

**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



Presentado por

**ZULMA ISABEL ALVAREZ CARTOLIN**

**DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA  
DE GEOLOCALIZACIÓN EN TIEMPO REAL DE LA EMPRESA DE  
TRANSPORTE PÚBLICO "CRISTO MORADO"-ANDAHUAYLAS.**

**Asesor:**

**DR. JULIO CÉSAR HUANCA MARÍN**

**Co – Asesor:**

**MTRO. JUAN JOSÉ ORE CERRÓN**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE  
SISTEMAS**

**ANDAHUAYLAS – APURÍMAC – PERÚ**

2020

## **DEDICATORIA**

Dedico el presente trabajo de titulación a Dios quien con su bendición llena completamente mi vida, por proveerme fuerza y permitirme seguir adelante.

A mis padres Guido e Isabel, quienes son mi mayor inspiración, por los consejos, valores y principios que me han inculcado, gracias a su apoyo diario me permitieron alcanzar esta meta en mi formación académica.

También se lo dedico a mis hermanos Yovana, Ronal, Anthony, Pepe, Guido y Carolina Alvarez, por el apoyo mutuo que siempre nos hemos brindado hasta en los peores momentos cuando todo se complica. Por último, a cada uno de mis amigos que siempre me apoyaron directa o indirectamente durante todos estos años de formación profesional.

## **AGRADECIMIENTO**

Agradecer a cada una de las personas que me apoyaron oportunamente para culminar con éxito, y en especial a aquellas que me abrieron las puertas y compartieron sus conocimientos a lo largo de mi formación académica. Además, a mi asesor, el Dr. Julio César Huanca Marín quien fue mi guía, brindándome conocimientos que permitieron el desarrollo del presente trabajo de titulación.

# ÍNDICE

ÍNDICE .....	4
PLANTEAMIENTO DEL PROBLEMA .....	15
1.1. Realidad problemática.....	15
1.2. Formulación del problema.....	16
1.3. Objetivos.....	16
1.3.1. Objetivo general .....	16
1.3.2. Objetivos específicos .....	16
1.4. Justificación .....	17
1.5. Viabilidad de la Investigación .....	17
1.5.1. Viabilidad técnica.....	17
1.6. Limitaciones del estudio .....	17
MARCO TEÓRICO.....	18
2.1. ANTECEDENTES INTERNACIONALES .....	18
2.2. ANTECEDENTES NACIONALES.....	18
2.3. ANTECEDENTES LOCALES .....	19
2.4. BASES TEÓRICA.....	20
2.4.1. Servicios.....	20
2.4.2. Transporte.....	20
2.4.3. Sistema de transporte .....	21
2.4.4. Transporte Público Urbano.....	21
2.4.5. Clasificación del Transporte Urbano de Pasajeros.....	22
2.4.6. Dispositivo Móvil.....	22
2.4.7. Smartphone.....	22
2.4.8. Aplicativos Móviles .....	23
2.4.9. Aplicaciones Híbridas.....	25
2.4.10. Aplicaciones Nativas .....	25
2.4.11. Sistemas operativos.....	26
2.4.12. Sistema operativo móviles.....	26
2.4.13. Tipos de sistemas móviles.....	26
2.4.14. Internet .....	27
2.4.15. Geolocalización .....	27
2.4.16. Base de Datos No SQL.....	27
2.4.17. Firebase.....	28
2.4.18. Tiempo Real .....	29

2.4.19. Framework.....	29
2.4.20. IONIC.....	30
2.4.21. Cordova.....	31
2.4.22. Angular.....	32
2.4.23. Nod js.....	32
2.4.24. PMBOK.....	33
2.4.25. PMI.....	33
2.4.26. Metodología Mobile-D.....	33
<b>PROPUESTA DE SOLUCIÓN.....</b>	<b>34</b>
3.....	34
3.1. Ingeniería de proyecto.....	34
3.2. Bases de datos de la aplicación.....	34
3.3. Diseño modular.....	59
3.4. Administración y gestión del proyecto basado en PMBOK.....	82
3.4.1. Plan de gestión de la integración.....	83
3.4.2. Plan de Gestión de alcance.....	87
3.4.3. Plan de gestión del cronograma.....	88
3.4.4. Plan de gestión de los costos.....	90
3.4.5. Plan de Gestión de la calidad.....	91
3.4.6. Plan de Gestión de los recursos.....	91
3.4.7. Plan Gestión de las Comunicaciones.....	92
3.4.8. Plan de Gestión de riesgo.....	93
3.4.9. Plan de gestión de las adquisiciones.....	93
3.4.10. Plan de gestión de los interesados.....	94
3.5. Metodología de desarrollo.....	95
3.5.1. Fase de Exploración.....	95
3.5.2. Fase de iniciación.....	98
3.5.3. Fase de Producción.....	116
3.5.4. Fase de Estabilización.....	129
3.5.5. Fase pruebas.....	180
<b>CONCLUSIONES.....</b>	<b>181</b>
<b>RECOMENDACIONES.....</b>	<b>182</b>
<b>Bibliografía.....</b>	<b>183</b>
<b>ANEXOS.....</b>	<b>187</b>

## Lista de tablas

Tabla 1. Ventajas de Aplicación Híbrida. Ventajas de Aplicación Nativa.....	24
Tabla 2: Plan de gestion de la integración .....	83
Tabla 3: Plan de gestión de alcance .....	88
Tabla 4: plan de gestión de cronograma .....	89
Tabla 5: Plan de gestión de costos .....	90
Tabla 6: plan de gestión de la calidad .....	91
Tabla 7: Plan de gestión de los recursos .....	91
Tabla 8: Plan Gestión de las Comunicaciones .....	92
Tabla 9: Plan de gestión de riesgo .....	93
Tabla 10: plan de gestión de las adquisiciones .....	93
Tabla 11: Plan de gestión de los interesados .....	94
Tabla 12: Información clave del proyecto .....	97
Tabla 13: Caso de Uso iniciar Sesión. ....	104
Tabla 14: Caso de uso geolocalización en tiempo real. ....	105
Tabla 15: Caso de uso perfil. ....	105
Tabla 16: Caso de uso registro administrador.....	106
Tabla 17: Caso de uso registro conductor.....	106
Tabla 18: Caso de uso lista de conductores. ....	107
Tabla 19: Caso de uso borrar conductor. ....	108
Tabla 20: Caso de uso actualizar conductor .....	108
Tabla 21: Caso de uso reporte salida y llegada.....	109
Tabla 22: Caso de uso cerrar sesión administrador. ....	109
Tabla 23: Caso de uso iniciar sesión con clave.....	109
Tabla 24: Caso de uso geolocalización del vehículo actual.....	110
Tabla 25: Caso de uso geolocalización de todos los vehículos .....	111
Tabla 26: Caso de uso orden de circulación de los vehículos mediante una alerta. ....	111
Tabla 27: Caso de uso control de salida y llegada .....	112
Tabla 28: Caso de uso cerrar sesión conductor.....	112
Tabla 29: Caso de uso iniciar sesión usuario. ....	113
Tabla 30: Caso de uso de registro usuario. ....	113
Tabla 31: Caso de uso recuperar contraseña. ....	114
Tabla 32: Caso de uso verificar usuario.....	114

Tabla 33: Caso de uso geocalización de vehículos.....	115
Tabla 34: Caso de uso perfil usuario.....	115
Tabla 35: Caso de uso Cerrar sesión usuario. ....	116
Tabla 36: Diseño de la aplicación móvil.....	117
Tabla 37: Concepto de la estructura de la aplicación .....	122
Tabla 38: Concepto del directorio de la aplicaación.....	124
Tabla 39 : Concepto del directorio de administrador de la aplicación. ....	125
Tabla 40: Concepto del directorio de conductor de la aplicación.....	126
Tabla 41. Concepto de directorio usuario de la aplicación .....	127
Tabla 42: Concepto de directorio assets de la aplicación .....	128
Tabla 43: Directorio de Servicios de la aplicación .....	129
Tabla 44: Directorio enviroments de la aplicación .....	129

## Lista de figuras

Figura 1. Funcionamiento de ionic. ....	30
Figura 2. Imagen de Apache Cordova. ....	31
Figura 3: Base de datos del módulo administrador.....	35
Figura 4: Base de datos módulo conductor.....	35
Figura 5: Base de datos módulo conductor.....	36
Figura 6: Base de datos de reporte de salida y llegada del vehículo en servicio .....	36
Figura 7: Diseño del módulo administrador iniciar sesión .....	37
Figura 8: Diseño del módulo administrador menú principal .....	38
Figura 9:Diseño del módulo administrador de geolocalización de los vehículos en servicio	39
Figura 10: Diseño del módulo administrador Perfil .....	40
Figura 11: Diseño del módulo administrador de registro .....	41
Figura 12: Diseño del módulo administrador de registro conductor .....	42
Figura 13: Diseño del módulo administrador de registro conductor .....	43
Figura 14: Diseño del módulo administrador lista de conductores.....	44
Figura 15:Diseño del módulo administrador reporte de salida y llegada .....	45
Figura 16: Diseño del módulo administrador actualizar conductor.....	46
Figura 17: Diseño del módulo administrador eliminar Conductor .....	47
Figura 18: Diseño del módulo conductor iniciar sesión con clave .....	48
Figura 19: Diseño del módulo conductor de ubicación actual.....	49
Figura 20:Diseño del módulo conductor geolocalización de vehículos .....	50
Figura 21:Diseño del módulo conductor orden de circulación de los vehículos mediante una alerta.....	51
Figura 22: Diseño del módulo conductor control registro de salida y llegada .....	52
Figura 23: Diseño del módulo conductor control registro de salida y llegada .....	53
Figura 24: Diseño del módulo usuario iniciar sesión.....	54
Figura 25: Diseño del módulo usuario registro.....	55
Figura 26: Diseño del módulo usuario verificar correo .....	56
Figura 27: Diseño del módulo usuario recuperar contraseña.....	57
Figura 28: Diseño del módulo usuario geolocalización de vehículos.....	58
Figura 29: Diseño del módulo usuario perfil .....	59
Figura 30: Diseño del módulo administrador iniciar sesión versión Ios .....	60
Figura 31: Diseño del módulo administrador menú principal versión Ios.....	61



Figura 32: Diseño del módulo administrador de geolocalización de los vehículos en servicio versión Ios .....	62
Figura 33: Diseño del módulo administrador Perfil versión Ios .....	63
Figura 34: Diseño del módulo administrador de registro versión Ios .....	64
Figura 35: Diseño del módulo administrador de registro conductor versión Ios .....	65
Figura 36: Diseño del módulo administrador de registro conductor versión Ios .....	66
Figura 37: Diseño del módulo administrador lista de conductores versión Ios .....	67
Figura 38: Diseño del módulo administrador reporte de salida y llegada versión Ios .....	68
Figura 39: Diseño del módulo administrador actualizar conductor versión Ios .....	69
Figura 40: Diseño del módulo administrador eliminar Conductor versión Ios .....	70
Figura 41: Diseño del módulo conductor iniciar sesión con clave versión Ios .....	71
Figura 42: Diseño del módulo conductor de ubicación actual versión Ios .....	72
Figura 43: Diseño del módulo conductor geolocalización de vehículos versión Ios .....	73
Figura 44: Diseño del módulo conductor orden de circulación de los vehículos mediante una alerta Versión Ios .....	74
Figura 45: Diseño del módulo conductor control registro de salida y llegada versión Ios .....	75
Figura 46: Diseño del módulo conductor control registro de salida y llegada versión Ios .....	76
Figura 47: Diseño del módulo usuario iniciar sesión versión Ios .....	77
Figura 48: Diseño del módulo usuario registro versión Ios .....	78
Figura 49: Diseño del módulo usuario verificar correo versión Ios .....	79
Figura 50: Diseño del módulo usuario recuperar contraseña versión Ios .....	80
Figura 51: Diseño del módulo usuario geolocalización de vehículos versión Ios .....	81
Figura 52: Diseño del módulo usuario perfil versión Ios .....	82
Figura 53: las 10 áreas de conocimiento de PMBOK .....	83
Figura 54: Metodologías de mobile-D .....	95
Figura 55: Diagrama de casos de uso .....	99
Figura 56: Diagrama de actividad crear conductor .....	100
Figura 57: Diagrama de actividad visualizar vehículos .....	100
Figura 58: Diagrama de actividad mostrar reporte de salida y llegada de vehículos .....	101
Figura 59: Diagrama de actividad orden de los vehículos mediante alertas .....	102
Figura 60: Diagrama de actividad muestras vehículo .....	102
Figura 61: Diagrama de actividad reporte de salida y llega en los paraderos iniciales y finales .....	103

Figura 62: Diagrama de actividad mostrar vehículos .....	103
Figura 63: Diagrama de actividad mostrar vehículo.....	104
Figura 64: Dependencias de paquetes de la aplicación.....	119
Figura 65: Plugins de la aplicación.....	120
Figura 66: Configuración de la aplicación module.ts .....	121
Figura 67: Estructura de la aplicación .....	122
Figura 68: Directorio de la aplicación .....	123
Figura 69: Módulos del directorio de la aplicación .....	124
Figura 70: Directorio del administrador de la aplicación.. .....	125
Figura 71. Directorio de conductor de la aplicación.....	126
Figura 72: Directorio de conductor de la aplicación.....	127
Figura 73: Directorio de Assets de la aplicación .....	128
Figura 74: Directorio de Servicios de la aplicación.....	128
Figura 75: Directorio de enviroments de la aplicación.....	129
Figura 76: Integración del proyecto con la base de datos.....	130
Figura 77: Integración de los módulos de acceso a la aplicación .....	130
Figura 78: crear-conductor.page.html.....	136
Figura 79: crear-conductor.page.html.....	138
Figura 80: lista-conductor.page.html .....	139
Figura 81: lista-conductor.page.ts .....	140
Figura 82: login-admin.page.html.....	141
Figura 83: login-admin.page.ts .....	143
Figura 84: mapa-admin.page.html .....	144
Figura 85:mapa-admin.page.ts.....	146
Figura 86: perfil-admin.page.html .....	147
Figura 87: perfil-admin.page.ts.....	150
Figura 88: registro-admin.page.html.....	151
Figura 89: registro-admin.page.ts .....	152
Figura 90: login-conductor.page.html.....	153
Figura 91: login-conductor.page.ts .....	155
Figura 92: mapa-conductor.page.html .....	156
Figura 93: mapa-conductor.page.ts.....	161
Figura 94: perfil-conductor.page.html .....	161

Figura 95: perfil-conductor.page.ts.....	162
Figura 96: vehiculos-conductor.page.html .....	162
Figura 97: vehiculos-conductor.page.ts .....	164
Figura 98: login-usuario.page.html.....	165
Figura 99: login-usuario.page.ts .....	168
Figura 100: mapa-usuario.page.html .....	168
Figura 101: mapa-usuario.page. ts .....	170
Figura 102: perfil-usuario.page.html .....	171
Figura 103: perfil-usuario.page. ts .....	174
Figura 104: registro-usuario.page.html.....	176
Figura 105: registro-usuario.page. ts .....	179
Figura 106: vereficar-email.page.html.....	179
Figura 107: verificar-email.page.ts .....	180
Figura 108: Pagina de descarga de Node.Js.....	187
Figura 109: Pagina de descarga de Node.Js.....	188
Figura 110: Descarga de Visual Code .....	189
Figura 111: Instalación de visual code.....	189
Figura 112: Icono de Visual code. ....	189
Figura 113: terminal en Visual code. ....	190
Figura 114: instalación en el terminal de Cordova .....	190
Figura 115: instalación en el terminal de ionic .....	190
Figura 116: iniciamos un nuevo proyecto en ionic .....	191
Figura 117: inicio de sesión de la aplicación .....	191
Figura 118: menú y sus respectivos rutas. ....	192
Figura 119: Pagina actualizar conductor.....	193
Figura 120: Pagina reporte de salida y llegada de vehículos en os paraderos iniciales y finales. .....	193
Figura 121: página de intercambio de painas inicio sesión. ....	194
Figura 122: inicio sesión del conductor mediante una clave. ....	195
Figura 123: páginas de reporte de entrada de salida y llegada en los paraderos iniciales y finales de cada vehículos en servicio. ....	196
Figura 124: orden de vehículos con alerta. ....	197
Figura 125: manual del usuario.....	198

## RESUMEN

La circulación en los vehículos de transporte público urbano en la ciudad de Andahuaylas tienen problemas de desorden vehicular, inseguridad, en lo específico con sus vehículos de la empresa de transporte Cristo Morado al conducir incontroladamente, como también los accidentes de tránsito que en seguidas ocasiones suelen suceder atropellos a los peatones o choque entre ellos o ya sea con vehículo particular y tiene que ser mejorada mediante distintos medios que permitan reducir el problema social que existe en la provincia de Andahuaylas. El presente trabajo de investigación se tituló: **“Desarrollo de aplicación móvil multiplataforma de geocalización en tiempo real para la empresa de transporte público Cristo Morado en la provincia de Andahuaylas, 2020”**, tiene como finalidad la ubicación en tiempo real de todos los vehículos de la empresa de transporte público urbano Cristo Morado de la ciudad de Andahuaylas, la reorganización de manera eficiente de los vehículos utilizando alertas para no sobrepasar a otro vehículo o en caso contrario de que otro vehículo ya le esté rebasando en las rutas que transita los vehículos y un mejor control de entrada y salida de los vehículos en los paraderos, además la aplicación se podrá ejecutarse en teléfonos inteligentes Smartphone con sistemas operativos Android, Windows e iOS, y que sea de uso intuitivo y fácil de entender.

Para el presente proyecto se empleó la metodología Mobile – D que se adecua a las características necesarias para que la aplicación móvil tenga un óptimo funcionamiento, por otra parte, la metodología Mobile - D facilita el trabajo de desarrollo y el diseño de la aplicación. Además, se usará algunas tecnologías y herramientas para el desarrollo y diseño de la aplicación como es el uso del framework IONIC 5.0, así como los plugins de Cordova como la API de google maps y los servicios de Firebase para obtener y gestionar las coordenadas de los vehículos en tiempo real y así la aplicación sea confiable y segura Todo esto con el fin de obtener una aplicación confiable y segura, que sea capaz de solucionar los problemas actuales de un servicio de transporte público en la Provincia de Andahuaylas.

## ABSTRACT

The circulation in urban public transport vehicles in the city of Andahuaylas have problems of vehicle disorder, insecurity, specifically with their vehicles from the Cristo Morado transport company when driving uncontrollably, as well as traffic accidents that often tend to Pedestrians are abused or collided with each other or with a private vehicle, and it has to be improved through different means to reduce the social problem that exists in the Andahuaylas province. The present research work was titled: "**Development of a multiplatform mobile application of geolocation in real time for the public transport company Cristo Morado in the province of Andahuaylas, 2020**", its purpose is to locate in real time all the vehicles of the Cristo Morado urban public transport company in the city of Andahuaylas, the efficient reorganization of vehicles using alerts so as not to overtake another vehicle or in the opposite case that another vehicle is already passing it on the routes that the vehicles travel and a better control of entry and exit of vehicles at bus stops, in addition the application can be run on Smartphone smartphones with Android, Windows and iOS operating systems, and that it is intuitive and easy to understand.

For this project, the Mobile - D methodology was used, which adapts to the characteristics necessary for the mobile application to function optimally, on the other hand, the Mobile - D methodology facilitates the development work and the design of the application. In addition, some technologies and tools will be used for the development and design of the application, such as the use of the IONIC 5.0 framework, as well as Cordova plugins such as the google maps API and Firebase services to obtain and manage the coordinates of the vehicles in real time and thus the application is reliable and safe. All this in order to obtain a reliable and safe application that is capable of solving the current problems of a public transport service in the Province of Andahuaylas.

## CHUMASQA

Tukuy clasi runa apaq carrukuna Andahuaylas Ilaqtapi sasachakun chaqrusqa chaynallataqmi manchachikunapaq purinku Empresa de Transporte Cristo Morado nisqampa carrunkuna, mana allinta puririchispa Andahuaylas Ilaqtapi, chaymi runakuna anchayta ankallirinku kikinku pura chanallataq runakunawampas. Chaymi llankarukun kay proyecto de investigación nisqam sutiyuq **“Desarrollo de aplicación móvil multiplataforma de geolocalización en tiempo real para la empresa de transporte público Cristo Morado en la provincia de Andahuaylas, 2020”**. Kay proyectowanmi empresa de transporte Cristo Morado tarikunqa munasqan ratupi, chaynallataqmi chaqrusqa carrukunata sumaqta puririchinqa alertas nisqanwan sutiyuq mana carrukuna ñaupachkanampaq ñannimpi puritinku, chaymi aplicación nisqanta churapakamunqa teléfonos inteligentes (smarphone) nisqampi, Andahuaylas Ilaqta runakunapaq allinnimpi.

Qinaspapas churapakamun metodología Mobile-D nisqan sutiyuq mana chaqrusqa rurakunampaq proyecto de investigación nisqampi, metodología Mobile-D llankan sumaqta aplicación puririnampaq, chaynallataqmi churapakamun tecnologiakuna sutiyukna: framework IONIC 5.0, Plugins de Cordova, APIS de Google Maps, Servicio Firebase y Coordenadas. Chaymi aplicación móvil nisqan sumaqta puririnqa Andahuaylas Ilaqtapi manaña sasachakuy, chaqrusqa kanampaq Empresa de Transporte Cristo Morado nisqampa.

## **PLANTEAMIENTO DEL PROBLEMA**

### **1.1. Realidad problemática**

En estos últimos años el problema de transporte urbano (público) es de preocupación de primer orden que más afecta en muchas regiones del Perú, según la última encuesta del observatorio ciudadano, algunos de ellos generan problemas de desorden vehicular, inseguridad, contaminación ambiental y sonora con sus vehículos poco cuidados afectando a los ciudadanos que hacen uso del transporte urbano. (RPP, 2018).

En Andahuaylas existen muchos problemas en cuanto se refiere al transporte público urbano local, como son las combis de las diferentes empresas que prestan este servicio, los conductores de los vehículos del transporte público siempre están infringiendo cualquier regla de tránsito, cometiendo atropellos al recoger los pasajeros o al no dejarlos en su destino, específicamente por apurados, también existen los desvíos imprevistos por otras rutas ya sea por algunas vías donde hay eventos o sierra de vías por algún motivo, ocasionando pérdida de tiempo en los pasajeros a bordo, se observa también el típico problema de los vehículos cuando los choferes conducen a altas velocidades por querer rebasar al combi de adelante o conducen con una paciencia insoportable porque tienen una gran distancia con el combi de adelante y atrás, también la espera de los pasajeros en los paraderos por largos ratos ocasionando pérdida de tiempo. Existe el incumplimiento de las normas y reglas que rige las empresas de transporte público urbano, como también la sobrecarga de pasajeros en un mismo vehículo ocasionando robos, acosos, contagios de enfermedades; el problema es tan grande que los choferes de los vehículos del transporte público están en constante competencia de ganarse un pasajero, tanto así que se pasan el semáforo en rojo a altas velocidades, en otros casos desviándose de las vías de carretera y a si sobresaliendo a las veredas en plena vía de transeúntes, exponiendo a los peatones a sufrir algún tipo de accidente de tránsito, ya que todo esto afecta a la salud mental de las personas ocasionando situaciones que elevan los niveles de ansiedad, estrés, etc. y en estas situaciones es cuando se demuestra la ausencia de las autoridades, la informalidad, el poco respeto por las leyes de tránsito y hacia las personas. Los choferes de los vehículos del transporte público no cuentan con un sistema de control para la ubicación de los vehículos

siendo éstos difíciles de gestionar a la hora de sus reportes en la llegada y salida de los paraderos finales.

No existe una organización en la circulación de la vía urbana con la empresa de transporte público Cristo Morado de la Provincia de Andahuaylas en las rutas de las ciudades de Talavera, Andahuaylas y San Jerónimo. Todos los días se pierde una o más horas en el transporte y solo llegan hasta determinados paraderos sin cumplir los horarios determinados y donde los pasajeros tienen que caminar largas rutas para llegar a sus puntos de destino, lo cual origina valiosa pérdida de tiempo, la inseguridad por las noches, gastos adicionales para llegar a sus destinos y movimientos económicos que atentan contra la economía familiar.

Hoy en día al esperar en los paraderos y lugares de ruta a los vehículos de transporte público existe aglomeración de personas lo cual genera un inminente punto de contagios y son más vulnerables a contraer el COVID-19 (CORONAVIRUS).

## **1.2. Formulación del problema**

Los vehículos en servicio de la empresa de transporte público urbano Cristo Morado de Andahuaylas tienen problemas en el orden de circulación por sus respectivas rutas ya que no cuentan con un sistema de circulación que ordene a los vehículos según su salida y llegada a los paraderos iniciales y finales en cada ruta designada, carencia de un óptimo control de los reportes de entrada y salida, y así mismo existe la aglomeración de personas en el tiempo que esperan a los vehículo en servicio en los diferentes paraderos de la ciudad.

## **1.3. 7 Objetivos**

### **1.3.1. Objetivo general**

Desarrollar una aplicación móvil multiplataforma de geolocalización en tiempo real de los vehículos de la empresa de transporte “Cristo Morado” en la ciudad de Andahuaylas.

### **1.3.2. Objetivos específicos**

- Crear el módulo de ubicación en tiempo real de los vehículos de la empresa de transporte público Cristo Morado.
- Implementar el módulo de reorganización enviando alertas a los vehículos de la empresa de transporte público Cristo Morado.



#### **1.4. Justificación**

En la actualidad los usuarios que hacen uso del transporte público en la ciudad de Andahuaylas están expuestos a la propagación del covid-19 donde que existe aglomeración de personas al esperar en los paraderos el transporte público urbano, por tal motivo se está desarrollando un aplicativo móvil de geocalización en tiempo real del transporte público urbano donde el aplicativo permitirá visualizar al usuario el lugar de la ruta donde se encuentra en tiempo real y así los usuarios optimizarán su tiempo y evitaran la propagación del COVIT-19 , como también el aplicativo móvil reorganizará el desorden de la circulación de los vehículos del transporte público Cristo Morado en las rutas que transitan de la ciudad de Talavera, Andahuaylas, San jerónimo.

Tiene como finalidad la ubicación en tiempo real de todos los vehículos, la eficiente gestión de entrada y salida de los vehículos en los paraderos, además la aplicación se podrá ejecutarse en teléfonos inteligentes Smartphone con sistemas operativos Android, Windows e iOS, y será de uso intuitivo y fácil de entender.

