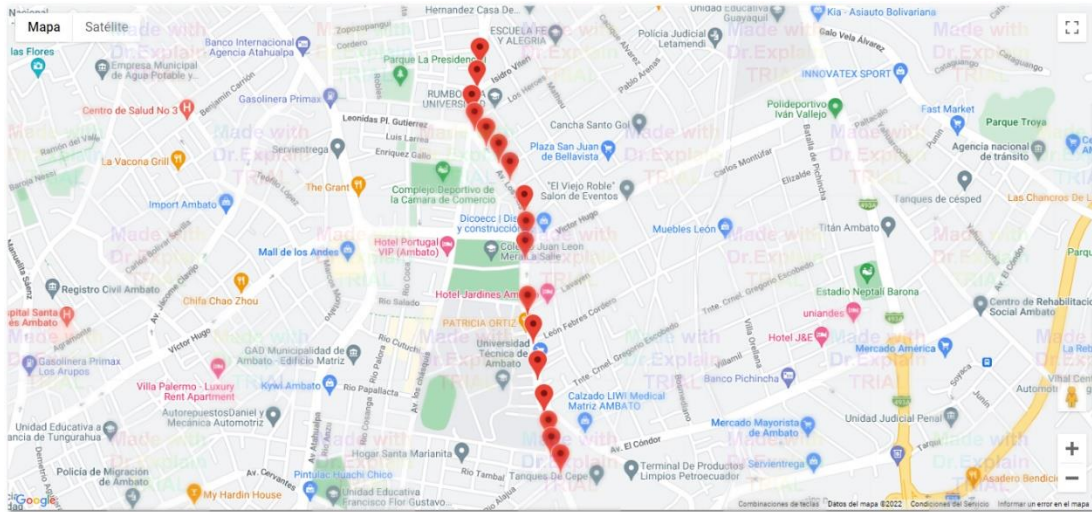
### Tabla recorrido

	Hora	Longitud	Latitud
<a href="#">Seleccionar</a>	20:55:05	-1.263441682	-78.623008728

Esta tabla está disponible después de que el gestor selecciona una ruta, desplegará un listado del recorrido con intervalos de 2 segundos.



### 3.5.1. Graficación por puntos



### 3.5.2. Graficación por líneas