#### **1.5. Viabilidad de la Investigación**

##### **1.5.1. Viabilidad técnica**

Se dispone de servicios, herramientas, equipo y/o materiales y metodología, tales como:

- **Servicios:** Internet, Telefonía móvil, luz eléctrica.
- **Herramientas:** Node.js 12.18.2, Ionic 5.0, Cordova, Firebase, Visual Studio Code, Navegadores Google Chrome, Internet explorer y chrome.
- **Equipo y/o materiales:** Laptop Toshiba Satellite L45-SI, Dispositivo móvil Samsung galaxy s10+.
- **Metodología:** Mobile-D.

#### **1.6. Limitaciones del estudio**

Alguna de las limitaciones que se presentaron fueron la poca información en internet referente a la empresa de transporte público cristo morado, como también son:

- Para el uso del aplicativo móvil el usuario deberá tener acceso a internet.
- La aplicación funcionará solo para smartphones que cuenten desde la versión de Android 4.0 en adelante.
- Solo accederán al servicio del aplicativo los usuarios que tengan smartphones.

## **MARCO TEÓRICO**

### **2.1. ANTECEDENTES INTERNACIONALES**

(Yago Marco , 2015) en su proyecto de investigación “APLICACIÓN WEB Y MÓVIL PARA EL SEGUIMIENTO DE AUTOBUSES ESCOLARES” menciona que el objetivo de este proyecto es la creación y despliegue de una plataforma web, mediante el uso del framework Cakephp, la cual permite gestionar un servicio de geolocalización de flotas por GPS. Esta plataforma se centra en ofrecer, mediante Push Notifications o email, un servicio de alertas sobre la proximidad de buses escolares a una zona definida, es decir, a las paradas dentro de una ruta. Este servicio está pensado para informar a los padres de la llegada de sus hijos a su parada habitual antes de que ésta tenga lugar. La plataforma se compone de varios módulos PHP ya que por una parte ofrecemos la posibilidad de realizar labores de administración; p.ej., creación de usuarios, asignación de dispositivos.

### **2.2. ANTECEDENTES NACIONALES**

(Ordoñez Marín, 2019) en su tesis “DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL Y MONITOREO EN TIEMPO REAL QUE PERMITA MEJORAR EL SERVICIO DEL TRANSPORTE INTERNO DE LA UNIVERSIDAD NACIONAL DE CAJAMARCA.” menciona que la visión de la aplicación web proporcionará una herramienta en la que los usuarios de los buses de la Universidad Nacional de Cajamarca puedan realizar un monitoreo en tiempo real de estos, conociendo la ubicación y tiempo de llegada al paradero de destino. Adicionalmente, se simplificará el tiempo de espera, los recursos utilizados en el transporte diario de los usuarios de los buses de la Universidad Nacional de Cajamarca disminuirán ya que los usuarios utilizarán los buses de la Universidad Nacional de Cajamarca que es gratuito. La tecnología websocket, permitirá optimizar el tiempo de comunicación entre los clientes y el servidor, optimizando el envío y recepción de la información en tiempo real.

(Meza Romero & Leaño Pariona, 2018) en su tesis “SISTEMA DE MONITOREO DE UNA RED DE BUSES DE TRANSPORTE PÚBLICO E INFORMACIÓN PARA

USUARIOS EMPLEANDO TRANSCEPTORES GPS/GSM.” Menciona que el diseño del hardware de su sistema de monitoreo está enfocado principalmente para ser implementado en una empresa de transporte público como punto de partida en la creación de un sistema inteligente de transporte para nuestra capital. Este sistema consta de las siguientes partes: un paradero que permite la visualización de datos mediante una pantalla LCD y un receptor GSM/GPRS, el cual recibirá información procesada por la central de monitoreo mediante mensajes SMS; el hardware de monitoreo, instalado en la unidad de transporte público, compuesto principalmente por un transceptor GPS/GSM-GPRS, el cual envía mediante SMS datos de posición (latitud, longitud), fecha, hora y velocidad de desplazamiento del móvil hacia la central de monitoreo, la cual esta implementada con el software de procesamiento que hemos desarrollado y un transceptor GSM/GPRS que le permite comunicarse con el bus y con el paradero.

### **2.3. ANTECEDENTES LOCALES**

(Altamirano Ochicua, 2018) en su trabajo de investigación “APLICACIÓN MÓVIL DE GESTIÓN DE INFRACCIONES DE TRÁNSITO GEOREFERENCIADAS PARA INSPECTORES DE TRÁNSITO DE LA MUNICIPALIDAD PROVINCIAL DE ANDAHUAYLAS” menciona que su idea principal de esta aplicación es permitir a los inspectores de tránsito o una autoridad competente administrar sus intervenciones, brindándoles la información de los conductores vehículos. Además, disponer de servicios asociados tales como ver los datos generales del conductor y el vehículo, infracciones, empresa a la que pertenece, vigencias de sus seguros obligatorios de tránsito, tarjeta de circulación y su estado de licencias en tiempo real, también se georreferenciara la ubicación exacta mediante para su posterior corroboración. La plataforma será un aplicativo móvil desarrollado en Android Studio con el fin de que cualquier dispositivo móvil con conexión a internet pueda utilizarlo. Además, Esta aplicación móvil utiliza una base de datos el cual se encarga de almacenar los datos de todas las intervenciones de cada inspector en tiempo real. Nuestra Arquitectura se basa en el empleo de la metodología Mobile – D que se adecua a las características necesarias requeridas para que la aplicación tenga un óptimo funcionamiento, además que facilite el trabajo de desarrollo y diseño del aplicativo. Se realizó la validación funcional del aplicativo con los inspectores de municipalidad de provincial de Andahuaylas, donde

se tuvo como resultado la aceptación del prototipo. El resultado es la reducción de la tasa de accidentes de tránsito y la disminución del tiempo de intervención al conductor.

## **2.4. BASES TEÓRICA**

### **2.4.1. Servicios**

Los servicios son un conjunto de actividades que produce una determinada empresa (público o privada) con el objetivo de satisfacer las necesidades de los clientes, sin la necesidad de tener por qué ofrecer un producto tangible para ello. Con los servicios los consumidores tendrán que quedarse con la experiencia y la satisfacción de haberlo consumido. Por ejemplo: un servicio de transporte como el tren puede tener como parte física los trenes, los billetes, las estaciones, etc. Todas estas cosas son bienes tangibles, en cambio el transporte en sí es un servicio. (Méndez, 2020).

### **2.4.2. Transporte**

Transporte es el medio por el cual son trasladados datos, objetos o seres vivos de un lugar a otro. La palabra transporte deriva del latín cuyos compuestos léxicos son el prefijo trans- que indica de un lado a otro y portare que significa llevar. El transporte o también llamados medios de transporte son los vehículos, tanto aéreos, marítimos como terrestres, que llevan mercancías, animales o personas hacia un destino. Los transportes públicos, en este sentido, son empresas administradas por el Estado con el fin de facilitar la movilidad de las personas que viven en dicha ciudad, región o país. (Saltos, 2020)

#### **2.4.2.1. Tipos de transporte**

Según (Saltos, 2020), los tipos de transporte se clasifican generalmente por los medios en las cuales se mueven siendo éstas:

- **Transportes terrestres:** Los medios de transportes terrestres son aquellos que se mueven a través de vías terrestres como, por ejemplo, los carros, camiones, autobuses, metro, trenes o bicicletas.
- **Transportes aéreos:** Los medios de transportes aéreos se mueven a través del aire como, por ejemplo, los aviones, helicópteros y drones. Este último como medio de transporte es usado para el envío de encomiendas.

- **Transportes marítimos:** Los transportes marítimos son vehículos que utilizan cuerpos de agua como medio de traslado. Dentro de los medios marítimos podemos encontrar: barcos, buques, veleros, submarinos o transbordadores.

### 2.4.3. Sistema de transporte

según (Wikipedia, 2020), el sistema de transporte requiere de varios elementos, que interactúan entre sí, para la práctica del transporte y sus beneficios:

- **La infraestructura**, que es la parte física de las condiciones que se requieren para dar aplicación al transporte, es decir se necesitan de vías y carreteras para el transporte terrestre urbano, provincial, regional e internacional, se necesitan aeropuertos y rutas aéreas para el transporte aéreo, asimismo se requieren canales y rutas de navegación para el transporte naviero ya sean estos por mar o por ríos y lagos. Otra parte de la infraestructura son las paradas y los semáforos en cuanto al transporte urbano, en el transporte aéreo son las torres de control y el radar, y en las navales son los puertos y los radares.
- **El vehículo o móvil**, es el instrumento que permite el traslado de personas, cosas u objetos, de un lugar a otro.
- **El operador de transporte**, es la persona encargada de la conducción del vehículo ó móvil, en la cual se van a trasladar personas, cosas u objetos.
- **Las normas y leyes**, Es la parte principal del sistema de transportes, es la que dictamina la manera de trasladarse de un lugar a otro, asimismo es la que regula y norma la operación de todos los demandantes y ofertantes del servicio de transporte.

### 2.4.4. Transporte Público Urbano

Transporte público o transporte en común es el término aplicado al transporte colectivo de pasajeros. A diferencia del transporte privado, los viajeros del transporte público tienen que adaptarse a los horarios y a las rutas que ofrezca el operador y dependen en mayor o menor medida de la intervención regulatoria del Gobierno. Usualmente los viajeros comparten el medio de transporte y las distintas unidades están disponibles para el público en general.

Incluye diversos medios como autobuses, taxis, trolebuses, tranvías, trenes, ferrocarriles, suburbanos y bicicletas. (Wikipedia, 2020)

#### 2.4.5. Clasificación del Transporte Urbano de Pasajeros

- **Transporte privado:** operado por el dueño de la unidad, circulando en la vialidad proporcionada, operada y mantenida por el Estado. Automóvil, bicicleta, motocicleta, peatón. Tracción animal o el animal mismo. (Pastor, 2017).
- **Transporte de alquiler:** utilizado por cualquier persona que pague una tarifa en vehículos proporcionados por un operador, chofer o empleado, ajustándose a los deseos de movilidad del usuario. Taxis, remis, Uber. Servicios de repuestas a la demanda: servicios contratados. Es un transporte público. (Pastor, 2017).
- **Transporte público:** sistemas de transportación que operan con rutas fijas y horarios predeterminados y que pueden ser utilizados por cualquier persona a cambio del pago de una tarifa previamente establecida. (Pastor, 2017)

#### 2.4.6. Dispositivo Móvil

Es un dispositivo o aparato relativamente de tamaño pequeño, con algunas capacidades de procesamiento de datos, que puede tener una conexión permanente o intermitente a una red, con una memoria muy limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones para distintos usos. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles, hasta los navegadores GPS, pasando por los teléfonos móviles, los PDAs, videojuegos portátiles, o los Tablet PCs. Normalmente, cada uno de los diferentes tipos de dispositivos móviles está diseñado para uso individual, por tanto, poniendo un ejemplo, no puede ser usado como una computadora familiar, con diferentes usuarios, sino que solo puede ser usado por una sola persona. Así también cabe mencionar que su nombre le aporta sentido porque son aparatos que pueden ser transportados fácilmente, o estar integrados a otro aparato más grande, como complemento o como controlador. (Rodrigo Ramírez, 2020).

#### 2.4.7. Smartphone

Un teléfono celular inteligente (denominado en inglés smartphone) es técnicamente una computadora de bolsillo, a la cual se le anexan las capacidades de un teléfono móvil

tradicional (como son realizar llamadas telefónicas, enviar mensajes SMS, etc.). Podemos ubicar al smartphone a mitad de camino entre un teléfono celular convencional y una computadora portátil ya que une las capacidades de ambos extremos. (Sabini, 2019)

#### **MENCIONA LOS VENTAJAS Y DESVENTAJAS DE LOS SMARTPHONE:**

##### **Ventajas:**

- Utilizar el Correo Electrónico,
- Disfrutar de las Redes Sociales,
- Comunicarse instantáneamente con texto, llamadas o videollamadas cualquier persona de manera relativamente gratuita y directa gracias a las aplicaciones de Mensajería Instantánea,
- Mirar televisión, cine, videos y escuchar música,
- Realizar transacciones como compras on line, transferencias bancarias o reservas a un restaurante,
- Jugar a infinidad de juegos en forma individual o grupal con otros usuarios conectados desde cualquier parte del mundo,
- Ejecutar búsquedas de cualquier tipo de información,
- Utilizar una completísima variedad de aplicaciones que realizan diferentes tipos de tareas,
- Orientarse con la aplicación de GPS,
- Sacar fotografías, editarlas, construir películas, compartirlas, subirlas a la nube, etc.

##### **Desventajas:**

- La más significativa es la de la nomofobia.
- Tienen vida útil relativamente corta debido a la rapidísima evolución tecnológica que se vive en nuestros tiempos.
- Su fragilidad en sus componentes lo que hace que haya que tener un especial cuidado para evitar los golpes y caídas.

#### **2.4.8. Aplicativos Móviles**

Es un programa o conjunto de instrucciones lógicas creado para llevar a cabo una tarea específica solucionando una necesidad, a este programa se puede acceder fácilmente mediante una descarga, desde un teléfono inteligente o algún otro aparato móvil con acceso a internet. Es decir, Las aplicaciones móviles son programas diseñados para ejecutarse en teléfonos inteligentes, tabletas y otros dispositivos móviles. Estas aplicaciones suelen realizar una tarea específica, como informar el pronóstico del tiempo o mostrar mapas para la navegación. Los dispositivos móviles tienen requisitos especiales debido a sus pantallas pequeñas y opciones de entrada limitadas. Además, una pantalla táctil es generalmente la única forma de ingresar información en un dispositivo móvil. (Espinoza Choque, 2019)

Cuadro de comparación entre aplicación híbrida y nativa.

*Tabla 1. Ventajas de Aplicación Híbrida. Ventajas de Aplicación Nativa*

HERRAMIENTAS	VENTAJAS	DESVENTAJAS
Aplicación híbrida	<ul style="list-style-type: none"> <li>✓ Es versátil</li> <li>✓ Sin permisos</li> <li>✓ Mantenimiento sencillo.</li> <li>✓ Responsiva.</li> <li>✓ Costo accesible.</li> <li>✓ Mejor experiencia de uso.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Sus funciones son limitadas ya que no tienen acceso a todos los recursos del Smartphone.</li> <li>✓ Generalmente, requieren de una conexión a internet para funcionar.</li> <li>✓ Visualmente, no son tan atractivas como las nativas.</li> <li>✓ Su rendimiento es menor que el de una app nativa.</li> </ul>
Aplicación Nativa	<ul style="list-style-type: none"> <li>✓ Mejor rendimiento.</li> <li>✓ Diseños más atractivos.</li> <li>✓ Actualizaciones.</li> <li>✓ Notificaciones.</li> <li>✓ Sin conexión.</li> </ul>	<ul style="list-style-type: none"> <li>✓ No todas las plataformas pueden gozar de las mismas funciones.</li> <li>✓ Su costo de inversión es más elevado.</li> <li>✓ El código desarrollado solo sirve para una sola plataforma. Si deseamos que nuestra aplicación esté</li> </ul>



	<p>✓ Descarga en tiendas oficiales.</p>	<p>disponible para otro sistema, tendremos que diseñarla de nuevo utilizando otro lenguaje.</p>
--	---	---

Fuente: (Claudia, 2017) & (Orlando, 2016).

#### **2.4.9. Aplicaciones Híbridas**

Las aplicaciones híbridas utilizan tecnologías web (HTML, Javascript y CSS) pero no son ejecutadas por un navegador. En su lugar, se ejecutan en un contenedor web (webview), como parte de una aplicación nativa, la cual está instalada en el dispositivo. Desarrollo de Aplicaciones Móviles Multiplataforma. Desde una aplicación híbrida es posible acceder a las capacidades del dispositivo, a través de diversas API. Las aplicaciones híbridas ofrecen grandes ventajas permitiendo la reutilización de código en las distintas plataformas, el acceso al hardware del dispositivo, y la distribución a través de las tiendas de aplicaciones. (Delía, 2017)

#### ***¿Por qué usar aplicaciones híbridas para desarrollo de aplicaciones en las mipymes?***

Los desarrolladores corporativos utilizan aplicaciones híbridas para hacer que el soporte del número de dispositivos móviles en la empresa lleve menos tiempo y sea menos costoso. Un enfoque híbrido también facilita las tareas administrativas en el entorno BYOD (trae tu propio dispositivo), donde se debe soportar muchos sistemas operativos móviles diferentes y formatos de dispositivos. Los desarrolladores privados utilizan el desarrollo híbrido para llegar la mayoría de clientes potenciales al poner sus aplicaciones en múltiples tiendas de aplicaciones sin tener que reescribir las aplicaciones para cada tipo de dispositivo. (ROMERO, VALERIO, & FERREIRA, 2019).

#### **2.4.10. Aplicaciones Nativas**

Las App Nativas (o client-side applications) son aplicaciones hechas específicamente para el dispositivo. Están desarrolladas para ser utilizadas en una plataforma específica, con un lenguaje específico y permitirán realizar todo el código en el dispositivo móvil, también offline (sin conexión a Internet) e interactuar mejor con el dispositivo, utilizando su capacidad al 100% como GPS, Galería de fotos y mucho más. (mobincude, 2020)

#### **2.4.11. Sistemas operativos**

El Sistema Operativo es el programa (o software) más importante de un Computador. Para que funcionen los otros programas, cada computador de uso general debe tener un sistema operativo. Los sistemas operativos realizan tareas básicas, tales como reconocimiento de la conexión del teclado, enviar la información a la pantalla, no perder de vista archivos y directorios en el disco, y controlar los dispositivos periféricos tales como impresoras, escáner, etc. (Bustamante B, 2016)

#### **2.4.12. Sistema operativo móviles**

Un sistema operativo móvil o (SO) móvil, es un sistema que controla un dispositivo móvil al igual que los PCs que utilizan Windows o Linux, los dispositivos móviles tienen sus sistemas operativos como Android, IOS entre otros. Los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica. A medida que los teléfonos móviles crecen en popularidad, los sistemas operativos con los que funcionan adquieren mayor importancia. (Castellanos, 2020)

#### **2.4.13. Tipos de sistemas móviles**

- **ANDROID:** Es una plataforma de software para dispositivos móviles que incluye un Sistema Operativo y aplicaciones de base. Android es un conjunto de herramientas y aplicaciones vinculadas a una distribución Linux para dispositivos móviles. El Sistema Operativo Android es de código abierto, gratuito y no requiere pago de licencias. (Bustamante B, 2016)
- **IOS:** Es un sistema operativo móvil desarrollado por Apple Inc. Este sistema operativo móvil está basado en el concepto de manipulación directa, es decir, que el usuario puede interactuar directamente con la pantalla del dispositivo por medio

de gestos multitáctiles como toques, pellizcos y deslices. Este sistema fue originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros. El sistema iOS es de código cerrado exclusivamente para los dispositivos móviles Apple. (Bustamante B, 2016)

- **SYMBIAN:** Symbian fue un sistema operativo propiedad de Nokia, el objetivo del Symbian era crear un sistema operativo para terminales móviles que pudiera competir con el de Palm o el Windows Mobile de Microsoft y posteriormente Android de Google , iOS de Apple, Windows Phone de Microsoft y BlackBerry OS de BlackBerry.
- **WINDOWS PHONE:** Sistema operativo móvil desarrollado por Microsoft, como sucesor de Windows Mobile. A diferencia de su predecesor está enfocado en el mercado de consumo en lugar de en el mercado empresarial. Con Windows Phone; Microsoft ofrece una nueva interfaz de usuario que integra varios de sus servicios propios como OneDrive, Skype y Xbox Live en el sistema operativo. Este sistema operativo es de código de cerrado. (Bustamante B, 2016)

#### **2.4.14. Internet**

El Internet, algunas veces llamado simplemente "La Red", es un sistema mundial de redes de computadoras, un conjunto integrado por las diferentes redes de cada país del mundo, por medio del cual un usuario en cualquier computadora puede, en caso de contar con los permisos apropiados, acceder información de otra computadora y poder tener inclusive comunicación directa con otros usuarios en otras computadoras. (Milenium, 2020)

#### **2.4.15. Geolocalización**

La geolocalización ayuda a identificar la ubicación geográfica real de los objetos, como dispositivos móviles o cualquier terminal conectada a Internet. El término "geolocalización" se refiere tanto al proceso de localización geográfica de los objetos como a la ubicación geográfica real identificada. (SendPulse, 2018)

#### **2.4.16. Base de Datos No SQL**

Las bases de datos NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. Esta página incluye recursos que lo ayudan a comprender mejor las bases de datos NoSQL y comenzar a usarlas. (Amazon Web Services, 2020)

### **¿Por qué debería usar una base de datos NoSQL?**

Amazon Web Services(2020) indica que las bases de datos NoSQL se adaptan perfectamente a muchas aplicaciones modernas, como dispositivos móviles, web y juegos, que requieren bases de datos flexibles, escalables, de alto rendimiento y altamente funcionales para proporcionar excelentes experiencias de usuario.

- **Flexibilidad:** las bases de datos NoSQL generalmente ofrecen esquemas flexibles que permiten un desarrollo más rápido y más iterativo. El modelo de datos flexible hace que las bases de datos NoSQL sean ideales para datos semiestructurados y no estructurados. Amazon Web Services(2020)
- **Escalabilidad:** las bases de datos NoSQL generalmente están diseñadas para escalar usando clústeres distribuidos de hardware en lugar de escalar añadiendo servidores caros y sólidos. Algunos proveedores de la nube manejan estas operaciones en segundo plano, como un servicio completamente administrado. Amazon Web Services(2020)
- **Alto rendimiento:** la base de datos NoSQL está optimizada para modelos de datos específicos y patrones de acceso que permiten un mayor rendimiento que el intento de lograr una funcionalidad similar con bases de datos relacionales. Amazon Web Services(2020)
- **Altamente funcional:** las bases de datos NoSQL proporcionan API altamente funcionales y tipos de datos que están diseñados específicamente para cada uno de sus respectivos modelos de datos. Amazon Web Services(2020)

#### **2.4.17. Firebase**

Firebase se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero. La plataforma

está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web. Contiene diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades. (Pérez cardona, 2016)

Pérez cardona(2016) indica en sus principales características lo siguiente:

- **Desarrollo:** Firebase permite la creación de mejores apps, minimizando el tiempo de optimización y desarrollo, mediante diferentes funciones.
- **Analítica:** Tener un control máximo del rendimiento de la app mediante métricas analíticas, todo desde un único panel y de forma gratuita.
- **Poder de Crecimiento:** Permite gestionar de manera fácil todos los usuarios de las aplicaciones.
- **Monetización:** Mediante AdMob, Firebase permite que puedas ganar dinero.
- **Rapidez:** Implementar Firebase puede ser fácil y rápido, gracias a su API que es muy intuitiva, sostenida en un solo SDK
- **Agilidad:** Firebase ofrece apps multiplataforma con una APIs integradas a SDK individuales para iOS, Android y Javascript.

#### 2.4.18. Tiempo Real

Los sistemas en tiempo real están presentes en la vida diaria, prácticamente en todo lo que nos rodea: en los aviones, trenes y automóviles, en el televisor, la lavadora o el horno de microondas, en los teléfonos celulares y en las centrales telefónicas digitales. Son un elemento imprescindible para garantizar la generación, transmisión y distribución de la energía eléctrica y para asegurar la calidad y la seguridad de incontables procesos industriales. (Wikipedia, 2020)

#### 2.4.19. Framework

en el desarrollo de software un framework es una estructura conceptual y tecnológica de asistencia definida, generalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. un framework es una especie de plantilla, esquema o estructura conceptual de base tecnológica que permite trabajar de una forma mucho más sencilla. De este modo **se evitan posibles errores de programación**, su uso se centra fundamentalmente en el desarrollo de aplicaciones web. (Valdivia Saavedra, 2020)

#### 2.4.20. IONIC

Se le denomina un SDK (Software Development Kit) completo y de código abierto para el desarrollo de aplicaciones móviles híbridas. La versión original fue lanzada en 2013 y construida sobre AngularJS y Apache Cordova. Ionic proporciona herramientas y servicios para desarrollar aplicaciones móviles híbridas utilizando tecnologías web como CSS, HTML5 y Sass (Syntactically awesome style sheets). Las aplicaciones pueden compilarse con estas tecnologías web y luego distribuirse a través de tiendas de aplicaciones nativas para ser instalado en dispositivos aprovechando Cordova. Ionic es la plataforma que facilita la creación de aplicaciones móviles con tecnologías web para los programadores web. La plantilla de Ionic permite a los desarrolladores crear aplicaciones móviles en diferentes plataformas que pueden instalarse en teléfonos con Android e iOS. (mySolutions, 2018)

#### ¿Por qué usar Ionic?

Ionic facilita la creación de aplicaciones híbridas utilizando HTML5 debido a su Framework de código abierto y también es completamente gratuito. El uso de Angular JS lo hace perfecto para el desarrollo de aplicaciones altamente interactivas y tiene una gran gama de herramientas y servicios que hacen que la ejecución de Ionic sea bastante simple. Ionic viene con CLI (Command-Line Interface), que ayuda a los desarrolladores móviles a construir y probar aplicaciones iónicas en cualquier plataforma. Permite a los usuarios desplazarse por miles de listas sin que su rendimiento se vea afectado. Los usuarios pueden crear sus propias aplicaciones, personalizarlas para Android, iOS y desplegarlas. (mySolutions, 2018)

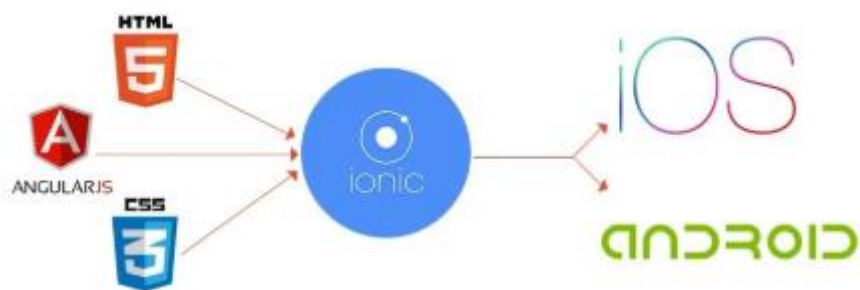


Figura 1: Funcionamiento de IONIC

Fuente: (mySolutions, 2018)

### 2.4.21. Cordova

Apache Cordova es un marco de código abierto que permite a los desarrolladores web usar su contenido HTML, CSS y JavaScript para crear una aplicación nativa para una variedad de plataformas móviles. Cordova toma su aplicación web y la procesa dentro de un WebView nativo. Un WebView es un componente de la aplicación (como un botón o una barra de pestañas) que se utiliza para mostrar contenido web dentro de una aplicación nativa. Puede pensar en un WebView como un navegador web sin ninguno de los elementos estándar de la interfaz de usuario, como un campo de URL o una barra de estado. La aplicación web que se ejecuta dentro de este contenedor es como cualquier otra aplicación web que se ejecute dentro de un navegador móvil: puede abrir páginas HTML adicionales, ejecutar código JavaScript, reproducir archivos multimedia y comunicarse con servidores remotos. Este tipo de aplicación móvil a menudo se denomina aplicación híbrida. (Griffith, 2020)



Figura 2. Imagen de Apache Cordova.

Fuente: (Griffith, 2020)

### **2.4.22. Angular**

Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application). Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse. (QUALITY DEVS, 2019)

### **2.4.23. Node.js**

Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. También aporta muchos beneficios y soluciona muchísimos problemas, por lo que sería más que interesante realizar nuestro curso de Node.js para obtener las bases, conceptos y habilidades necesarias que nos motiven a profundizar en sus opciones e iniciar la programación. (Lucas, 2019)

Node.js fue creado por los desarrolladores originales de JavaScript. Lo transformaron de algo que solo podía ejecutarse en el navegador en algo que se podría ejecutar en los ordenadores como si de aplicaciones independientes se tratara. Gracias a Node.js se puede ir un paso más allá en la programación con JavaScript no solo creando sitios web interactivos, sino teniendo la capacidad de hacer cosas que otros lenguajes de secuencia de comandos como Python pueden crear. (Lucas, 2019)

### **¿Cómo funciona Node.js?**

El funcionamiento interno del entorno de ejecución para JavaScript, Node.js, es bastante interesante. En comparación con las técnicas tradicionales de servicio web donde cada conexión (que crea una solicitud) genera un nuevo subproceso, ocupando la RAM del sistema y regularmente maximizando la cantidad de RAM



disponible, Node.js opera en un solo subproceso, utilizando el modelo entrada y salida sin bloqueo de la salida, lo que le permite soportar decenas de miles de conexiones al mismo tiempo mantenidas en el bucle de eventos. (Lucas, 2019)

#### **2.4.24. PMBOK**

Es la publicación principal de PMI y es un recurso fundamental para la gestión efectiva de proyectos en cualquier industria. Se ha actualizado para reflejar las últimas buenas prácticas en gestión de proyectos. La Guía PMBOK® es desarrollada por profesionales activos y expertos en la materia, luego revisada por la comunidad de gestión de proyectos antes de su lanzamiento, para asegurar que siempre refleje el estado actual de la profesión. (Project Management Institute, 2017)

#### **2.4.25. PMI**

Es una organización sin fines de lucro que avanza la profesión de la dirección de proyectos a través de estándares y certificaciones reconocidas mundialmente, a través de comunidades de colaboración, de un extenso programa de investigación y de oportunidades de desarrollo profesional. (Project Management Institute, 2017)

#### **2.4.26. Metodología Mobile-D**

Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: extreme programming, Crystal Methodologies y Rational Unified Process. Esta metodología está enfocada al desarrollo de software móvil. (Hernandez Vasquez, 2018)

El objetivo del método Mobile-D es conseguir ciclos de desarrollo muy rápidos en equipos muy pequeños. Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: extreme programming, Crystal Methodologies y Rational Unified Process. (Rodríguez, 2011)

**Se compone distintas fases:**

- **Exploración:** Se centra la atención en la planificación y a los conceptos básicos del proyecto. Aquí es donde hacemos una definición del alcance del proyecto y su establecimiento con las funcionalidades donde queremos llegar.
- **Iniciación:** Configura el proyecto identificando y preparando todos los recursos necesarios como hemos comentado anteriormente en esta fase la dedicaremos un día a la planificación y el resto al trabajo y publicación.
- **Producto:** Se repiten interactivamente las subfases. Se usa el desarrollo dirigido por pruebas (TDD), antes de iniciar el desarrollo de una funcionalidad debe existir una prueba que verifique su funcionamiento. En esta fase podemos decir que se lleva a acabo toda la implementación.
- **Estabilización:** Se realizan las acciones de integración para enganchar los posibles módulos separados en una única aplicación.
- **Pruebas:** Una vez parado totalmente el desarrollo se pasa una fase de testeo hasta llegar a una versión estable según lo establecido en las primeras fases por el cliente. Si es necesario se reparan los errores, pero no se desarrolla nada nuevo.

## PROPUESTA DE SOLUCIÓN

### 3.1. Ingeniería de proyecto

Para poder desarrollar la aplicación multiplataforma de geolocalización en tiempo real de la empresa de transporte cristo morado de Andahuaylas, constará de gestión del proyecto, metodología de desarrollo y diferentes herramientas tales como software y hardware.

### 3.2. Bases de datos de la aplicación

En la siguiente imagen se muestra la estructura de la base de datos NoSQL FIREBASE.

- **Módulo del administrador**

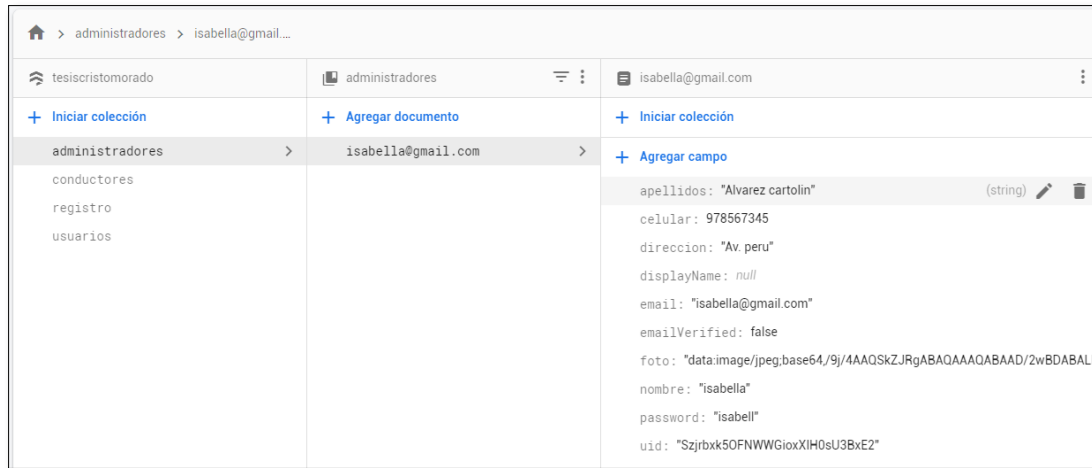


Figura 3: Base de datos del módulo administrador

Fuente: Elaboración propia

- **Módulo del conductor**

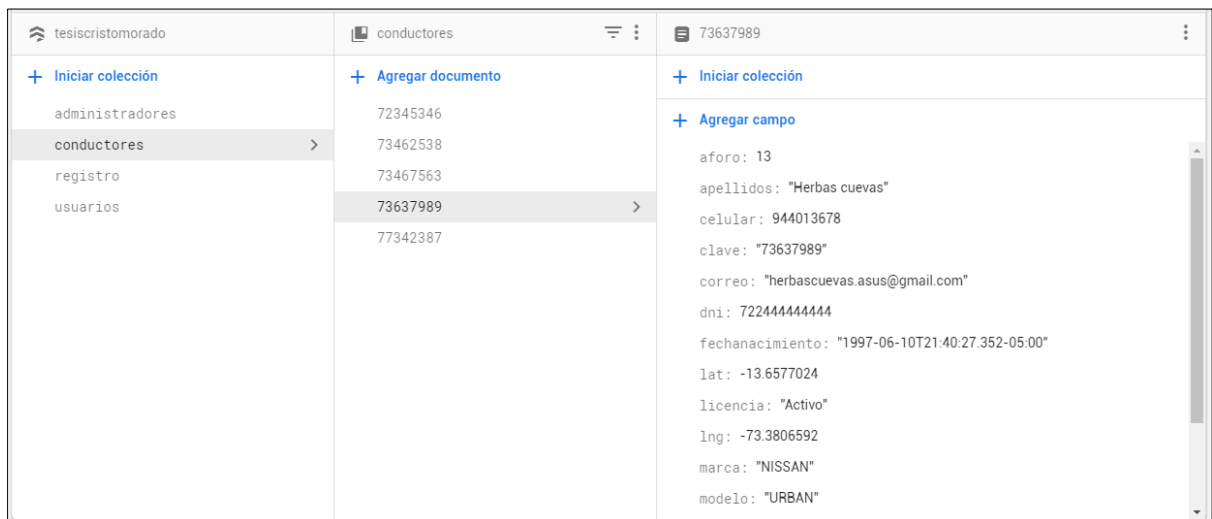


Figura 4: Base de datos módulo conductor

Fuente: Elaboración propia

- **Módulo del usuario**

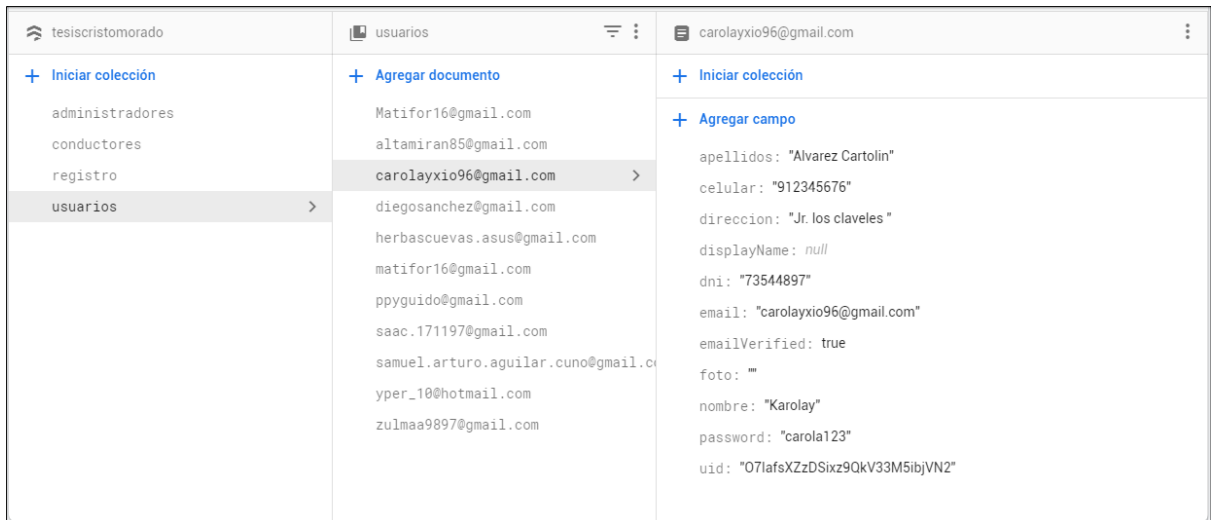


Figura 5: Base de datos módulo conductor

Fuente: Elaboración propia

- **Módulo del reporte de salida y llegada de vehículos en servicio**

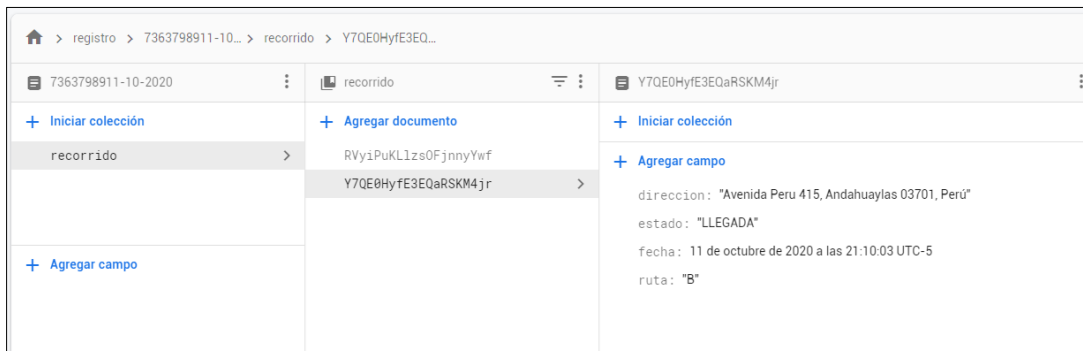


Figura 6: Base de datos de reporte de salida y llegada del vehículo en servicio

Fuente: Elaboración propia.

### 3.3. Diseño modular para sistema operativo Android.

El desarrollo de la aplicación tendrá los siguientes módulos:

- **MÓDULO ADMINISTRADOR**
  - Iniciar sesión:  
Página donde el administrador podrá autenticarse en la aplicación.

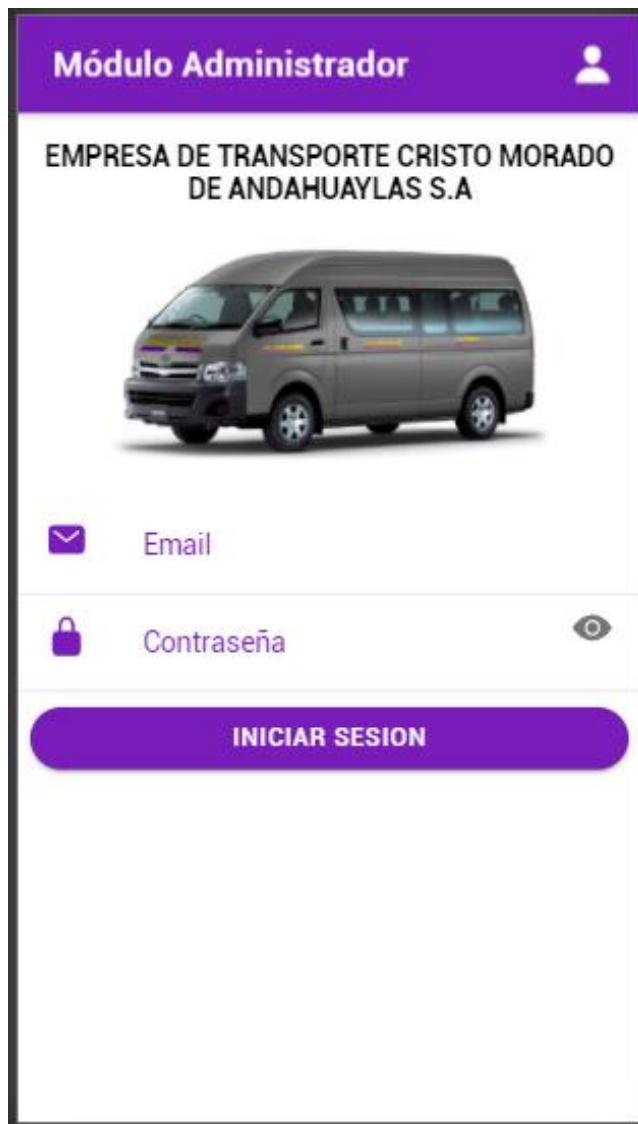


Figura 7: Diseño del módulo administrador iniciar sesión

Fuente: Elaboración propia

- Menu: página donde muestra todas las páginas y rutas de la aplicación.



*Figura 8: Diseño del módulo administrador menú principal*

Fuente: Elaboración propia

- Geolocalización: página donde se visualiza la geolocalización en tiempo real de todos los vehículos en servicio.

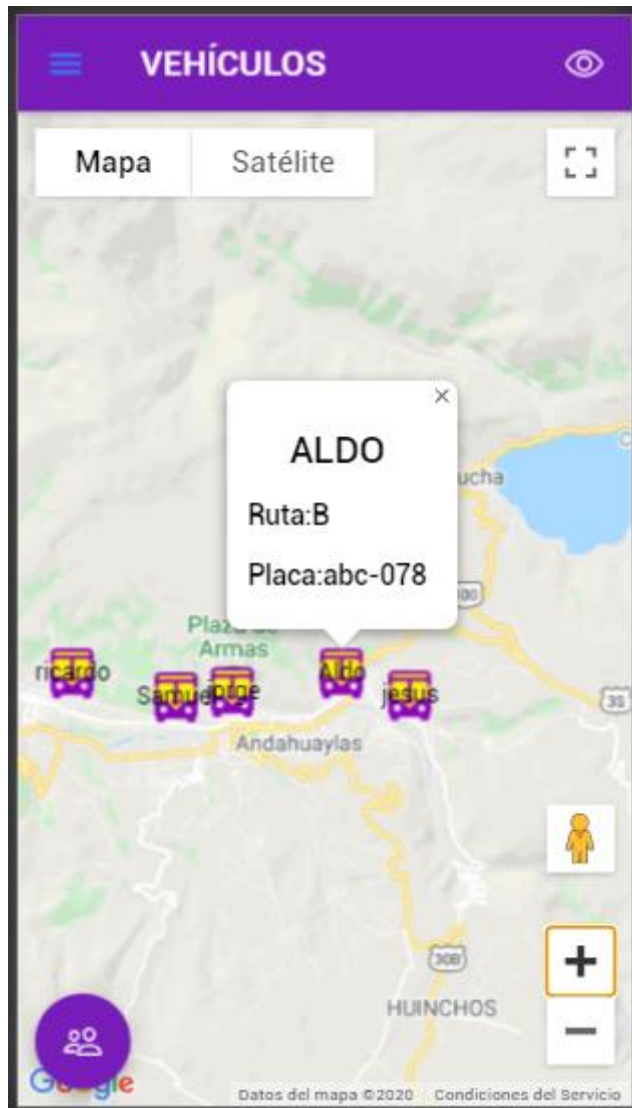


Figura 9: Diseño del módulo administrador de geolocalización de los vehículos en servicio  
Fuente: Elaboración propia

- Perfil: Pagina donde se muestra los datos personales del administrador.



*Figura 10: Diseño del módulo administrador Perfil*

Fuente: Elaboración propia

- Registro de administrador: página para el registro respectivo de un nuevo administrador.



REGISTRO ADMINIST...

Nombre

Apellidos

Dirección

Celular

Email

Contraseña

REGISTRAR

*Figura 11: Diseño del módulo administrador de registro*

Fuente: Elaboración propia

- Registro conductor: página donde se registra los conductores.

The image shows a mobile application interface for driver registration. At the top, there is a purple header with a hamburger menu icon on the left, the word 'CONDUCTOR' in the center, and a back arrow on the right. Below the header, the form is titled 'Datos personales'. It contains several input fields: 'clave', 'nombre', 'Apellidos', 'Correo', and 'DNI'. The 'Fecha de nacimiento' field is followed by a 'Select Date' button. Below that is the 'Celular' field. The 'Licencia de conducir (BREVETE)' field is followed by an 'Activo' checkbox.

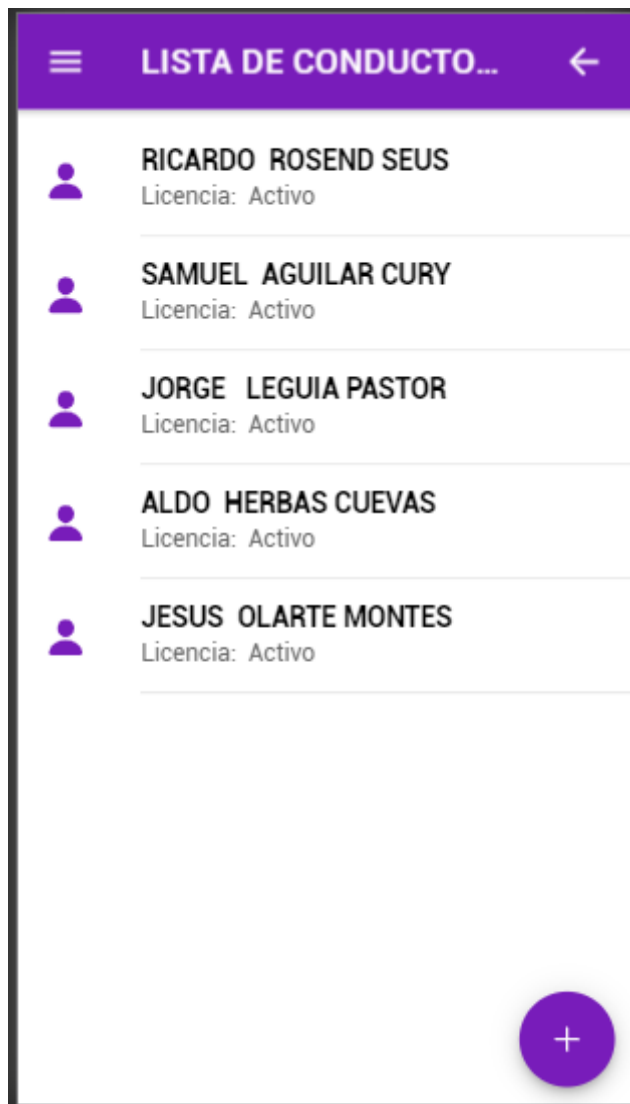
*Figura 12: Diseño del módulo administrador de registro conductor*  
Fuente: elaboración propia

The image shows a mobile application interface for driver registration. At the top, there is a purple header with a hamburger menu icon on the left, the word 'CONDUCTOR' in the center, and a back arrow on the right. Below the header, there is a radio button labeled 'Inactivo'. A section titled 'Datos del Vehículo' contains several input fields: 'Marca', 'Modelo', 'Placa del Vehículo', 'Aforo', and 'Seguro Obligatorio de Accidentes de Tr...'. Below these fields are two radio buttons: 'Activo' (unselected) and 'Inactivo' (selected). At the bottom of the form is a purple button labeled 'CREAR'.

*Figura 13: Diseño del módulo administrador de registro conductor*

Fuente: Elaboración propia

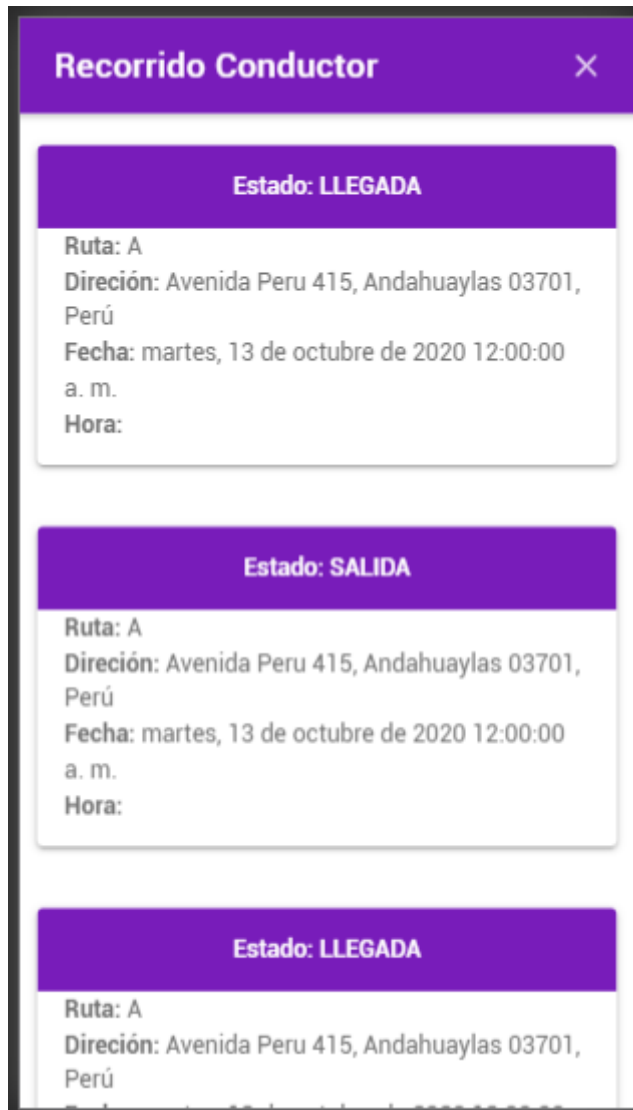
- Lista de conductores: Pagina donde se muestra la lista de conductores.



*Figura 14: Diseño del módulo administrador lista de conductores*

Fuente: Elaboración propia

- Reporte de salida y llegada: página donde se muestra el reporte de cada uno de los conductores.



*Figura 15: Diseño del módulo administrador reporte de salida y llegada*  
Fuente: Elaboración propia

- Actualizar conductor: página para actualizar al conductor.

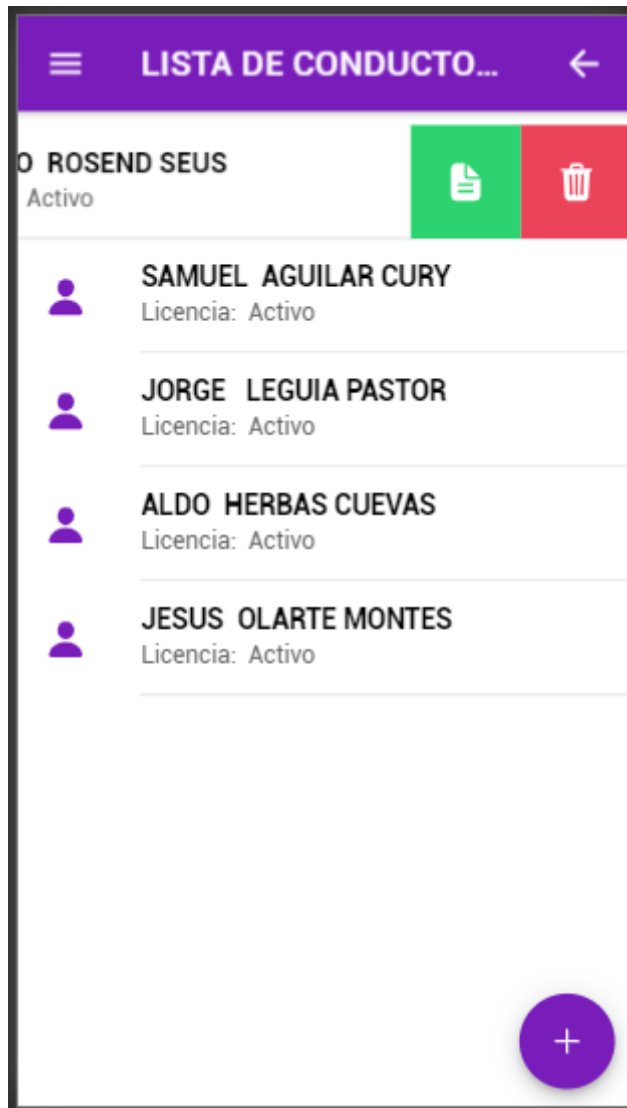
The image shows a mobile application interface for updating driver information. The title bar is purple with the text 'Actualizar-conductores' and a close button (X). Below the title bar, the section 'Datos personales' is displayed. The form contains the following fields:

- nombre: JORGE
- Apellidos: LEGUIA PASTOR
- Correo: jorge@gmail.com
- DNI: 73467563
- Fecha de nacimiento: 17 12 1988
- Celular: 912372645
- Licencia de conducir (BREVETE):
  - Activo
  - Inactivo

Figura 16: Diseño del módulo administrador actualizar conductor

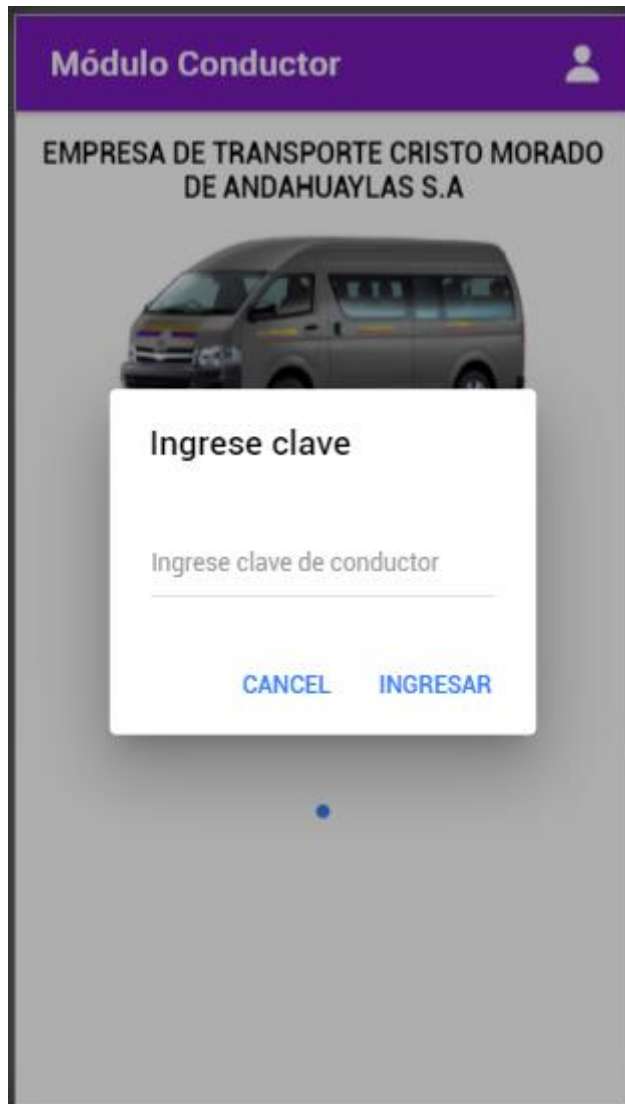
Fuente: Elaboración propia

- **Eliminar conductor:** Pagina para eliminar a un conductor.



*Figura 17: Diseño del módulo administrador eliminar Conductor*  
Fuente: Elaboración propia

- **MODULO CONDUCTOR**
  - Iniciar sesión con clave: Página donde el conductor inicia sesión mediante una clave, generado por el administrador.



*Figura 18: Diseño del módulo conductor iniciar sesión con clave*  
Fuente: Elaboración propia.

- Geolocalización de ubicación actual: Página donde se observa la ubicación actual y la geolocalización en tiempo real del conductor.



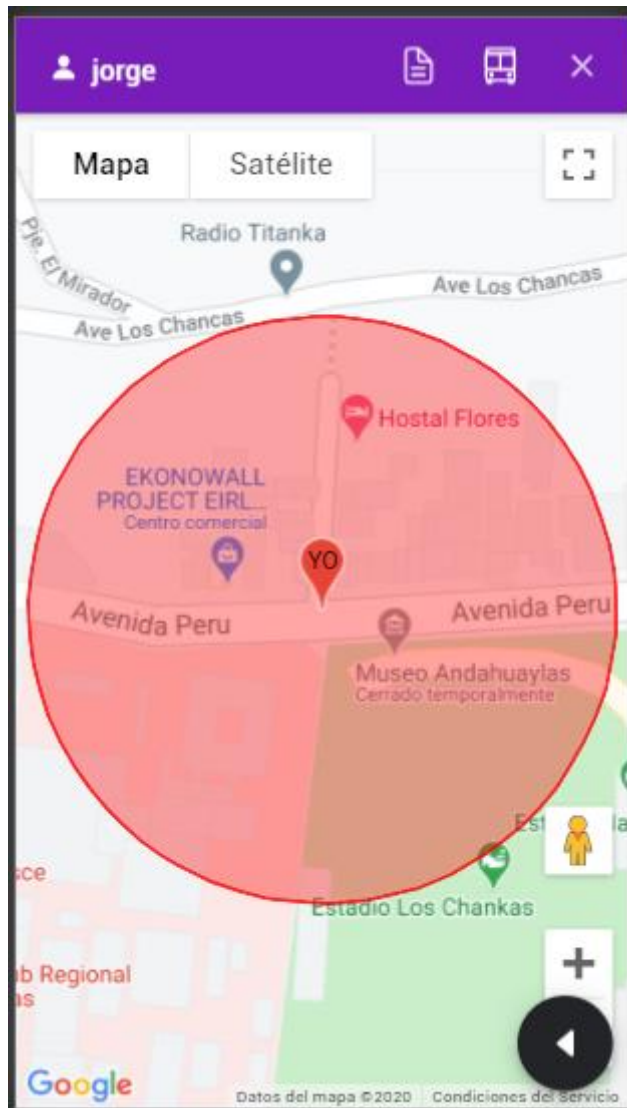


Figura 19: Diseño del módulo conductor de ubicación actual

Fuente: Elaboración propia

- Geolocalización de vehículos: Pagina donde se muestran a todos los vehículos en servicio.



*Figura 20: Diseño del módulo conductor geolocalización de vehículos*

Fuente: Elaboración propia

- Orden de circulación de los vehículos mediante una alerta: Pagina donde se visualiza las alertas cuando otro vehículo entra en su radio de 100 metros.



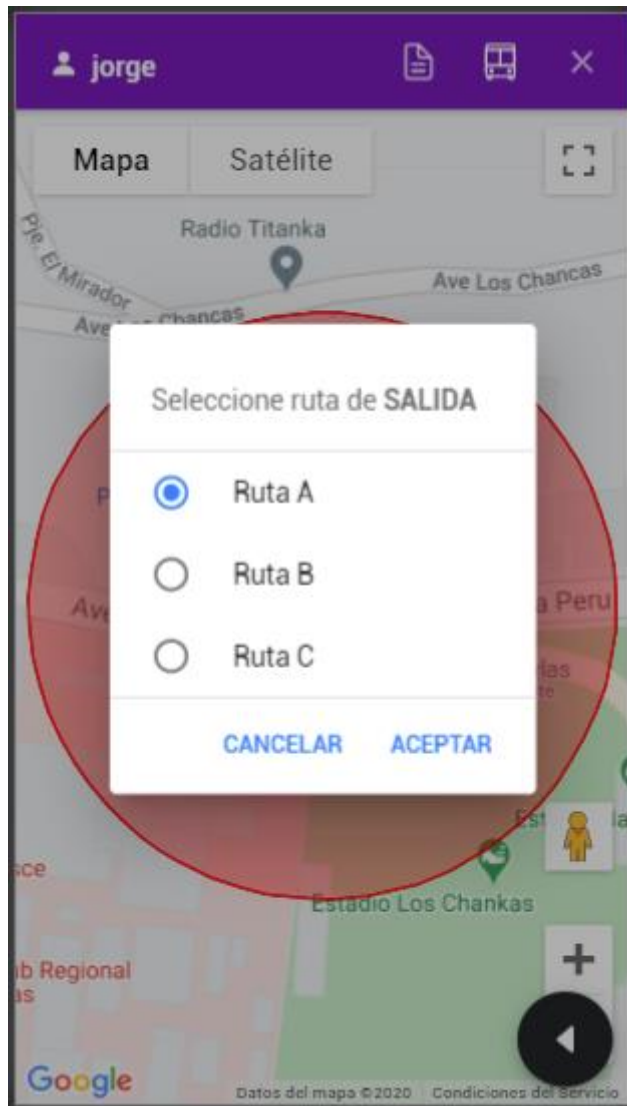
Figura 21: Diseño del módulo conductor orden de circulación de los vehículos mediante una alerta

Fuente: Elaboración propia

- Control registro de salida y llegada: página donde se reporta la salida y llegada de los vehículos en servicio a los paraderos iniciales y finales.



Figura 22: Diseño del módulo conductor control registro de salida y llegada  
Fuente: Elaboración propia



*Figura 23: Diseño del módulo conductor control registro de salida y llegada*  
Fuente: Elaboración propia

- **MODULO USUARIO**

- Iniciar sesión: página donde los usuarios podrán autenticarse en la aplicación.

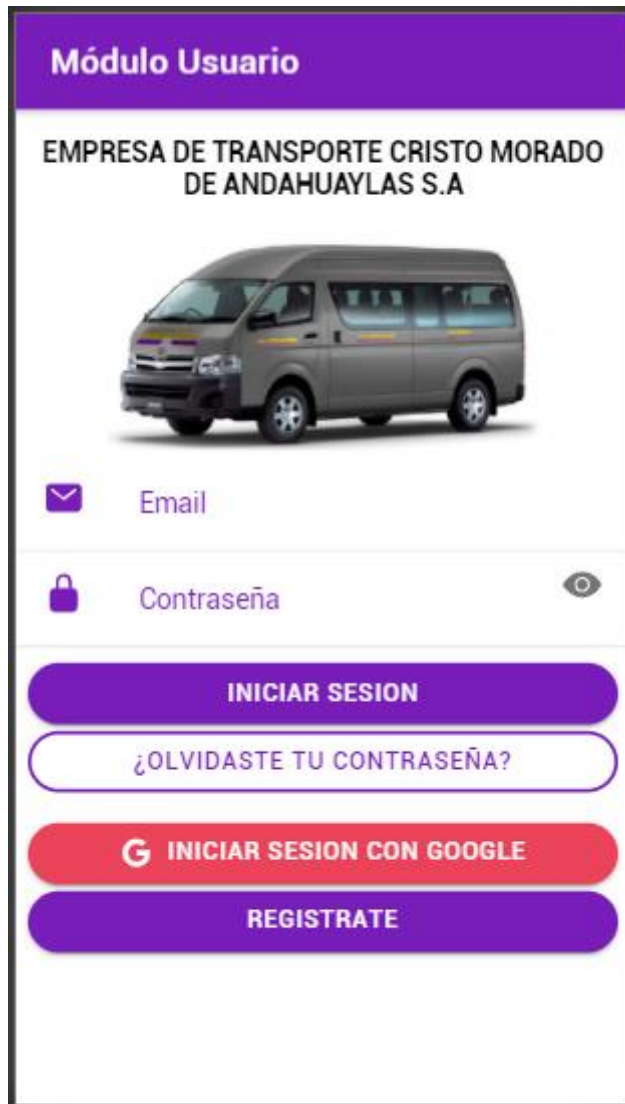


Figura 24: Diseño del módulo usuario iniciar sesión

Fuente: Elaboración propia

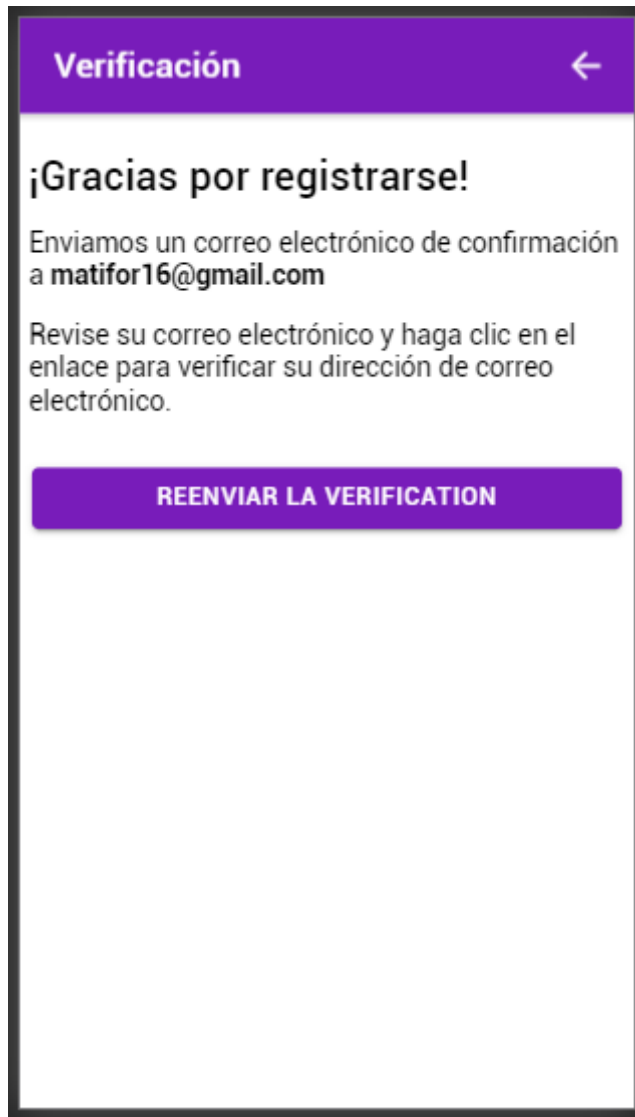
- Registro: página donde el usuario podrá registrarse en la aplicación.

The image shows a mobile application interface for a registration form. At the top, there is a purple header bar with the word "REGISTROS" in white capital letters on the left and a white left-pointing arrow on the right. Below the header, the form consists of several white input fields with thin grey borders, each preceded by a label: "Nombre:", "Apellidos:", "DNI:", "Dirección:", "Celular:", "Email:", and "Contraseña:". To the right of the "Contraseña:" label is a small grey eye icon. At the bottom of the form is a wide, rounded purple button with the word "REGISTRAR" in white capital letters.

*Figura 25: Diseño del módulo usuario registro*

Fuente: Elaboración propia

- Verificar correo: apartado donde el usuario podrá verificar su correo electrónico.

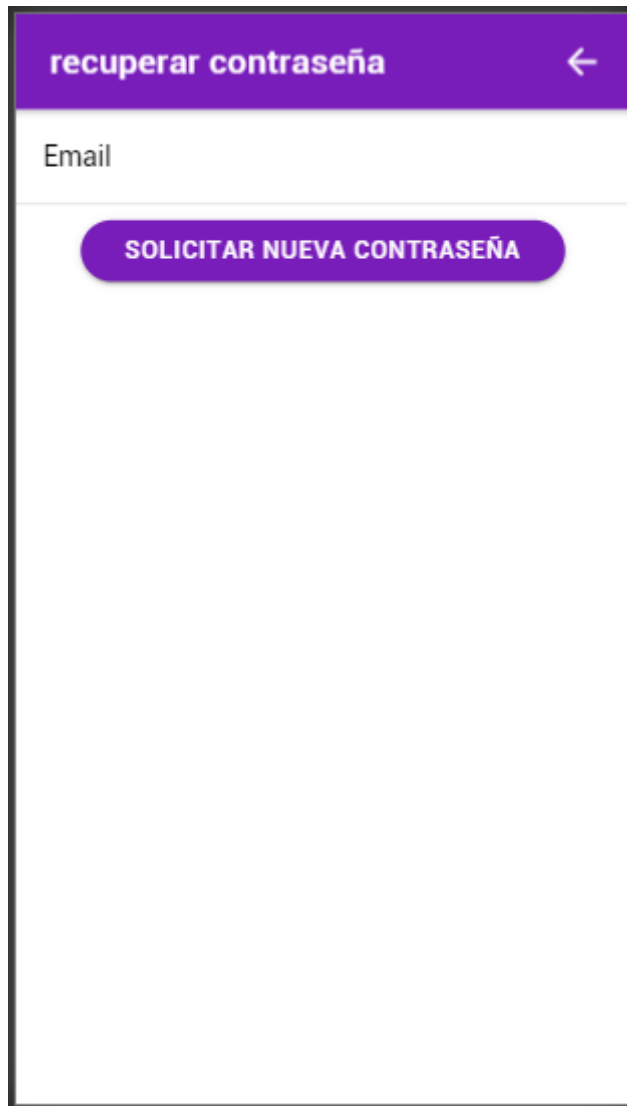


*Figura 26: Diseño del módulo usuario verificar correo*

Fuente: Elaboración propia

- Recuperar contraseña: Pagina donde el usuario podrá recuperar su contraseña.





*Figura 27: Diseño del módulo usuario recuperar contraseña*

Fuente: Elaboración propia

- Geolocalización de vehículos: página donde los usuarios podrán visualizar la geolocalización en tiempo real de todos los vehículos en servicio.

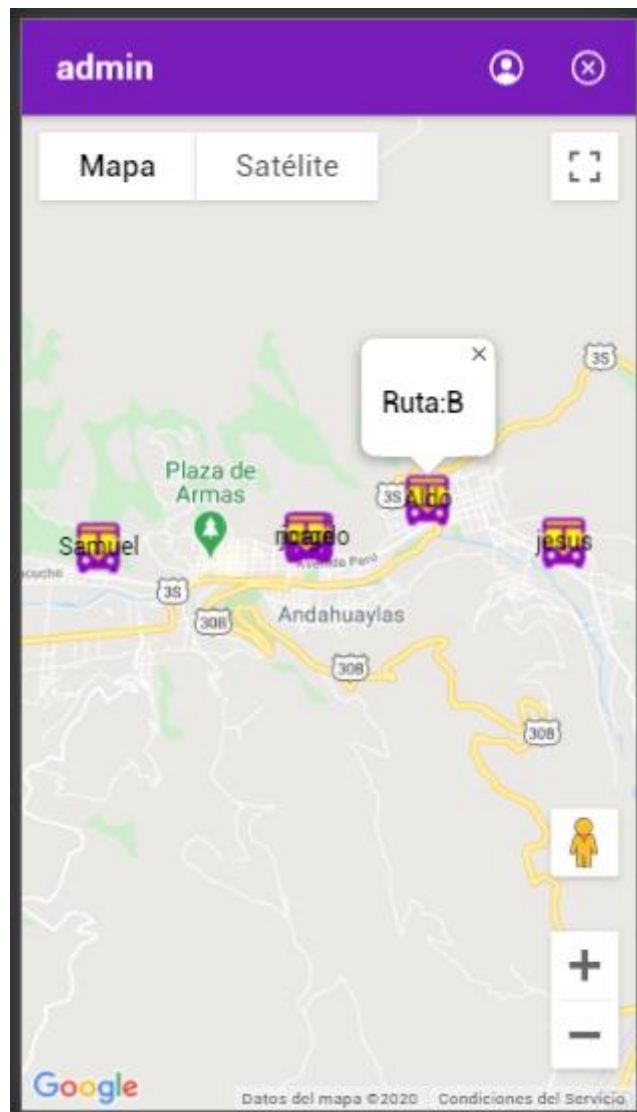


Figura 28: Diseño del módulo usuario geolocalización de vehículos

Fuente: Elaboración propia

- Perfil: Pagina donde el usuario podrá visualizar todos sus datos personales.

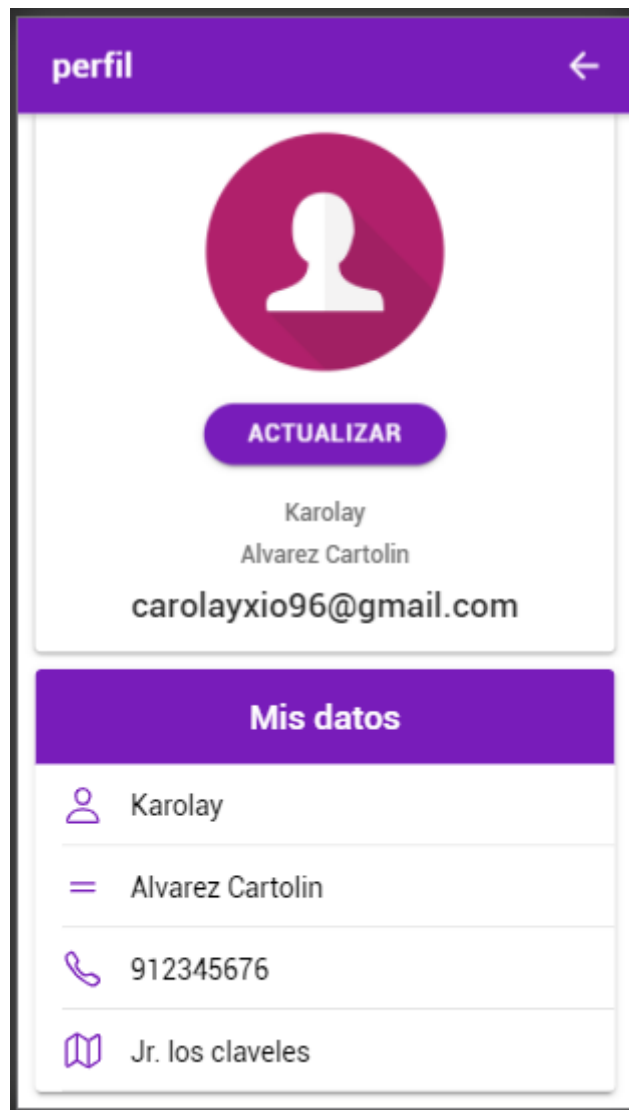


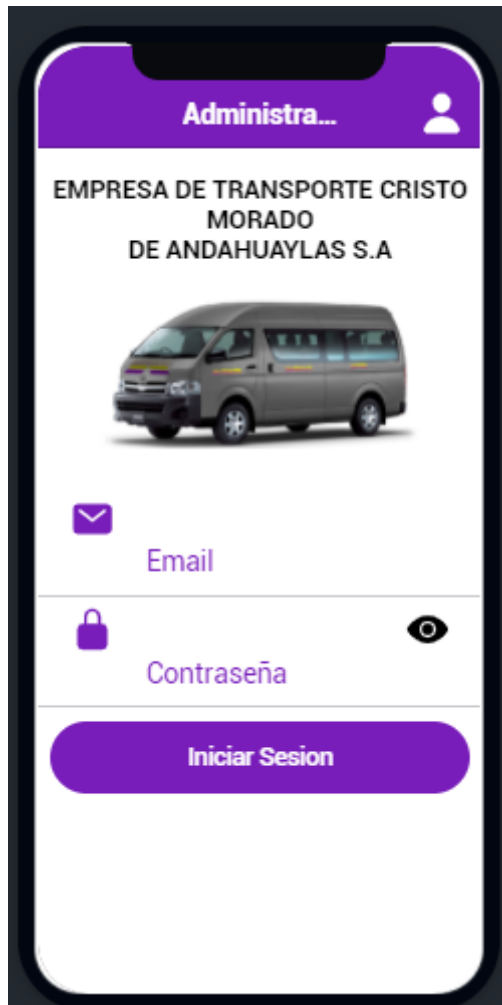
Figura 29: Diseño del módulo usuario perfil

Fuente: Elaboración propia

### 3.4. Diseño modular en sistema operativo IOS

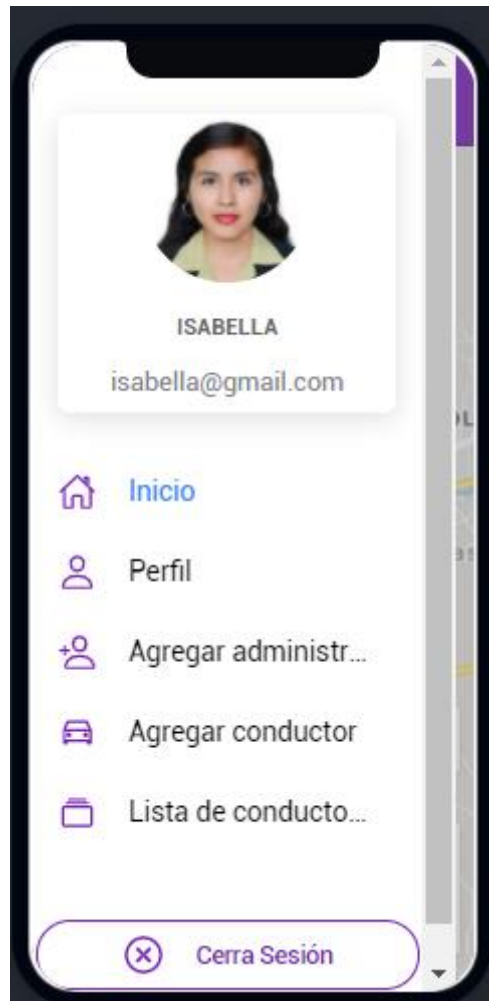
El desarrollo de la aplicación tendrá los siguientes módulos en su versión IOS:

- MÓDULO ADMINISTRADOR
  - Iniciar sesión:  
Página donde el administrador podrá autenticarse en la aplicación.



*Figura 30: Diseño del módulo administrador iniciar sesión versión Ios*  
Fuente: Elaboración propia

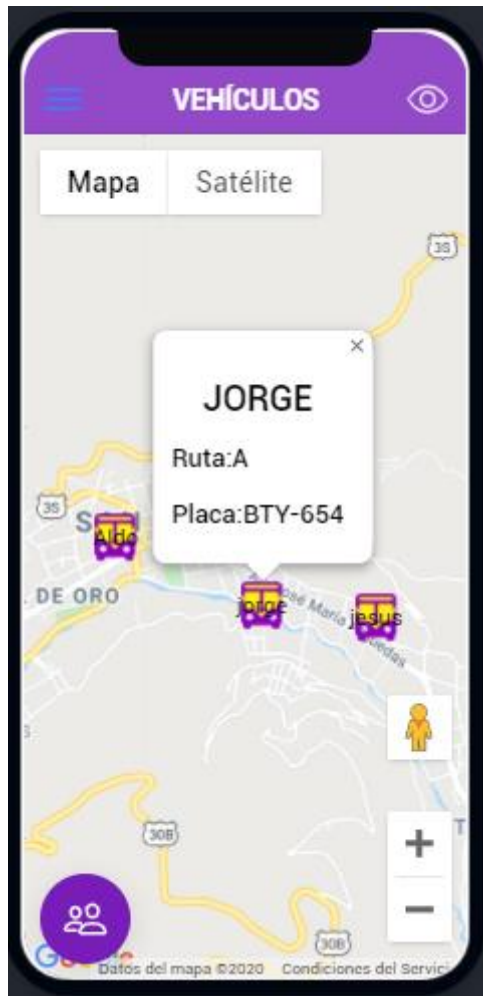
- Menú: página donde muestra todas las páginas y rutas de la aplicación.



*Figura 31: Diseño del módulo administrador menú principal versión Ios*

Fuente: Elaboración propia

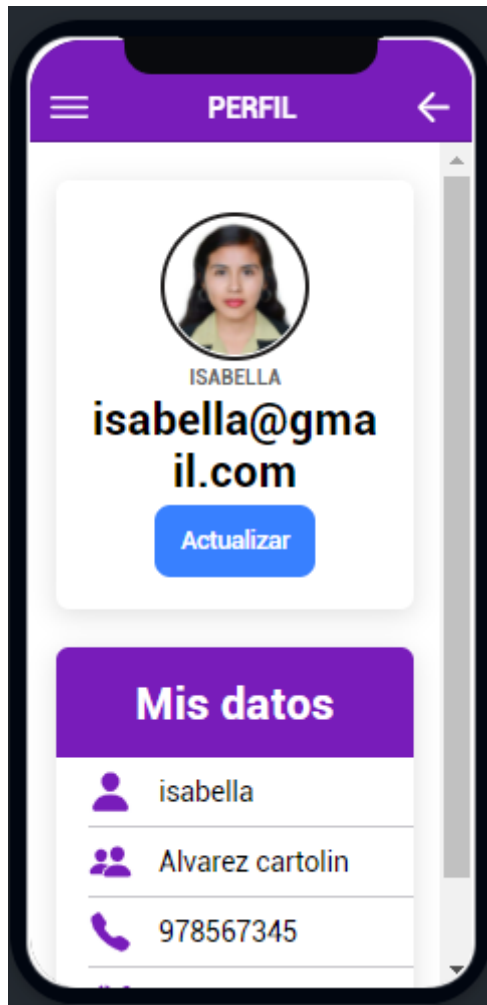
- Geolocalización: página donde se visualiza la geolocalización en tiempo real de todos los vehículos en servicio.



*Figura 32: Diseño del módulo administrador de geolocalización de los vehículos en servicio versión 1.0*

Fuente: Elaboración propia

- Perfil: Pagina donde se muestra los datos personales del administrador.



*Figura 33: Diseño del módulo administrador Perfil versión Ios*

Fuente: Elaboración propia

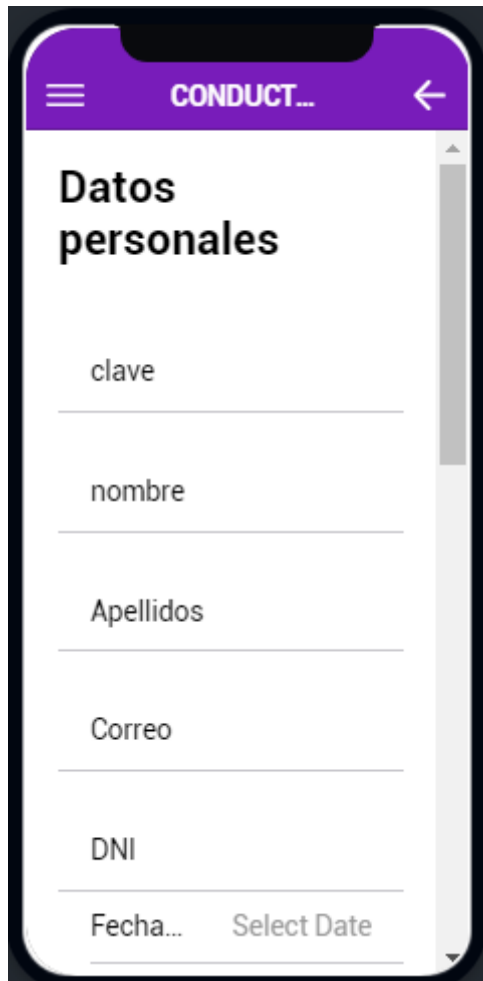
- Registro de administrador: página para el registro respectivo de un nuevo administrador.

The image shows a mobile application interface for a registration module. At the top, there is a purple header bar with a hamburger menu icon on the left, the text 'REGISTRO ...' in the center, and a back arrow icon on the right. Below the header, the form consists of six white input fields with light gray borders, each containing a label: 'Nombre', 'Apellidos', 'Dirección', 'Celular', 'Email', and 'Contraseña'. At the bottom of the form, there is a purple rounded rectangular button with the white text 'Registrar'.

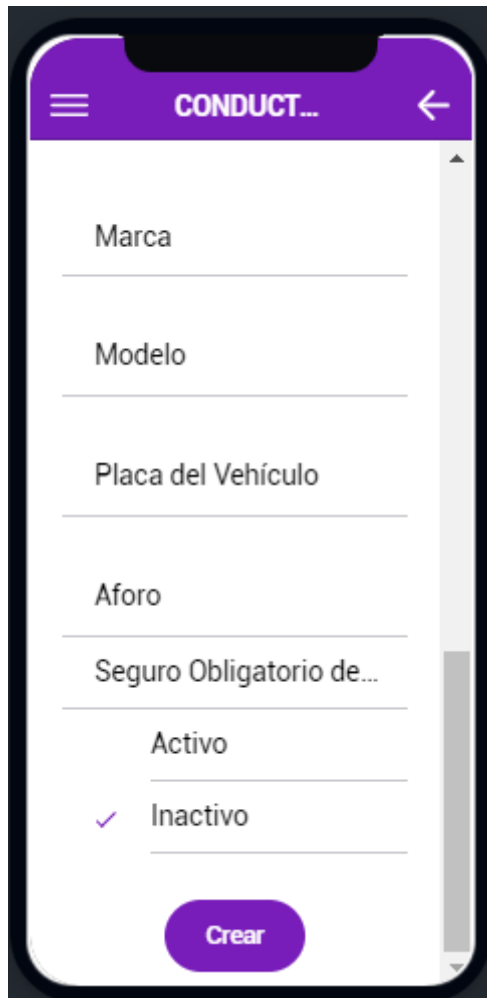
*Figura 34: Diseño del módulo administrador de registro versión los*  
Fuente: Elaboración propia

- Registro conductor: página donde se registra los conductores.



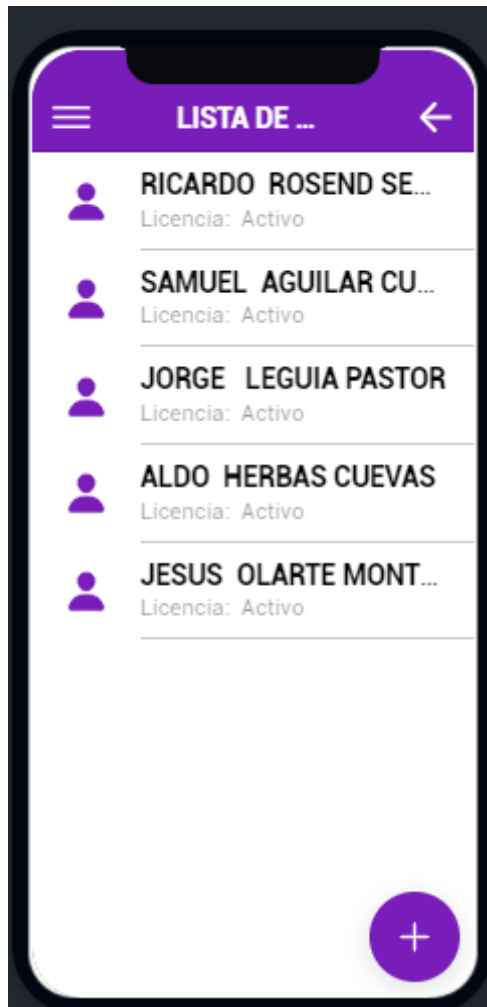


*Figura 35: Diseño del módulo administrador de registro conductor versión Ios*  
Fuente: elaboración propia



*Figura 36: Diseño del módulo administrador de registro conductor versión Ios*  
Fuente: Elaboración propia

- **Lista de conductores:** Pagina donde se muestra la lista de conductores.



*Figura 37: Diseño del módulo administrador lista de conductores versión Ios*

Fuente: Elaboración propia

- Reporte de salida y llegada: página donde se muestra el reporte de cada uno de los conductores.

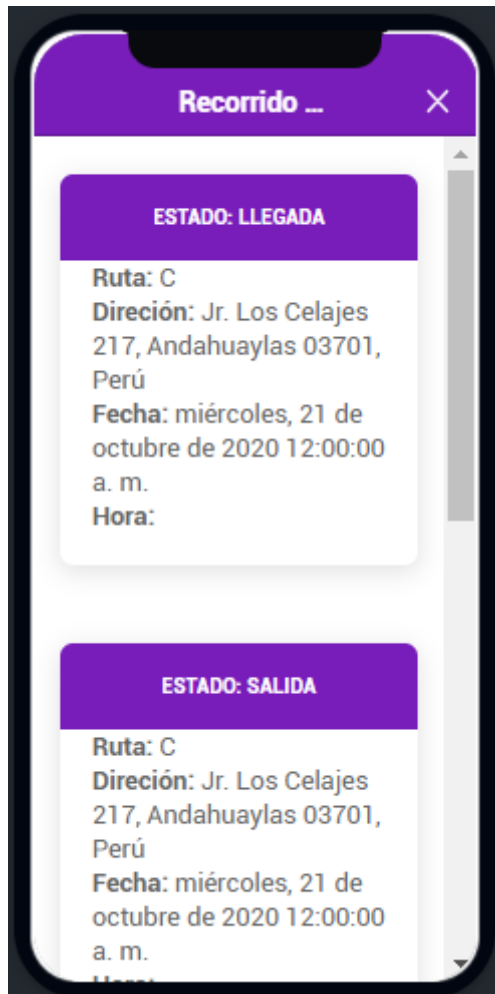


Figura 38: Diseño del módulo administrador reporte de salida y llegada versión Ios  
Fuente: Elaboración propia

- Actualizar conductor: página para actualizar al conductor.

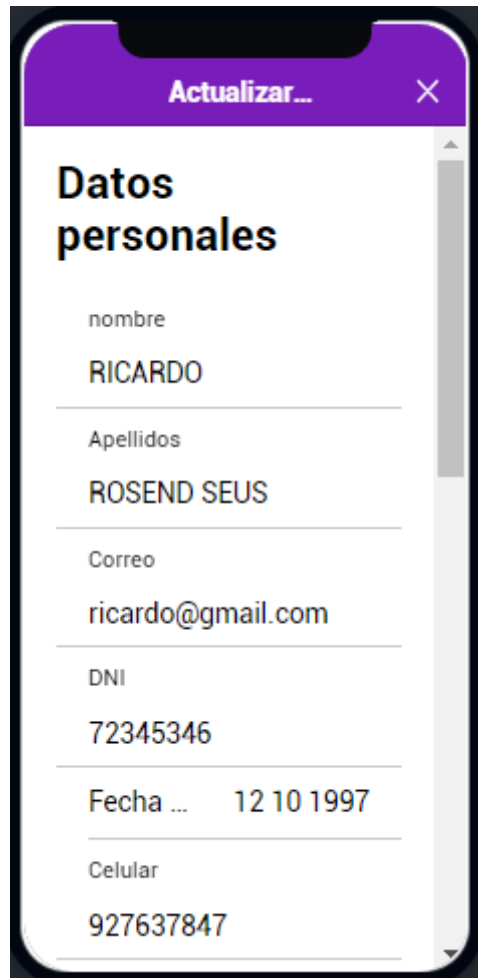
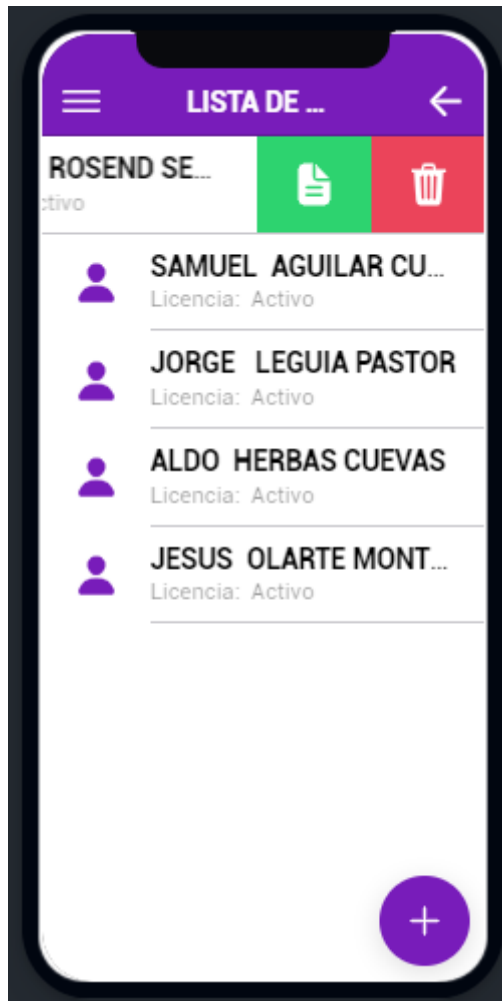


Figura 39: Diseño del módulo administrador actualizar conductor versión Ios  
Fuente: Elaboración propia

- Eliminar conductor: Pagina para eliminar a un conductor.



*Figura 40: Diseño del módulo administrador eliminar Conductor versión Ios*  
Fuente: Elaboración propia

- **MODULO CONDUCTOR**
  - Iniciar sesión con clave: Página donde el conductor inicia sesión mediante una clave, generado por el administrador.



*Figura 41: Diseño del módulo conductor iniciar sesión con clave versión Ios*  
Fuente: Elaboración propia.

- Geolocalización de ubicación actual: Página donde se observa la ubicación actual y la geolocalización en tiempo real del conductor.



Figura 42: Diseño del módulo conductor de ubicación actual versión Ios

Fuente: Elaboración propia

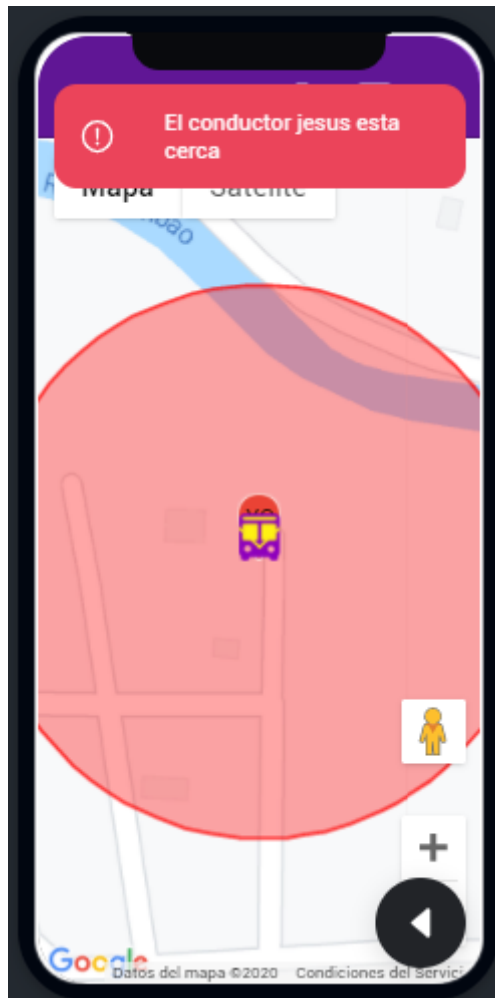
- Geolocalización de vehículos: Pagina donde se muestran a todos los vehículos en servicio.





*Figura 43: Diseño del módulo conductor geolocalización de vehículos versión Ios*  
Fuente: Elaboración propia

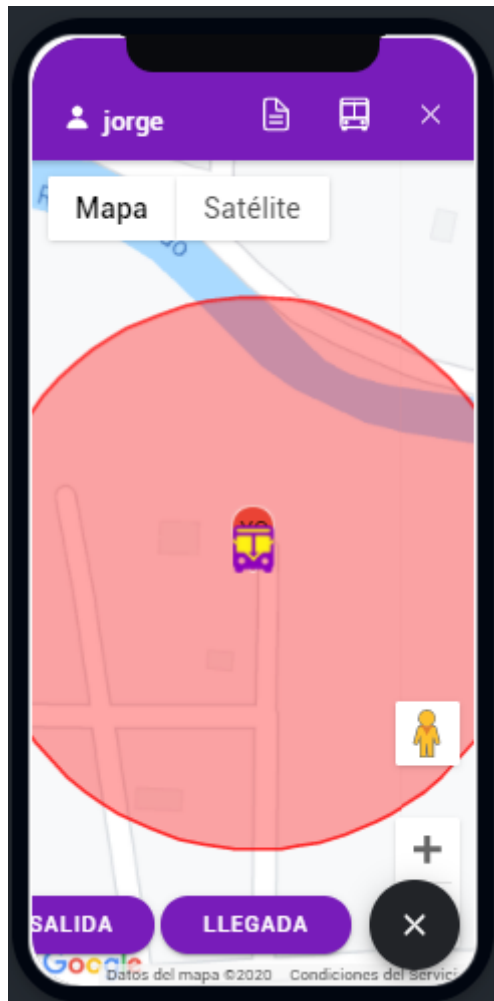
- Orden de circulación de los vehículos mediante una alerta: Pagina donde se visualiza las alertas cuando otro vehículo entra en su radio de 100 metros.



*Figura 44: Diseño del módulo conductor orden de circulación de los vehículos mediante una alerta Versión Ios*

Fuente: Elaboración propia

- Control registro de salida y llegada: página donde se reporta la salida y llegada de los vehículos en servicio a los paraderos iniciales y finales.



*Figura 45: Diseño del módulo conductor control registro de salida y llegada versión Ios*  
Fuente: Elaboración propia

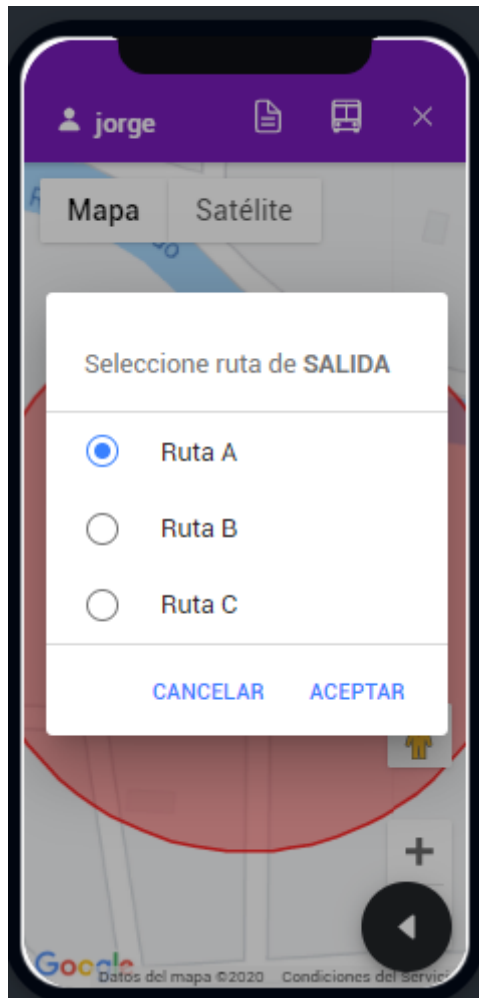


Figura 46: Diseño del módulo conductor control registro de salida y llegada versión Ios  
Fuente: Elaboración propia

- **MODULO USUARIO**

- Iniciar sesión: página donde los usuarios podrán autenticarse en la aplicación.

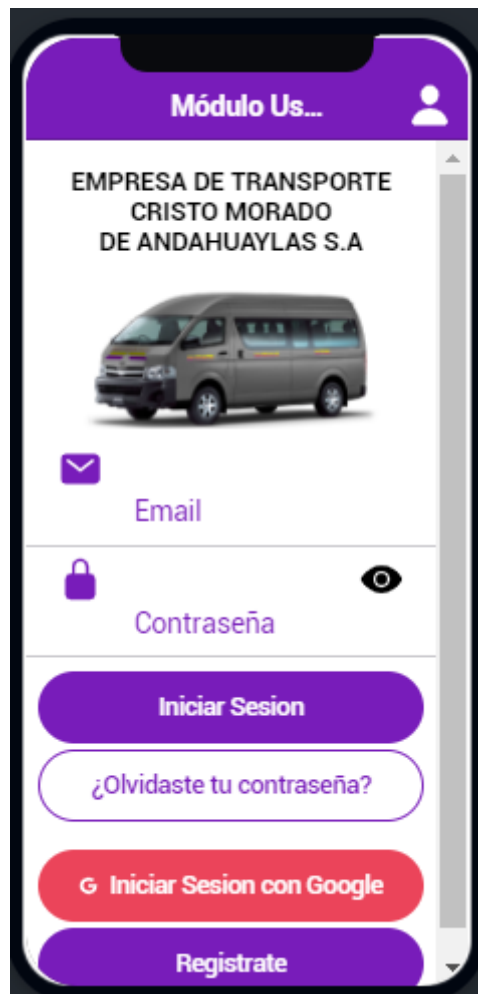


Figura 47: Diseño del módulo usuario iniciar sesión versión Ios  
Fuente: Elaboración propia

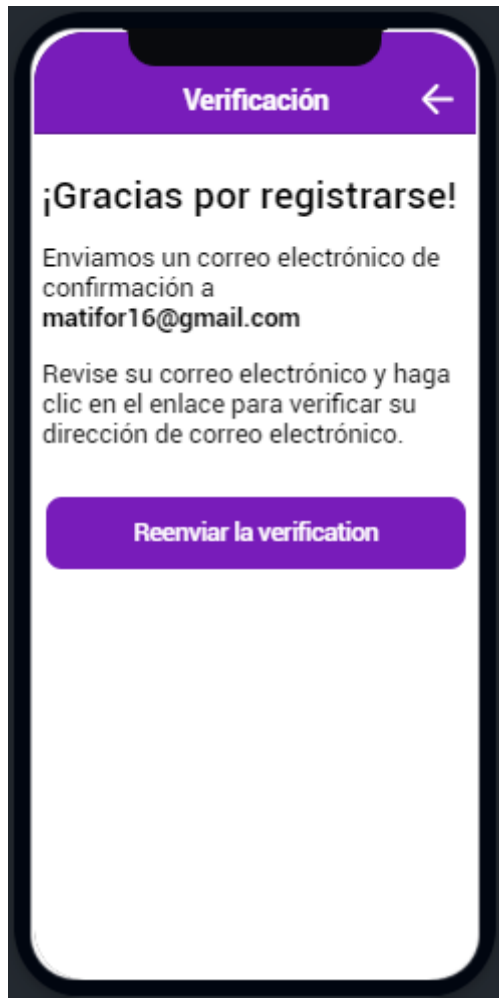
- Registro: página donde el usuario podrá registrarse en la aplicación.

The image shows a mobile application interface for a registration form. At the top, there is a purple header bar with the word "REGISTROS" in white capital letters and a white back arrow icon to its right. Below the header, the form consists of seven white input fields, each with a label in gray text: "Nombre:", "Apellidos:", "DNI:", "Dirección:", "Celular:", "Email:", and "Contraseña:". The fields are separated by thin gray horizontal lines. At the bottom of the form, there is a purple rounded rectangular button with the word "Registrar" in white text. A vertical gray scrollbar is visible on the right side of the form, indicating that the content can be scrolled.

*Figura 48: Diseño del módulo usuario registro versión Ios*

Fuente: Elaboración propia

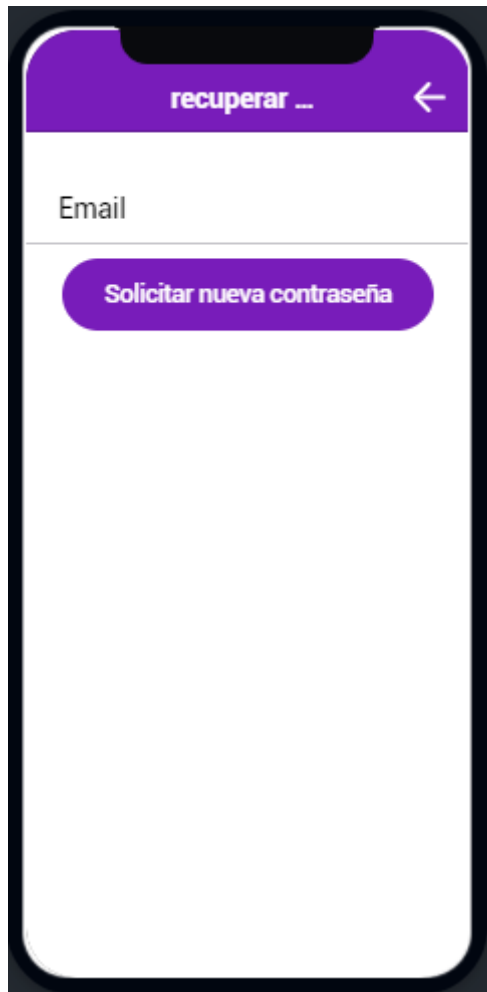
- Verificar correo: apartado donde el usuario podrá verificar su correo electrónico.



*Figura 49: Diseño del módulo usuario verificar correo versión Ios*

Fuente: Elaboración propia

- Recuperar contraseña: Pagina donde el usuario podrá recuperar su contraseña.



*Figura 50: Diseño del módulo usuario recuperar contraseña versión Ios*

Fuente: Elaboración propia

- Geolocalización de vehículos: página donde los usuarios podrán visualizar la geolocalización en tiempo real de todos los vehículos en servicio.





Figura 51: Diseño del módulo usuario geolocalización de vehículos versión Ios  
Fuente: Elaboración propia

- Perfil: Pagina donde el usuario podrá visualizar todos sus datos personales.

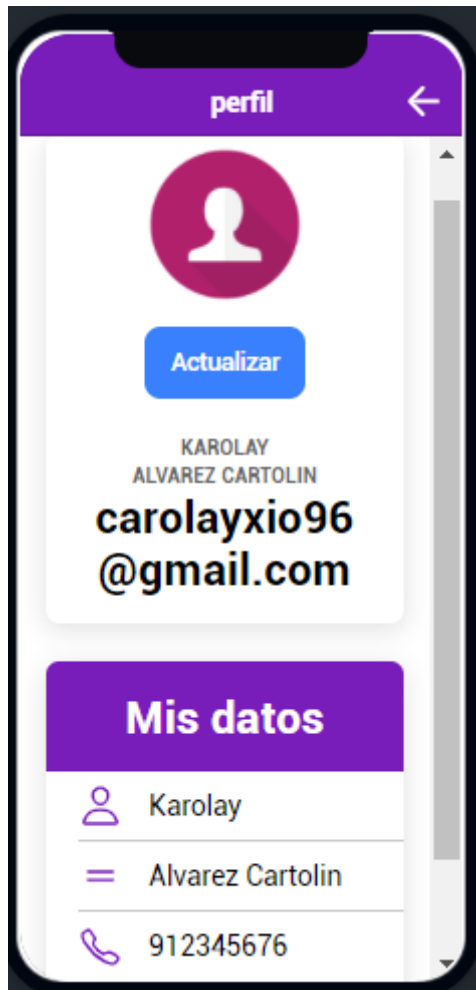


Figura 52: Diseño del módulo usuario perfil versión Ios

Fuente: Elaboración propia

### 3.5. Administración y gestión del proyecto basado en PMBOK.

La metodología para la gestión del proyecto en la presente investigación denominada DESARROLLO DE UNA APLICACIÓN MÓVIL MULTIPLATAFORMA DE GEOLOCALIZACIÓN EN TIEMPO REAL DE LA EMPRESA DE TRANSPORTE PÚBLICO “CRISTO MORADO”-ANDAHUAYLAS es PMBOK (Guía de los Fundamentos de Gestión de proyectos) sexta edición.

La metodología PMBOK establece las 10 áreas de conocimiento, con las cuales se desarrolla en este presente proyecto de investigación.



Figura 53: las 10 áreas de conocimiento de PMBOK

Fuente: Elaboración propia

### 3.5.1. Plan de gestión de la integración

la Gestión de la Integración del Proyecto engloba los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar todos los procesos de la gestión de proyectos. Los procesos de la dirección de proyectos se agrupan en cinco categorías conocidas como Grupos de Procesos de la Dirección de Proyectos (o Grupos de Procesos). (Díaz Díaz, 2020)

- Grupo de Procesos de Inicio.
- Grupo de Procesos de Planificación.
- Grupo de Procesos de Ejecución.
- Grupo de Procesos de Monitoreo y Control.
- Grupo de Procesos de Cierre.

Tabla 2: Plan de gestión de la integración

<b>PROYECTO</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de Transporte público Cristo Morado de la ciudad de Andahuaylas.
-----------------	---

<b>PATROCINADO, PREPARADO Y REVISADO POR</b>	Zulma Isabel Alvarez Cartolin - Jefe del proyecto
<b>REVISIÓN</b> (Correlativo)	<b>DESCRIPCIÓN (REALIZADO POR)</b> (Motivo de la revisión y entre paréntesis quien la realizó)
<b>01</b>	Preparación de acta de Constitución (Zulma Isabel Alvarez cartolin)
<b>BREVE DESCRIPCIÓN DEL PRODUCTO O SERVICIO DEL PROYECTO</b> (Características, funcionalidades, soporte entre otros)	
<p>La presente investigación tiene como objetivo el desarrollo y la implementación de un aplicativo móvil, haciendo uso de las herramientas ionic, google maps, firebase las cuales nos permitirá desarrollar una aplicación intuitiva al usar la aplicación, permitiendo a los conductores de la empresa de transporte público urbano Cristo Morado a tener más orden en la circulación de los vehículos por sus respectivas rutas mediante una alerta de todos los vehículos que están cerca de su ubicación en 100 metros de radio, como también el reporte de la entrada y salida en los paraderos iniciales y finales, igualmente los usuarios podrán visualizar en tiempo real la ubicación exacta de los vehículos que están en servicio con sus respectivos rutas.</p>	
<b>ALINEAMIENTO DEL PROYECTO</b>	
<b>1. OBJETIVOS ESTRATEGICOS DE LA ORGANIZACIÓN</b>	<b>2. PROPÓSITO DEL PROYECTO</b>
<p>1.1. Desarrollo de un módulo para el usuario que muestre en tiempo real a todos los vehículos en servicio.</p> <p>1.2. Desarrollo de un módulo para el conductor que le permita guardar sus coordenadas para que posteriormente los usuarios y administrador pueda visualizar en tiempo real a los vehículos en servicio, igualmente el orden de los vehículos con una alerta</p>	<p>1.1. Impulsar la creación de aplicaciones con el uso de framework y aplicarlos en la mejora de los las mismas.</p> <p>1.2. Promover la investigación del desarrollo de aplicaciones haciendo uso de hardware y software libre para automatizar procesos.</p>

<p>cuando un segundo conductor se aproxime a su posición.</p> <p>1.3. Desarrollo de un módulo para el administrador que permita crear conductores, borrar conductores, agregar administradores, visualizar el reporte de entrada y salida de los vehículos en servicio y finalmente visualizar la ubicación de todos los vehículos en tiempo real.</p>	
<p style="text-align: center;"><b>3. OBJETIVOS DEL PROYECTO</b> (Principalmente en términos de costo, tiempo alcance, calidad)</p>	
<p>3.1. Desarrollar una aplicación móvil multiplataforma</p> <p>3.2. Lograr que los vehículos se ordenen mediante una alerta cuando otros vehículos estén cerca de su radio de 100 metros.</p> <p>3.3. Lograr que los conductores realicen su reporte de entrada y salida en los paraderos iniciales y finales.</p> <p>3.4. Realizar taller de capacitación a los conductores de la empresa de transporte urbano cristo morado en la ciudad de Andahuaylas.</p> <p>3.5. Finalizar el proyecto en un plazo de 7 meses a partir de la fecha del acta de constitución del proyecto.</p>	
<p style="text-align: center;"><b>4. FACTORES CRÍTICOS DE ÉXITO DEL PROYECTO</b> (Componentes o características que deben cumplirse en el proyecto para considerar exitoso)</p>	
<p>4.1. Proceso de adquisición del software dentro de las fechas establecidas con los módulos y características especificados</p> <p>4.2. Diseño del software conforme a los requerimientos establecidos por parte de los interesados del proyecto.</p> <p>4.3. Envía alerta de notificación a todos los vehículos que están cerca de su ubicación en 100 metros de radio.</p>	

4.4. Realiza el reporte de salida y llegada en los paraderos iniciales y finales.	
<b>5. REQUERIMIENTOS DE ALTO NIVEL</b> (Principales condiciones y/o capacidades que debe cumplir el producto o servicio y la gestión del proyecto)	
5.1. Investigación de todas las herramientas que serán usadas en el proyecto.	
5.2. Diseñar del módulo de la ubicación en tiempo real de todos los vehículos en servicio.	
5.3. Implementación del módulo de alerta con el propósito de ordenar a todos los vehículos en tiempo real	
5.4. Realizar las pruebas de funcionamiento de la aplicación Ubimobile..	
<b>EXTENSIÓN Y ALCANCE DEL PROYECTO</b>	
<b>6. FASES DEL PROYECTO</b> (Agrupamiento lógico de actividades relacionadas que usualmente culminan elaborando un entregable principal).	<b>7. PRINCIPALES ENTREGABLES</b> (Un único y verificable producto, resultado o capacidad de realizar un servicio que debe ser elaborado para contemplar un proceso, una fase o un proyecto).
<b>Fase I- Gestión del Proyecto</b>	Acta de Constitución del Proyecto.
<b>Fase II-Diseño</b>	Proceso de definición de las interfaces, y de la arquitectura
<b>Fase III – Requerimiento de Adquisiciones</b>	Adquisición de una laptop, smartphones y otros componentes. Adquisición del entorno de desarrollo(Ionic, Firebase, API de google maps, Cordova, Angular, etc)
<b>Fase IV – Implementación</b>	Implementar los 3 módulos de la aplicación: <ul style="list-style-type: none"> <li>• Módulo Administrador.</li> <li>• Módulo Conductor.</li> <li>• Módulo Usuario</li> </ul>

<b>Fase V – Arranque del Sistema</b>	Pruebas de la aplicación.
<b>Fase VI – Término de Obra</b>	Aplicación Móvil multiplataforma
<b>8. INTERESADOS CLAVE</b> (Persona u organización que ésta activamente involucrado en el proyecto o cuyos intereses pueden ser afectados positivamente o negativamente por la ejecución del proyecto o por el producto que elabora).	
8.1.	Usuarios de las ciudades Talavera de la Reina, Andahuaylas, San Jerónimo.
8.2.	Empresa de transporte Público Urbano Cristo Morado de la ciudad de Andahuaylas.
<b>9. RIESGOS</b> (Eventos o condiciones inciertas que, si ocurriese, tiene un efecto positivo o negativo sobre los objetivos del proyecto)	
9.1.	Actualizaciones de versiones en el framework usados en el proyecto.
9.2.	Restricciones de uso limitado en la plataforma firebase
9.3.	Falta de energía eléctrica y de cobertura de internet
<b>10. HITOS PRINCIPALES DEL PROYECTO</b> (Un evento significativo para el proyecto)	
10.1.	Aprobación del acta de constitución del proyecto
10.2.	Alcanzar todos los objetivos trazados.
<b>11. PRESUPUESTO DEL PROYECTO</b>	
El costo de la ejecución del proyecto será auto financiado.	

Fuente: Elaboración propia

### 3.5.2. Plan de Gestión de alcance

Incluye los procesos requeridos para garantizar que el proyecto incluye todo el trabajo requerido y únicamente el trabajo requerido para completarlo con éxito el producto final.

Tabla 3: Plan de gestión de alcance

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>1. Describir cómo será administrado el alcance del Proyecto</b>	
En primera instancia se establece a quienes estará dirigido el proyecto, teniendo en cuenta ese punto, se determina que la aplicación móvil será usada por los conductores de la empresa de transporte Cristo Morado y los usuarios de hacen uso de la aplicación Ubimobile.	
<b>2. Evaluar la estabilidad del alcance del proyecto</b> (cómo manejar los cambios, la frecuencia e impacto de los mismos)	
Toda actualización o alteración a los cambios del proyecto deben ser evaluados, aprobados y optimizados. El jefe de proyectos debe calificar el impacto y alertar alternativas de solución. Los cambios requeridos serán revisados en las reuniones quincenales, se aprueba la solución de la misma en reuniones posteriores a la que fue requerido.	
<b>3. ¿Cómo los cambios al alcance, serán identificados y clasificados?:</b>	
El responsable del proyecto, notificara los cambios que se requiera en el alcance quien realizara el análisis de mismo, del mismo modo si se necesita información adicional el podrá requerir del solicitante	
<b>4. Descripción cómo los cambios del alcance serán integrados al proyecto:</b>	
Si la importancia del proyecto no cambia la línea base del desarrollo del proyecto será aprobado por el responsable del proyecto, y se notificará nueva actualización de las bases de todos los planes del proyecto.	
<b>5. Comentarios adicionales:</b>	
La aplicación móvil multiplataforma tendrá nuevas actualizaciones escalables en el tiempo.	

Fuente: Elaboración propia

### 3.5.3. Plan de gestión del cronograma

Incluye los procesos involucrados en planificar, estimar, presupuestar, financiar, obtener financiamiento, gestionar y controlar los costos de modo que se complete el



proyecto dentro del presupuesto aprobado.

Tabla 4: plan de gestión de cronograma

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.	
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto	
<b>Persona(s) autorizada(s) a solicitar cambio en cronograma:</b>		
<b>Nombre</b>	<b>Cargo</b>	<b>Ubicación</b>
Zulma Isabel Alvarez Cartolin	Jefe del proyecto	Andahuaylas, Talavera, San Jerónimo
<b>Persona(s) que aprueba(n) el requerimiento de cambio de cronograma:</b>		
<b>Nombre</b>	<b>Cargo</b>	<b>Ubicación</b>
Zulma Isabel Alvarez Cartolin	Jefe del Proyecto	Andahuaylas, Talavera, San Jerónimo
<b>Razones aceptables para cambios en cronograma del proyecto:</b> (por ejemplo, retrasos debido a entrega de materiales o disponibilidad de personal, clima, adelantar el cumplimiento debido a término de fase o proceso, etc.)		
<ul style="list-style-type: none"> <li>-solicitud de cambio de parte de la empresa de transporte Cristo Morado y de los usuarios.</li> <li>- Mal establecimiento de la secuencia de actividades. Fallos o averías del servicio de internet.</li> <li>- Fallos de los equipos de trabajo (laptop, Smartphone y otros accesorios).</li> </ul>		
<b>Describir cómo calcular y reportar el impacto en el proyecto por el cambio en cronograma:</b> (Tiempo, costo, calidad. Etc.):		
<ul style="list-style-type: none"> <li>- El cambio se notificará al jefe de proyecto en fechas adecuadas.</li> <li>- Descripción del problema especificando la gravedad de una urgencia.</li> <li>- Impacto del mismo sobre el proyecto (Costo, calidad, tiempo y alcance).</li> <li>- Describir las alternativas de solución detallando el impacto en las diferentes áreas (costo, calidad, tiempo y alcance).</li> </ul>		
<b>Describir cómo los cambios al cronograma serán administrados:</b>		
<b>La administración del alcance se realizará de la siguiente manera:</b>		
<b>1) Designación de Responsabilidades:</b>		

<ul style="list-style-type: none"> <li>• Planificación (Jefe del proyecto).</li> <li>• Ejecución (Jefe del proyecto).</li> <li>• Desarrollo (Jefe del proyecto).</li> <li>• Seguimiento y control (Jefe del proyecto).</li> <li>• Pruebas de la aplicación (Jefe del proyecto)</li> </ul>
---

Fuente: Elaboración propia

### 3.5.4. Plan de gestión de los costos

Tabla 5: Plan de gestión de costos

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.	
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto	
<b>Persona(s) autorizada(s) a solicitar cambio en cronograma:</b>		
<b>Nombre</b>	<b>Cargo</b>	<b>Ubicación</b>
Zulma Isabel Alvarez Cartolin	Jefe del proyecto	Andahuaylas, Talavera, San Jerónimo
<b>Persona(s) que aprueba(n) el requerimiento de cambio de cronograma:</b>		
Nombre	Cargo	Ubicación
Zulma Isabel Alvarez Cartolin	Jefe del Proyecto	Andahuaylas, Talavera, San Jerónimo
<b>Persona(s) que aprueba(n) el requerimiento de cambio de costo interno ofrecido:</b>		
-Jefe del proyecto		
<b>Razones aceptables para cambios en el costo del proyecto:</b> (por ejemplo: Aprobación de cambios en el alcance, incremento de costo en los materiales, etc.):		
<b>Describir cómo calcular y reportar el impacto en el proyecto por el cambio en costo:</b> (Tiempo, costo, calidad. Etc.):		

Fuente: Elaboración propia

### 3.5.5. Plan de Gestión de la calidad

Tabla 6: plan de gestión de la calidad

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>GESTIÓN DE CALIDAD DEL PROYECTO</b> (Descripción de cómo se van a aplicar los procesos de gestión de calidad del proyecto. Herramientas a emplear, normativas, reglamentos, responsables, áreas de aplicación, etc.)	
<b>PLANIFICAR LA CALIDAD:</b> (Explicar y sustentar cómo se ha elaborado el presente plan de gestión de calidad)	
<b>REALIZAR ASEGURAMIENTO DE CALIDAD</b> (Explica cómo se va a realizar el proceso de aseguramiento de calidad )	

Fuente: Elaboración propia

### 3.5.6. Plan de Gestión de los recursos

Tabla 7: Plan de gestión de los recursos

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>ROL EN EL PROYECTO</b> (Denominación del proyecto)	Jefe del proyecto
<b>FUNCIÓN / RESPONSABILIDAD PRINCIPAL</b> (Descripción de los que se espera de este rol)	

<b>COMPETENCIAS REQUERIDAS / RESPONSABILIDADES</b> (Conocimiento, habilidades, actitudes)				
<b>DISPONIBILIDAD</b> (Posibilidad de contar con el recurso humano en el tiempo requerido)				
<b>Descripción del puesto a requerir</b>	<b>Externo interno</b>	<b>Fecha ingreso</b>	<b>Fecha retiro</b>	<b>Tiempo requerido</b>
<b>INCORPORACIÓN AL PROYECTO</b> (Cómo se incorpora el personal al proyecto)				
<b>Evento disipador</b> (Hecho que identifica la necesidad de haber gestiones sobre el personal)		<b>Sincronización previa</b> (Acciones requeridas para gestionar al personal)		
<b>Renuncia</b>				
<b>Enfermedad</b>				

Fuete: Elaboración propia

### 3.5.7. Plan Gestión de las Comunicaciones

Tabla 8: Plan Gestión de las Comunicaciones

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>GESTIÓN DE COMUNICACIONES DEL PROYECTO</b> (Descripción de cómo se va a aplicar los procesos de gestión de comunicaciones del proyecto, herramientas a emplear, responsables, áreas de aplicaciones, etc.)	
<b>PLANIFICACIÓN DE LA COMUNICACIÓN</b> (Explicar y sustentar cómo se ha elaborado el presente plan de gestión de comunicación)	

Fuente: Elaboración propia

### 3.5.8. Plan de Gestión de riesgo

Tabla 9: Plan de gestión de riesgo

<b>NOMBRE DEL PROYECTO:</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PREPARADOR POR:</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>Descripción de la metodología de gestión del riesgo a ser usada:</b>	
<b>Alcances</b>	
El jefe del proyecto realizará la identificación, priorización y seguimiento de los riesgos más críticos en el proyecto.	
<b>Herramientas</b>	
<ul style="list-style-type: none"> <li>• Ideas</li> <li>• Opinión de la alta gerencia de la empresa de transportes</li> <li>• feedback del usuario</li> </ul>	
<b>Fuentes de datos</b>	
La identificación de los riesgos será por parte de todos los integrantes del proyecto según experiencia y juicio de especialistas	
<b>Roles y Responsabilidades:</b>	
<p>- <b>Jefe del proyecto:</b> Es el responsable de la identificación, seguimiento y control en la ejecución del proyecto.</p> <p>- <b>Programador:</b> encargado de desarrollar la aplicación, e integrar las herramientas.</p> <p>- <b>Diseñador:</b> encargado del diseño del sistema.</p>	

Fuente: Elaboración propia

### 3.5.9. Plan de gestión de las adquisiciones

Tabla 10: plan de gestión de las adquisiciones

<b>NOMBRE DEL PROYECTO</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en
----------------------------	--

	tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PLAN DE GESTIÓN DE ADQUISICIONES</b>	
<b>Herramientas</b>	<b>Características</b>
<b>Computadora.</b>	Intel Core I5, disco duro de 1024 Gb memoria interna RAM de 4Gb.
<b>Smartphone</b>	Android versión 10 o iPhone Con la finalidad de usar la geolocalización.
<b>Sistema operativo Windows con licencia</b>	Windows 10 a más, preferentemente de arquitectura de 64Bits. Es importante que cuente con software original para su correcto funcionamiento y así evitar deficiencias.
<b>StarUML</b>	Herramienta de código libre usado para el modelado de software (diagramas) basado en los estándares UML.
<b>Firebase</b>	Base de datos en tiempo real que sincroniza con los usuarios sin tener la necesidad de actualizar constantemente.
<b>Framework Ionic</b>	Versión 5.0.0, usada para el desarrollo de aplicaciones híbridas y multiplataforma.
<b>API de google maps</b>	La clave API de Google Maps se utiliza para acceder a la API de Google Maps. La aplicación móvil utiliza las capacidades de Google Maps por lo cual requiere obtener y agregar una clave.

Fuente: Elaboración propia

### 3.5.10. Plan de gestión de los interesados

Tabla 11: Plan de gestión de los interesados

<b>PROYECTO</b>	Desarrollo de una aplicación móvil multiplataforma de geolocalización en
-----------------	--

	tiempo real de la empresa de transporte público “Cristo Morado”-Andahuaylas.
<b>PREPARADO POR</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>REVISADO POR</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto
<b>REVISADO POR</b>	Zulma Isabel Alvarez Cartolin – Jefe del Proyecto

Fuente: Elaboración propia

### 3.6. Metodología de desarrollo

Para el desarrollo de este proyecto se utilizó la metodología Mobile-D, ésta metodología consta de cinco fases:



Figura 54: Metodologías de mobile-D

Fuente: Elaboración propia

#### 3.6.1. Fase de Exploración

En éste apartado del proyecto se determinará el Plan del Proyecto, el nombre que tiene la aplicación móvil, identificación de conductores y usuarios del proyecto, objetivo general y la plataforma en el cual se desarrollará el proyecto.

#### **3.6.1.1. Establecimiento de actores**

Se deberá identificar a todos los actores en influyen en la aplicación, miembros del equipo que participarán de forma activa en el tiempo que tome desarrollar el proyecto y la aplicación móvil.

- **Jefe del proyecto:** Encargado y responsable de la viabilidad, disponibilidad de todos los recursos que sean necesarios para la ejecución del proyecto.
- **Analista de Sistemas:** Encargado y responsable del diseño de la infraestructura, interfaces y aspectos visual de la aplicación móvil. Para ello deberá considerarse los requisitos funcionales, el alcance y limitaciones que la aplicación móvil tiene.
- **Programador de Sistemas:** Encargado y responsable en el desarrollo e implantación de los módulos que son diseñados por el Analista de Sistemas. De la misma forma será el encargado de realizar los test de prueba de la aplicación móvil a la hora de finalizar el desarrollo del proyecto.

Este proyecto fue desarrollado por una sola persona en este caso (Tesis) cumpliendo con las fechas establecidas dentro del plan.

#### **3.6.1.2. Estableciendo usuarios**

Se identifica a los diferentes tipos de usuario que hará el uso de la aplicación

- **Usuarios Directos:** Conductores de los vehículos de la empresa de transporte público cristo morado y usuarios de la provincia de Andahuaylas que hacen uso del servicio de transporte público urbano
- **Usuario Indirectos:** Desarrolladores que usen Frameworks, específicamente (IONIC) que se está usando en este proyecto.



### 3.6.1.3. Información clave del proyecto

Tabla 12: Información clave del proyecto

NOMBRE DE LA APLICACIÓN	Desarrollo De Una Aplicación Móvil Multiplataforma De Geolocalización En Tiempo Real De La Empresa De Transporte Público “Cristo Morado”- Andahuaylas.
OBJETIVO GENERAL	Desarrollar una aplicación móvil multiplataforma de geolocalización en tiempo real de los vehículos de la empresa de transporte “Cristo Morado” en la ciudad de Andahuaylas.
PLATAFORMA	Android e Ios

Fuente: Elaboración propia.

### 3.6.1.4. Principales Requerimientos Funcionales

- La estructura de la aplicación móvil será escalable.
- La aplicación móvil será utilizada solo por usuarios y conductores autenticados verificados.
- La aplicación móvil estará disponible en las plataformas de Android, iOS
- La aplicación móvil mostrará a los vehículos circulando en tiempo de real de la ruta de Talavera, Andahuaylas y San Jerónimo.

### 3.6.1.5. Principales Requerimientos No Funcionales

- El framework a utilizar para el desarrollo de esta aplicación será IONIC 5, Firebase Google Maps y plugins de Cordova.
- La aplicación móvil deberá ser intuitiva para su uso.
- La aplicación móvil tendrá interfaces gráficas bien estructuradas.
- La aplicación móvil tendrá un diseño responsive.
- La aplicación móvil no debe ocupar más de 20MB en el dispositivo móvil instalado.

### **3.6.2. Fase de iniciación**

Las herramientas de desarrollo, hardware para hacer las pruebas y diagramas de caso de uso.

#### **3.6.2.1. Materiales de Desarrollo**

- IONIC 5
- AngularJS
- Visual Studio Code
- Apache Cordova
- Google Maps Api
- Firebase (NoSQL)
- Plugin de Geolocalización
- Android Estudios
- Java (jdk)

#### **3.6.2.2. Hardware**

- Toshiba Intel(R) Core TM i3
- Toshiba Intel(R) Core TM i5

#### **3.6.2.3. Herramientas de Prueba**

- Alcatel One Touth
- Samsung JS 10

### 3.6.2.4. Diagrama de casos de uso de la aplicación Móvil

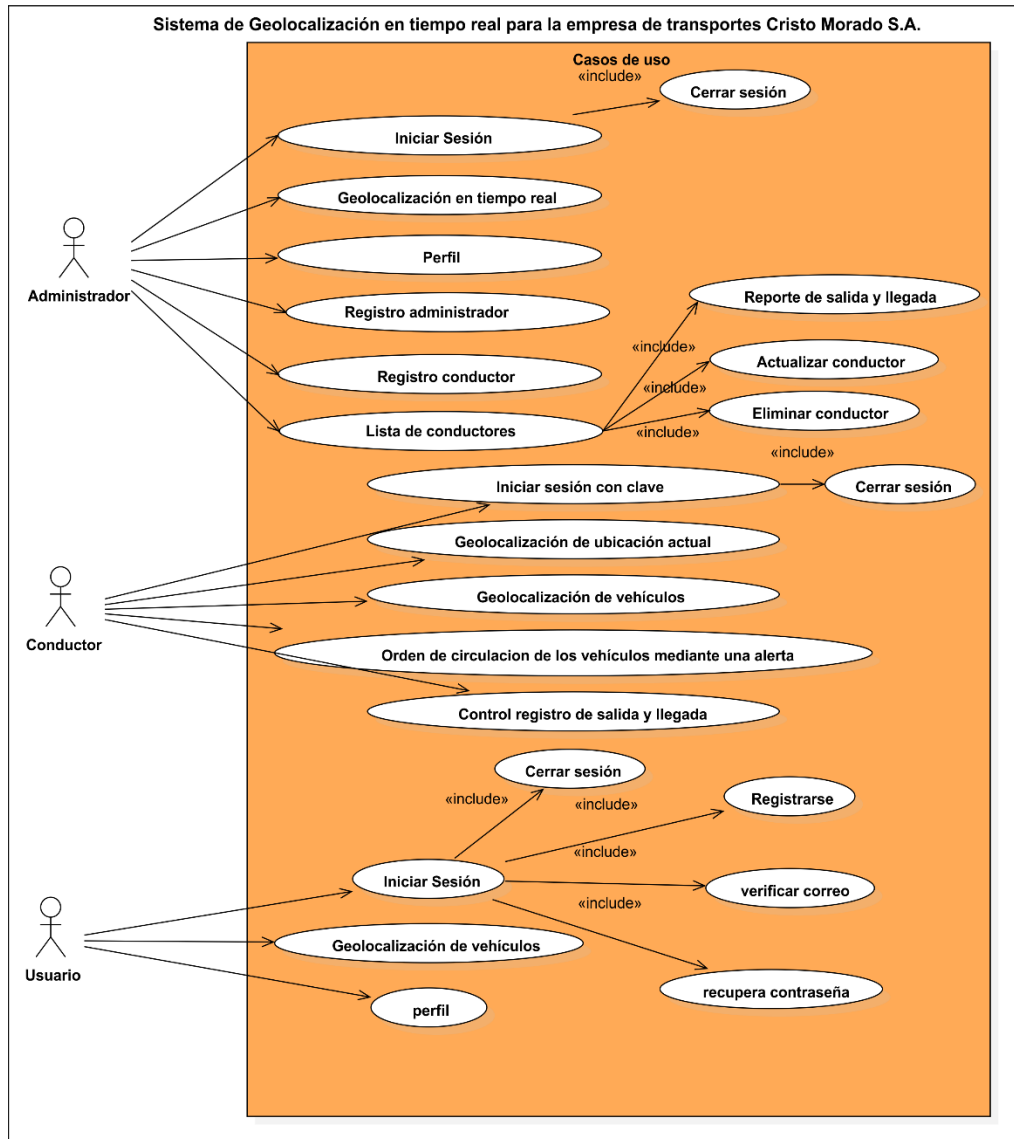


Figura 55: Diagrama de casos de uso

Fuente: Elaboración propia

#### Diagramas modulo Administrador

- Diagrama de actividad crear conductor

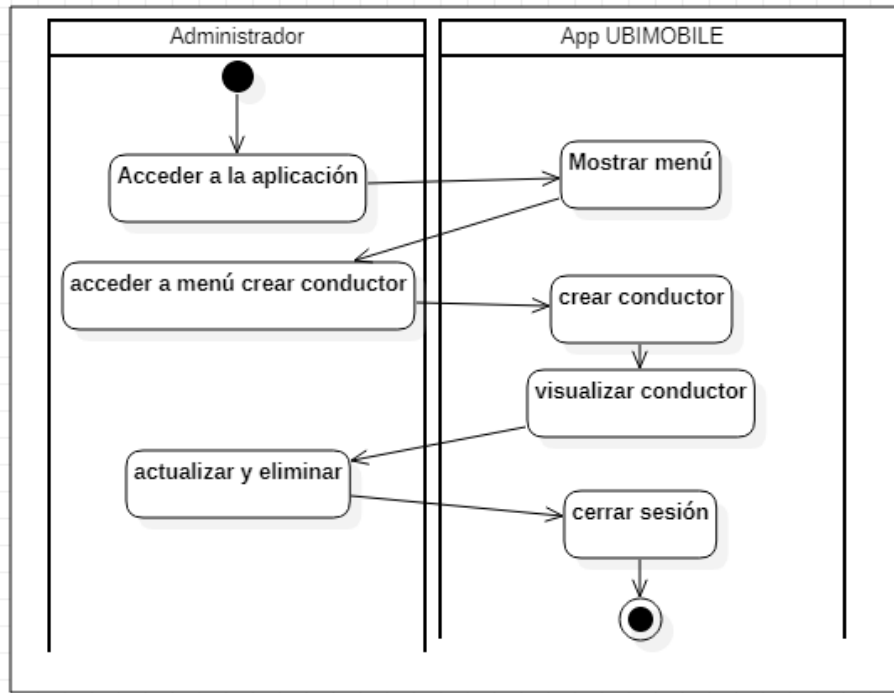


Figura 56: Diagrama de actividad crear conductor

Fuente: Elaboración propia

- Diagrama de actividad visualizar vehículos

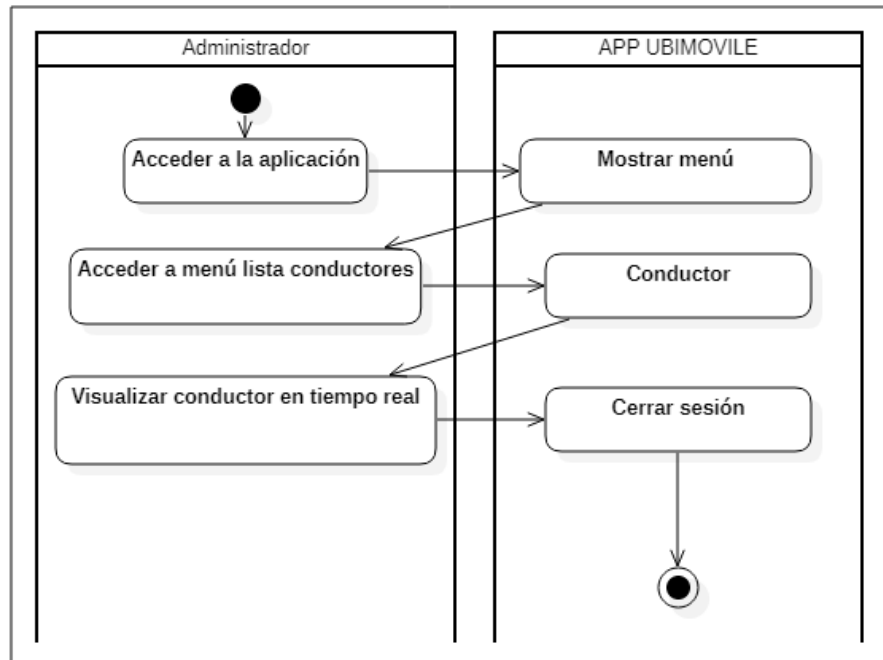


Figura 57: Diagrama de actividad visualizar vehículos

Fuente: Elaboración propia

- Diagrama de actividad mostrar reporte de salida y llegada de vehículos

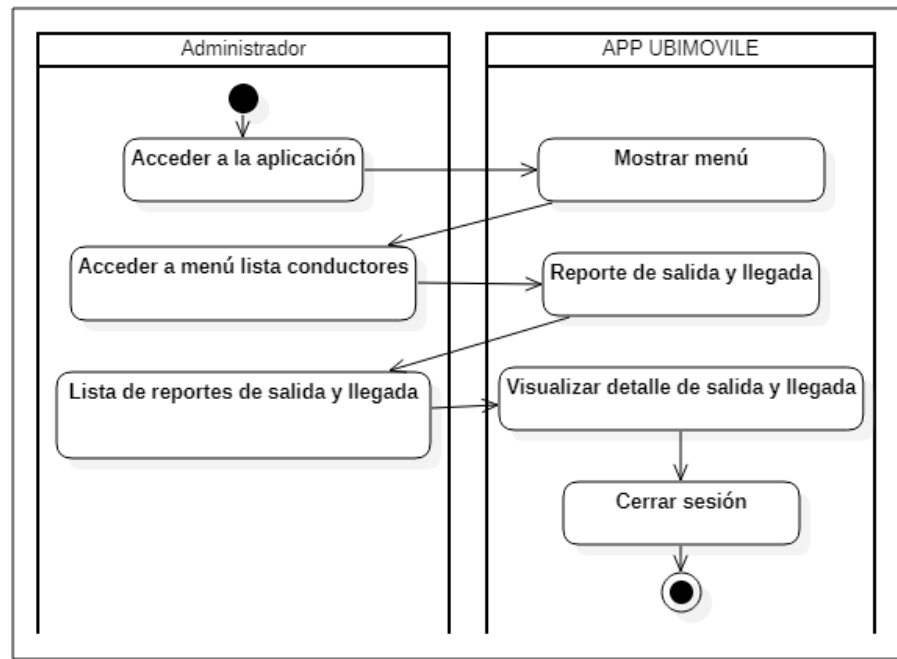


Figura 58: Diagrama de actividad mostrar reporte de salida y llegada de vehículos

Fuente: Elaboración propia

### Diagrama modulo Conductor

- Diagrama de actividad orden de los vehículos mediante alertas

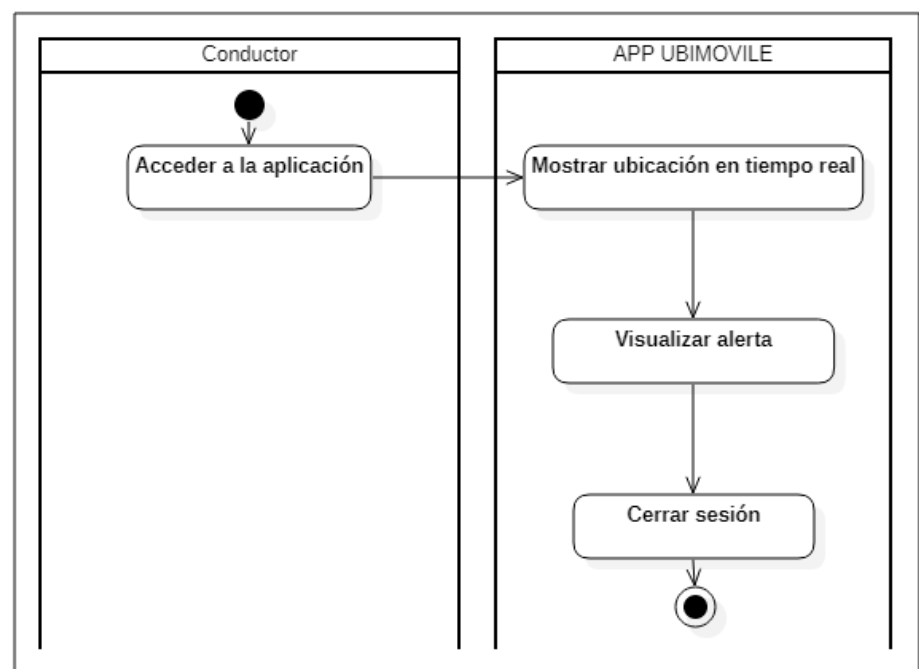


Figura 59: Diagrama de actividad orden de los vehículos mediante alertas

Fuente: Elaboración propia

- Diagrama de actividad muestras vehículo

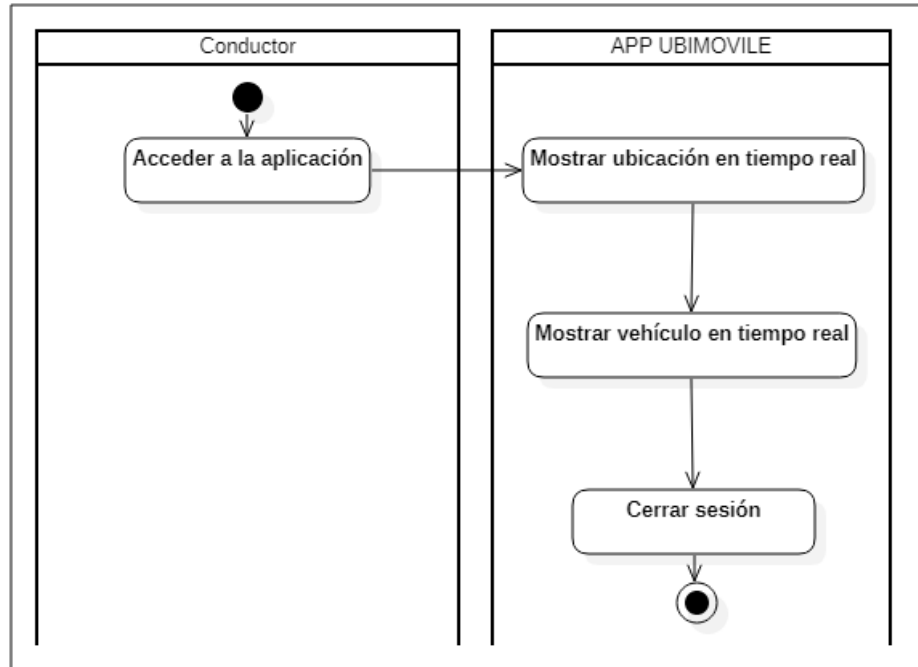


Figura 60: Diagrama de actividad muestras vehículo

Fuente: Elaboración propia

- Diagrama de actividad reporte de salida y llega en los paraderos iniciales y finales

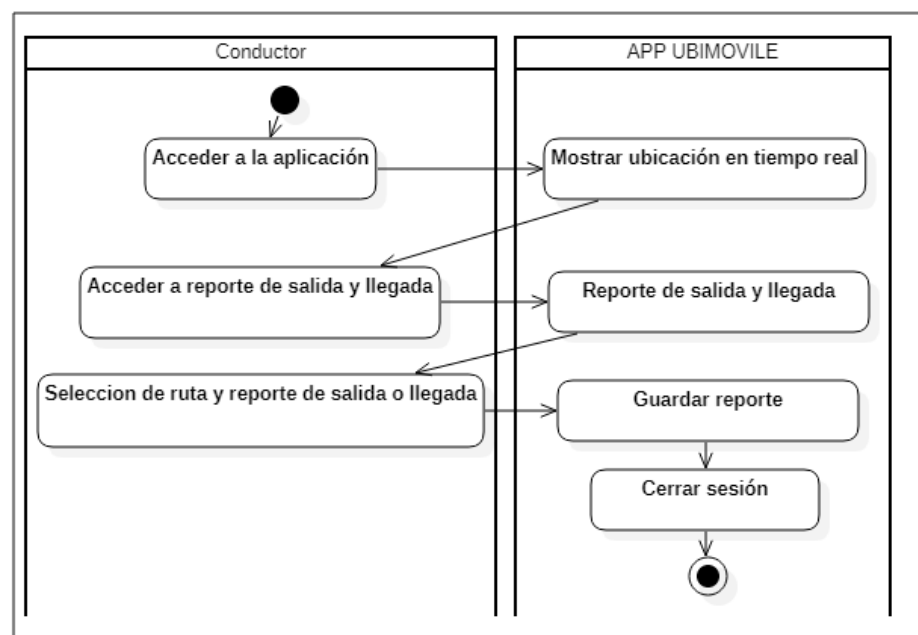


Figura 61: Diagrama de actividad reporte de salida y llega en los paraderos iniciales y finales

Fuente: Elaboración propia

- Diagrama de actividad mostrar vehículos

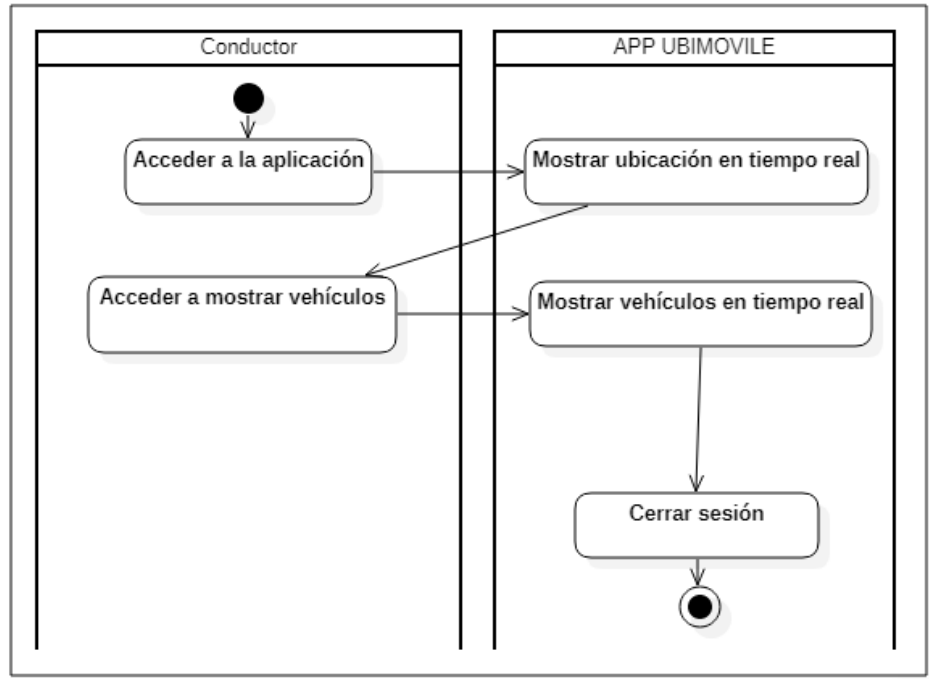


Figura 62: Diagrama de actividad mostrar vehículos

Fuente: Elaboración propia

## Diagrama modulo Usuario

- Diagrama de actividad mostrar vehiculo

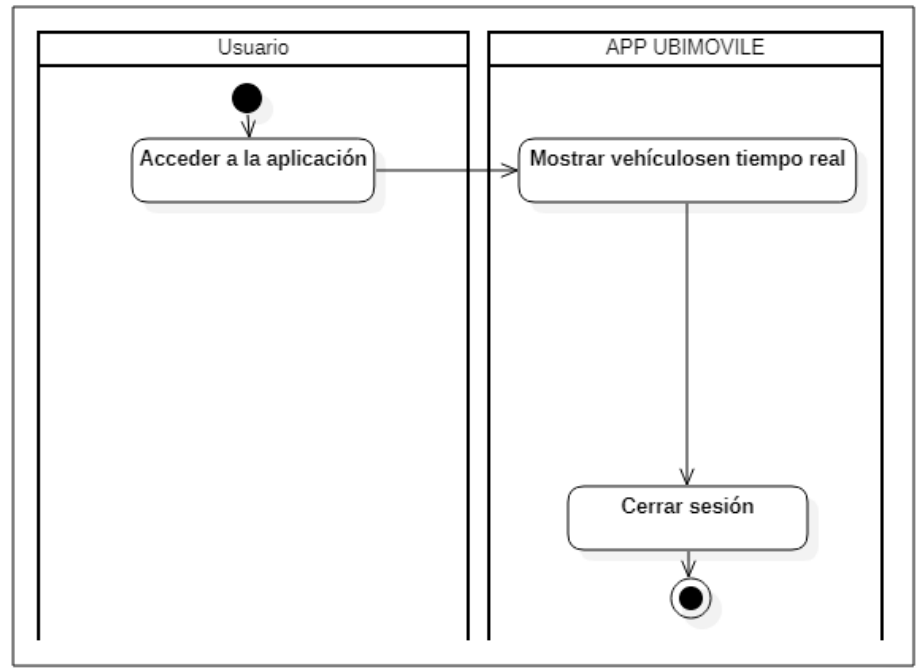


Figura 63: Diagrama de actividad mostrar vehiculo

Fuente: Elaboración propia

## MODULO ADMINISTRADOR

- Vista Login: página para la autenticación del administrador median un formulario, cuenta con la acción de login mediante un botón.

Tabla 13: Caso de Uso iniciar Sesión.

CASO DE USO	Iniciar Sesión
ACTORES	Administrador
PROPÓSITO	Acceso con seguridad Guard a la página principal de la aplicación. la geolocalización en tiempo real de los conductores.
RESUMEN	El administrador ingresa sus respectivos datos como el correo y contraseña que solicita el módulo de administrador para su respectivo



	inicio de sesión en la página principal.
TIPO	Primario
PRE-CONDICIONES	Abrir la aplicación.

Fuente: Elaboración propia.

- Vista geolocalización en tiempo real de todos los vehículos en servicio: Página principal del módulo administrador, geolocalización de todos los vehículos en estado de servicio activo de la empresa de transporte Cristo Morado de Andahuaylas.

Tabla 14: Caso de uso geolocalización en tiempo real.

CASO DE USO	Geolocalización en tiempo real a todos los vehículos en estado de servicio activo.
ACTORES	Administrador
PROPÓSITO	Mostrar al administrador a todos los vehículos en servicio.
RESUMEN	El administrador tendrá la capacidad de visualizar su geolocalización en tiempo real a todos los vehículos en servicio.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista perfil: Pagina donde se muestra los datos personales del administrador

Tabla 15: Caso de uso perfil.

CASO DE USO	Perfil del Administrador
ACTORES	Administrador

PROPÓSITO	Mostrar datos personales del administrador
RESUMEN	Mostrar lo datos personales del administrados como: nombre, Apellidos, dirección, celular, correo y su respectiva foto de perfil.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista Registro de administradores: página de registro al administrador.

*Tabla 16: Caso de uso registro administrador.*

CASO DE USO	Registro
ACTORES	Administrador
PROPÓSITO	Registrar a un nuevo administrador.
RESUMEN	Registro de un nuevo administrador con todos los datos del formulario presentado en la página de registro del administrador con todos los campos validados.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista Registro de conductores y vehículo: Esta página permitirá al administrador registrar a un nuevo conductor y los datos de sus respectivos vehículos.

*Tabla 17: Caso de uso registro conductor*

CASO DE USO	Registro del conductor y vehículo
ACTORES	Administrador

PROPÓSITO	Registrar a un nuevo conductor y vehículo
RESUMEN	Esta página permitirá registrar a un nuevo conductor y su vehículo mediante un formulario, requiere los siguientes campos importantes de un conductor y vehículo: Clave del conductor, nombre, apellido, DNI, celular, correo, fecha de nacimiento, estado de licencia. Y de sus respectivos vehículos: Marca, modelo placa, aforo y estado de (SOAT).
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista lista de conductores: Página que muestra una lista de todos los conductores registrados y sus respectivos datos detallados dentro de un modal.

Tabla 18: Caso de uso lista de conductores.

CASO DE USO	Lista de conductores
ACTORES	Administrador
PROPÓSITO	Mostrar la lista de todos los conductores registrados.
RESUMEN	Mostrar un alista de todos los conductores registrados como también lo datos personales del conductor mediante un modal.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista borrar conductores: página donde el administrador podrá borrar un conductor que ya no pertenece a la empresa.

Tabla 19: Caso de uso borrar conductor.

CASO DE USO	Borrar conductor
ACTORES	Administrador
PROPÓSITO	Eliminar al conductor
RESUMEN	Borrar al conductor dentro de la lista de conductores mediante un botón de opciones.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista actualizar conductores: página donde el administrador podrá actualizar los datos de un conductor de la empresa.

Tabla 20: Caso de uso actualizar conductor

CASO DE USO	Actualizar conductor
ACTORES	Administrador
PROPÓSITO	Eliminar al conductor
RESUMEN	Borrar al conductor dentro de la lista de conductores mediante un botón de opciones.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista mostrar reporte de entrada y salida de los conductores: Página donde visualiza el reporte de entrada y salida de los vehículos durante el día.

Tabla 21: Caso de uso reporte salida y llegada

CASO DE USO	Reporte de entrada y salida
ACTORES	Administrador
PROPÓSITO	Mostrar el reporte de entrada y salida de los vehículos
RESUMEN	Mostrar la entrada y salida de los vehículos en servicio en los paraderos finales e iniciales con la fecha, hora y dirección.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

- Cerrar sesión: botón para cerrar sesión

Tabla 22: Caso de uso cerrar sesión administrador.

CASO DE USO	Cerrar sesión
ACTORES	Administrador
PROPÓSITO	Cerrar sesión de un administrador logeado.
RESUMEN	Botón que permitirá al administrador cerrar su sesión el modulo del administrador.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

## MODULO CONDUCTOR

- Vista Iniciar Sesión con clave: Página principal del módulo conductor, inicio de sesión con una clave que el administrador le brinde.

Tabla 23: Caso de uso iniciar sesión con clave.

CASO DE USO	Iniciar sesión con clave
ACTORES	Conductor

PROPÓSITO	Inicio de sesión con clave
RESUMEN	autenticación del conductor con una clave que el administrador le haya otorgado, para poder iniciar sesión en el módulo del conductor.
TIPO	Primario
PRE-CONDICIONES	Abrir la aplicación.

Fuente: Elaboración propia.

- Vista geolocalización en tiempo real del vehículo: Pagina para mostrar al conductor su ubicación en tiempo real.

Tabla 24: Caso de uso geolocalización del vehículo actual

CASO DE USO	Geolocalización del vehículo
ACTORES	Conductor
PROPÓSITO	Mostrar en tiempo real la geolocalización del vehículo en servicio.
RESUMEN	Página que muestra su ubicación actual en tiempo real al conductor, como también tendrá botones para registrar su entrada y salida de los paraderos finales e iniciales, como también mostrar vehículos cuando están dentro de su radio de 100 metros.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista geolocalización en tiempo real de todos los vehículos en servicio: página que muestra a todos los vehículos en servicio.

Tabla 25: Caso de uso geolocalización de todos los vehículos

CASO DE USO	Geolocalización en tiempo real de todos los vehículos.
ACTORES	Conductor
PROPÓSITO	Mostrar en tiempo real la geolocalización del vehículo en servicio.
RESUMEN	Página que muestra la geolocalización en tiempo real de todos los vehículos en servicio identificándose con su respectiva ruta, nombre y placa.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Alerta para el orden de circulación de los vehículos en servicio: página donde muestra una alerta al vehículo cuando exista vehículos dentro de 100 metros de su ubicación.

Tabla 26: Caso de uso orden de circulación de los vehículos mediante una alerta.

CASO DE USO	Orden de circulación de los vehículos mediante una alerta
ACTORES	Conductor
PROPÓSITO	Ordenar a todos los vehículos en servicio enviando alertas de notificación cuando está dentro de su radio de 100 metros.
RESUMEN	Página donde el conductor podrá observa a todos los vehículos que estén cerca de su ubicación, controlando con una alerta a los

	otros vehículos entren a su radio de 100 metros.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Informe de entrada y salida de los vehículos en los paraderos finales: Página donde podrá observar su entrada y salida en los paraderos iniciales y finales.

Tabla 27: Caso de uso control de salida y llegada

CASO DE USO	Control de Registro de salida y llegada
ACTORES	Conductor
PROPÓSITO	Observar el reporte de salida y llegada de los vehículos
RESUMEN	Mostrar la salida y llegada de los vehículos en servicio en los paraderos finales e iniciales con la fecha, hora y dirección.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Cerrar sesión: botón para cerrar sesión

Tabla 28: Caso de uso cerrar sesión conductor.

CASO DE USO	Cerrar sesión
ACTORES	Conductor
PROPÓSITO	Cerrar sesión de un Conductor logeado.



RESUMEN	Botón que permitirá al administrador cerrar su sesión el modulo del administrador.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

## MÓDULO USUARIO

- Vista Iniciar Sesión: Muestra un formulario para la autenticación del usuario en el módulo de usuario.

Tabla 29: Caso de uso iniciar sesión usuario.

CASO DE USO	Inicio de sesión
ACTORES	Usuario
PROPÓSITO	Autenticación para el usuario
RESUMEN	Muestra un formulario con los campos de correo y contraseña para que el usuario se autentique correctamente, validación de correo existente, botón para el login con Google y Facebook, También las rutas la página de registro y recuperar contraseña
TIPO	Primario
PRE-CONDICIONES	Iniciar la aplicación.

Fuente: Elaboración propia.

- Vista Registro de usuario: página donde muestra un formulario completo para el registro de un usuario.

Tabla 30: Caso de uso de registro usuario.

CASO DE USO	Registrarse
-------------	-------------

ACTORES	Usuario
PROPÓSITO	Registrar a un nuevo Usuario.
RESUMEN	Registro de un nuevo usuario con todos los datos del formulario presentado en la página de registro del usuario y con todos los campos validados.
TIPO	Primario
PRE-CONDICIONES	Iniciar la aplicación

Fuente: Elaboración propia.

- Vista Recupera contraseña: página donde muestra un formulario para ingresar el correo de recuperación.

Tabla 31: Caso de uso recuperar contraseña.

CASO DE USO	Recuperar contraseña
ACTORES	Usuario
PROPÓSITO	Recuperar la contraseña del usuario
RESUMEN	Página donde muestra un formulario para poder recupera la contraseña mediante un correo valido y existente.
TIPO	Primario
PRE-CONDICIONES	Iniciar la aplicación

Fuente: Elaboración propia.

- Vista Verificar Usuario: Apartado que permite al usuario enviar una nota de verificación a su respectivo correo.

Tabla 32: Caso de uso verificar usuario.

CASO DE USO	Verificar usuario
ACTORES	Usuario
PROPÓSITO	Verificar el correo de un usuario

RESUMEN	Permite enviar una nota de verificación a su respectivo correo del usuario para que al iniciar sesión sea válido su correo.
TIPO	Primario
PRE-CONDICIONES	Iniciar la aplicación

Fuente: Elaboración propia.

- Vista Vehículos en servicio: página donde el usuario podrá visualizar la ubicación a todos los vehículos en tiempo real.

Tabla 33: Caso de uso geolocalización de vehículos.

CASO DE USO	Geolocalización e tiempo real de todos los vehículos.
ACTORES	usuario
PROPÓSITO	Mostrar en tiempo real la geolocalización del vehículo en servicio.
RESUMEN	Página que muestra la geolocalización en tiempo real de todos los vehículos en servicio identificándose con su respectiva ruta
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Vista perfil: página para que el usuario pueda visualizar sus datos.

Tabla 34: Caso de uso perfil usuario

CASO DE USO	Perfil del usuario
ACTORES	Usuario

PROPÓSITO	Mostrar datos personales del usuario
RESUMEN	Mostrar lo datos personales del usuario como: nombre, Apellidos, dirección, celular, correo y su respectiva foto de perfil.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

- Cerrar sesión: botón para cerrar sesión del usuario

Tabla 35: Caso de uso Cerrar sesión usuario.

CASO DE USO	Cerrar sesión
ACTORES	Conductor
PROPÓSITO	Cerrar sesión de un usuario logeado.
RESUMEN	Botón que permitirá al administrador cerrar su sesión el modulo del administrador.
TIPO	Primario
PRE-CONDICIONES	Haberse iniciado sesión anteriormente.

Fuente: Elaboración propia.

### 3.6.3. Fase de Producción

En la fase de producción es donde se diseña la aplicación móvil de acuerdo a los casos de uso que se desarrolló en la fase de inicialización, en el que se determinó los requerimientos del proyecto, logrando determinar la funcionalidad que tiene la aplicación móvil. La implementación de la aplicación móvil se llevó a cabo usando un ciclo de desarrollo iterativo e incremental, también se hizo uso del desarrollo dirigido

por pruebas (TDD, Test-Driven Development), el cual significa que cada funcionalidad desarrollada fue probada. 59 Para cada nueva iteración que se hizo, se realizó las siguientes actividades:

- Análisis de requerimientos de la iteración.
- Desarrollo de la iteración con base a los requerimientos.
- Integración de la nueva funcionalidad desarrollada con la rama principal de la aplicación.
- Retroalimentación en base a la iteración.

### 3.6.3.1. Diseño de la Aplicación Móvil

Tomando en cuenta los análisis realizados en la fase de iniciación, en donde se definió los requerimientos de la aplicación móvil. En la siguiente tabla se define la estructura de la aplicación móvil en términos generales.

Tabla 36: Diseño de la aplicación móvil.

Inicio de la aplicación	Los módulos contarán con inicio de sesión(modulo Administrador, modulo Conductor, modulo usuario)
Página de geolocalización	Los tres módulos contarán con la ubicación en tiempo real de todos los vehículos en servicio.
Página de reporte de entrada y salida del vehículo	El modulo del conductor reportará su entrada y salida en los paraderos iniciales y finales, el administrador podrá visualizar la entrada y salida de todos los conductores.
Orden de los vehículos con alerta	Lo vehículos son ordenados mediante una alerta, si existe un vehículo en servicio y entra a su radio de 100 metros, este enviara una alerta de que otro vehículo está en su radio.
Base de datos	El proyecto contara con una base de datos en firebase para gestionar las

	consultas que realizara el administrador, conductor y usuario en la base de datos.
Guards	Implantación del sistema guards para la seguridad de la aplicación.

Fuente: Elaboración propia.

### 3.6.3.2. Dependencias de Paquetes de la Aplicación

Para usar las diferentes funcionalidades de Ionic, debemos importar diferentes paquetes y estos paquetes son de uso global en el proyecto. En el siguiente gráfico se muestra las dependencias utilizadas para este proyecto.

```

"private": true,
"dependencies": {
  "@angular/common": "~10.0.0",
  "@angular/core": "~10.0.0",
  "@angular/fire": "^6.0.3",
  "@angular/forms": "~10.0.0",
  "@angular/platform-browser": "~10.0.0",
  "@angular/platform-browser-dynamic": "~10.0.0",
  "@angular/router": "~10.0.0",
  "@ionic-native/android-permissions": "^5.28.0",
  "@ionic-native/camera": "^5.28.0",
  "@ionic-native/core": "^5.0.0",
  "@ionic-native/geolocation": "^5.28.0",
  "@ionic-native/location-accuracy": "^5.28.0",
  "@ionic-native/splash-screen": "^5.0.0",
  "@ionic-native/status-bar": "^5.0.0",
  "@ionic/angular": "^5.0.0",
  "@ionic/storage": "^2.3.1",
  "cordova-android": "9.0.0",
  "firebase": "^7.21.1",
  "moment": "^2.29.0",
  "rxjs": "~6.5.5",
  "tslib": "^2.0.0",
  "zone.js": "~0.10.3"
},
"devDependencies": {
  "@angular-devkit/architect": ">= 0.900 < 0.1100",
  "@angular-devkit/build-angular": "~0.1000.0",
  "@angular/cli": "~10.0.5",
  "@angular/compiler": "~10.0.0",
  "@angular/compiler-cli": "~10.0.0",
  "@angular/language-service": "~10.0.0",
  "@ionic/angular-toolkit": "^2.3.0",
  "@types/jasmine": "~3.5.0",

```

```

"@types/jasminewd2": "~2.0.3",
"@types/node": "^12.11.1",
"codelyzer": "^6.0.0",
"cordova-plugin-android-permissions": "^1.1.0",
"cordova-plugin-camera": "^5.0.0",
"cordova-plugin-device": "^2.0.2",
"cordova-plugin-geolocation": "^4.0.2",
"cordova-plugin-ionic-keyboard": "^2.2.0",
"cordova-plugin-ionic-webview": "^4.2.1",
"cordova-plugin-request-location-accuracy": "^2.3.0",
"cordova-plugin-splashscreen": "^5.0.2",
"cordova-plugin-statusbar": "^2.4.2",
"cordova-plugin-whitelist": "^1.3.3",
"cordova-sqlite-storage": "^5.1.0",
"firebase-tools": "^8.0.0",
"fuzzy": "^0.1.3",
"inquirer": "^6.2.2",
"inquirer-autocomplete-prompt": "^1.0.1",
"jasmine-core": "~3.5.0",
"jasmine-spec-reporter": "~5.0.0",
"karma": "~5.0.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage-istanbul-reporter": "~3.0.2",
"karma-jasmine": "~3.3.0",
"karma-jasmine-html-reporter": "^1.5.0",
"open": "^7.0.3",
"protractor": "~7.0.0",
"ts-node": "~8.3.0",
"tslint": "~6.1.0",
"typescript": "~3.9.5"
},

```

Figura 64: Dependencias de paquetes de la aplicación.

Fuente: Elaboración propia

### 3.6.3.3. Plugins necesarios para la Aplicación Móvil-Cordova

Todos plugins se instala en la aplicación móvil de acuerdo a una necesidad que se requiera en el proyecto y son de uso pleno de los dispositivos móviles de acuerdo a la plataforma en que se compila, como puede ser Android e IOS, estos plugins permiten el acceso de la aplicación móvil a la cámara, GPS, micrófono, etc. Bajo las dependencias de Cordova. En el siguiente gráfico se muestra los plugins utilizados para el desarrollo de la aplicación móvil.

```

"plugins": {

```

```

"cordova-plugin-whitelist": {},
"cordova-plugin-statusbar": {},
"cordova-plugin-device": {},
"cordova-plugin-splashscreen": {},
"cordova-plugin-ionic-webview": {
  "ANDROID_SUPPORT_ANNOTATIONS_VERSION": "27.+"
},
"cordova-plugin-ionic-keyboard": {},
"cordova-plugin-camera": {
  "ANDROID_SUPPORT_V4_VERSION": "27.+"
},
"cordova-sqlite-storage": {},
"cordova-plugin-geolocation": {},
"cordova-plugin-request-location-accuracy": {
  "PLAY_SERVICES_LOCATION_VERSION": "16.+"
},
"cordova-plugin-android-permissions": {}
},
"platforms": [
  "android",
  "IOS"
]
}

```

Figura 65: Plugins de la aplicación.

Fuente: Elaboración propia.

### 3.6.3.4. Configuración inicial del Proyecto

Como primer paso para generar el proyecto se genera una aplicación con el comando en el terminal node.Js:

```
$ ionic start AppCristomorado blank
```

Este comando permite la creación de toda la estructura del proyecto, junto a esto también se creará el archivo Package.json, que contiene todas las configuraciones de NodeJS. En el archivo “app.module.ts” se deberá importar todas la dependencias y páginas que se usarán en el desarrollo del proyecto. En el siguiente gráfico se muestra la configuración del archivo app.module.ts.

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';

import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';

```



```

import { StatusBar } from '@ionic-native/status-bar/ngx';

import { AppComponent } from './app.component';
import { AppRoutingModuleModule } from './app-routing.module';

import { AngularFireAuthModule } from '@angular/fire/auth';
import { AngularFireModule } from '@angular/fire';
import { environment } from '../environments/environment';
import { Camera } from '@ionic-native/camera/ngx';

import { IonicStorageModule } from '@ionic/storage';
import { Geolocation } from '@ionic-native/geolocation/ngx';
import { AndroidPermissions } from '@ionic-native/android-permissions/ngx';
import { LocationAccuracy } from '@ionic-native/location-accuracy/ngx';

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    IonicStorageModule.forRoot(),
    AppRoutingModuleModule,
    AngularFireModule.initializeApp(environment.firebaseConfig),
    AngularFireAuthModule,
  ],
  providers: [
    StatusBar,
    SplashScreen,
    Camera,
    Geolocation,
    AndroidPermissions,
    LocationAccuracy,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}

```

Figura 66: Configuración de la aplicación `module.ts`

Fuente: Elaboración propia.

### 3.6.3.5. Estructura de los directorios y archivos del proyecto

Al momento de ejecutar el comando `$ ionic start AppCristoMorado blank`, este se encarga de crear la estructura de directorios y archivos necesarios de

manera automática para empezar con la codificación de la aplicación móvil. En el siguiente gráfico se muestra la estructura de directorios y archivos del proyecto.

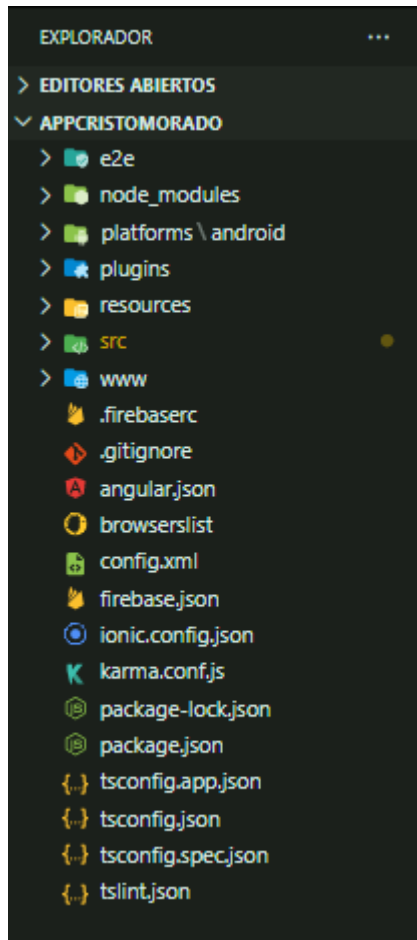


Figura 67: Estructura de la aplicación

Fuente: Elaboración propia.

Tabla 37: Concepto de la estructura de la aplicación

Src	Este es el directorio más importante, porque en ella estarán alojadas todos los archivos de configuración y de programación que forman el proyecto y la aplicación.
www	Contenedor de archivos javascript en una carpeta build e imágenes locales para el icono de la aplicación y el

	SplashScreen, por default o solo de uso local de imágenes.
Package.json	Es un archivo json que contiene información general del proyecto, las dependencias, paquetes y los plugins instalados. Es un archivo muy importante que sirve para migrar el proyecto e instalar nuevamente de ser necesario.
config.xml, ionic.config.json, ionic.starter.json, package.lock.json, tsconfig.json, tslint.json	Todo estos archivos son creados y usados por el framework IONIC independiente.

Fuente: Elaboración propia.

- **Directorio App**

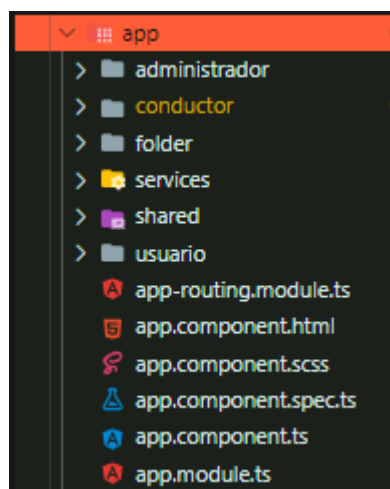


Figura 68: Directorio de la aplicación

Fuente: Elaboración propia

Tabla 38: Concepto del directorio de la aplicación

App	Directorio donde se almacena los diferentes módulos (Administrador, conductor, usuario).
Archivos “.ts” (TypeScript)	TypeScript es un lenguaje de programación libre y de código abierto desarrollado por Microsoft, es un súper conjunto de JavaScript con herramientas de programación orientada a objetos y añade tipado estático.
Archivos “.HTML” .	En este proyecto es el típico archivo html pero con etiqueta ion de ionic 5 y uso de directivas y funcionalidades de angular.
Archivos Sass “.scss”	SASS es una extensión de CSS escrita en RUBY que añade características muy potentes y elegantes a este lenguaje de estilos (CSS). Permite el uso de variables, reglas CSS anidadas, mixins, importación de hojas de estilos y muchos más.
Archivos Routing.ts	Todos estos archivos son creados para especificar la ruta del directorio

Fuente: Elaboración propia.

- **App-routing.module.ts**

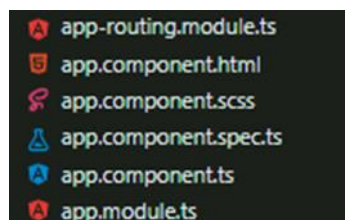


Figura 69: Módulos del directorio de la aplicación

Fuente: Elaboración propia

El módulo del sistema de rutas: llamado RouterModule. Rutas de la aplicación: es un array con un listado de rutas que nuestra aplicación soportará. Enlaces de navegación: son enlaces HTML en los que incluiremos una directiva para indicar que deben funcionar usando el sistema de routing.

- **Directorio Administrador**

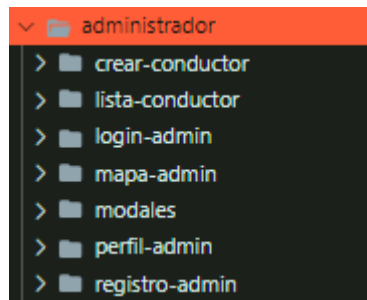


Figura 70: Directorio del administrador de la aplicación..

Fuente: Elaboración propia

Tabla 39 : Concepto del directorio de administrador de la aplicación.

<b>Crear-conductor</b>	Esta página es el encargado de crear a los conductores con sus respectivos datos personales y los datos de sus vehículos, considerando su estado de su BREVETE y el estado del SOAT del vehículo.
<b>Lista-conductor</b>	esta página muestra una lista de todos los conductores. Con las opciones de ver su reporte de entrada y salida de los paraderos iniciales y finales, como también la opción de borrar al conductor
<b>Login-admin</b>	Esta página muestra un formulario para que el administrador se autentifique en el módulo del administrador
<b>Mapa-admin</b>	Página encargada de mostrar el mapa y en ella a todos los vehículos en tiempo real y la opción de seguir a un conductor en específico.

<b>Modales</b>	<p><b>Info-conductor:</b> página que muestra información del conductor, pero en caso de que la clave del conductor coincida con la lista independientemente.</p> <p><b>Lista-conductor:</b> página adicional que muestra un alista de los conductores para el seguimiento individual de los vehículos de un conductor en específico.</p>
<b>Perfil-admin</b>	Página donde muestra los datos personales del administrador.
<b>Registro-admin</b>	Página encargada de registrar a un nuevo administrador.

Fuente: Elaboración propia.

- **Directorio Conductor**

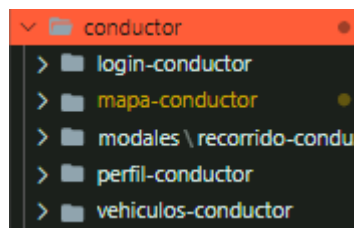


Figura 71. Directorio de conductor de la aplicación.

Fuente: Elaboración propia

Tabla 40: Concepto del directorio de conductor de la aplicación.

<b>Login-conductor</b>	Páginas de autenticación mediante clave, que solo un administrador puede generar.
<b>Mapa-conductor</b>	<p>-Página encargada de subir las coordenadas de los vehículos en servicio en tiempo real a la base de datos.</p> <p>-Enviar alertas de los vehículos que estén dentro de su radio de 100 metros.</p> <p>-Mostrar a los vehículos que estén dentro de su radio.</p>

	-Mostrar nombre, placa y ruta del vehículo en servicio.
<b>Modales</b>	<b>Recorrido:</b> página donde el conductor visualiza su recorrido de la entrada y salida de los paraderos iniciales y finales.
<b>Perfil-conductor</b>	Página encargada de mostrar los datos de los conductores y su respectivo vehículo.
<b>Vehículos-conductor</b>	Página encargada de mostrar a todos los vehículos que estén en servicio.

Fuente: Elaboración propia.

- **Directorio Usuario**

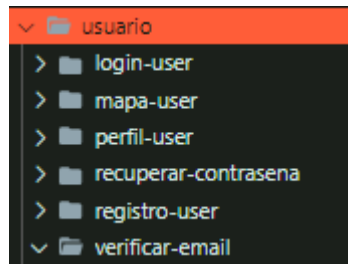


Figura 72: Directorio de conductor de la aplicación

Fuente: Elaboración propia

Tabla 41. Concepto de directorio usuario de la aplicación

<b>Login-user</b>	Esta página muestra un formulario para que el usuario se autentifique en el módulo usuario.
<b>Mapa-user</b>	Página encargada de mostrar el mapa y en ella a todos los vehículos en tiempo real y la opción de seguir a un conductor en específico.
<b>Perfil-user</b>	Página donde muestra los datos personales del usuario.
<b>Recuperar-contrasena</b>	Página encargada de enviar un formulario a su correo respectivo para la recuperación de su contraseña.
<b>Registro-user</b>	página encargada de registrar a un usuario.

<b>Verificar-email</b>	Página encargada de reenviar la verificación de correo si en caso de que no recibió una notificación de verificación.
------------------------	---

Fuente: Elaboración propia.

- **Directorio assets**

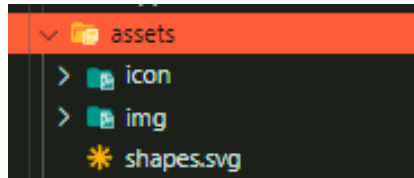


Figura 73: Directorio de Assets de la aplicación

Fuente: Elaboración propia

Tabla 42: Concepto de directorio assets de la aplicación

Assets	Directorio donde se importan imágenes e iconos para el uso local de la aplicación
--------	---

Fuente: Elaboración propia.

- **Directorio Services**

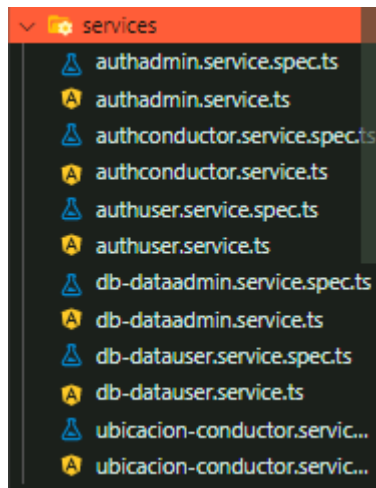


Figura 74: Directorio de Servicios de la aplicación

Fuente: Elaboración propia



Tabla 43: Directorio de Servicios de la aplicación

Services	Los servicios son directorios donde pueden ser reutilizado de forma global en la aplicación en diferentes paginas
----------	---

Fuente: Elaboración propia.

- **Directorio enviroments**

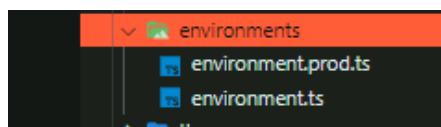


Figura 75: Directorio de enviroments de la aplicación

Fuente: Elaboración propia

Tabla 44: Directorio enviroments de la aplicación

Enviroments	Directorio usado para la configuración y conexión con la base de datos firebase.
-------------	--

Fuente: Elaboración propia.

### 3.6.4. Fase de Estabilización

En esta fase se procedió a integrar las funcionalidades implementadas en los diferentes módulos y de presentarse algún error de código u otros se realizarán las correcciones correspondientes.

El propósito de esta fase es la aseguración de la calidad en la implementación del proyecto.

#### 3.6.4.1. Integración del proyecto con la base de datos

La aplicación móvil hace el uso de una API REST, el cual permite interactuar con una base de datos que se encuentra en un servidor en la red.

Integración de la base datos firebase en el proyecto

```
export const environment = {
  production: false,
  firebaseConfig: {
    apiKey: "AIzaSyDY-JCg1LO_PECyjbVP_ob1ziYJPSFshM",
```

```

authDomain: "tesisristomorado.firebaseio.com",
databaseURL: "https://tesisristomorado.firebaseio.com",
projectId: "tesisristomorado",
storageBucket: "tesisristomorado.appspot.com",
messagingSenderId: "385988085147",
appId: "1:385988085147:web:68cd064a4eeacac52e4d3e",
measurementId: "G-M27ZGVW27D"
},
};

```

Figura 76: Integración del proyecto con la base de datos.

Fuente: Elaboración propia

### 3.6.4.2. Integración de los módulos de Acceso a la aplicación

En los tres módulos que existen en este proyecto, designaremos como página principal el login de usuario.

```

const routes: Routes = [
  {
    path: "",
    redirectTo: 'login-user',
    pathMatch: 'full'
  },
];

```

Figura 77: Integración de los módulos de acceso a la aplicación

Fuente: Elaboración propia

#### Modulo Administrador:

Contenido del archivo crear-conductor.page.html

```

<ion-header>
  <ion-toolbar color="tertiary">
    <ion-buttons slot="start">
      <ion-menu-button autoHide="false"></ion-menu-button>
    </ion-buttons>
    <ion-buttons slot="primary">
      <ion-button routerLink="/mapa-admin" routerDirection="root">
        <ion-icon slot="icon-only" name="arrow-back-outline"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title style="text-align: center;">CONDUCTOR</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>

```

```

<form [formGroup]="agregarconductorForm">

  <ion-card-header>
    <ion-card-title><b>Datos personales</b></ion-card-title>
  </ion-card-header>

  <ion-card-content>
    <ion-item lines="full">
      <ion-label position="floating">clave</ion-label>
      <ion-
input formControlName="clave" type="text" minlength="6" maxlength="16"
required></ion-input>
    </ion-item>
    <div *ngIf="clave && clave.invalid && (clave.dirty || clave.touched)"
>
      <div class="mensajeError" *ngIf=" !!clave.errors.required" style="c
olor: brown;">
        * Complete! el campo es obligatorio.
      </div>
      <div class="mensajeError" *ngIf=" !!clave.errors.minlength" style=
"color: brown;">
        * Debe tener 6 caracteres como minimo.
      </div>
      <div class="mensajeError" *ngIf=" !!clave.errors.maxlength" style
="color: brown;">
        * Debe tener 15 caracteres como máximo.
      </div>
    </div>
    <ion-item lines="full">
      <ion-label position="floating">nombre</ion-label>
      <ion-input style="text-
transform: uppercase;" formControlName="nombre" type="text" minlength="
3" required></ion-input>
    </ion-item>
    <div *ngIf="nombre && nombre.invalid && (nombre.dirty || nombre.t
ouched)">
      <div class="mensajeError" *ngIf=" !!nombre.errors.required" style=
"color: brown;">
        * Complete! el campo es obligatorio.
      </div>
      <div class="mensajeError" *ngIf=" !!nombre.errors.minlength" style
="color: brown;">
        * Debe tener más de 3 caracteres.
      </div>
    </div>
    <ion-item lines="full">
      <ion-label position="floating">Apellidos</ion-label>

```

```

        <ion-input style="text-
transform: uppercase;" formControlName="apellidos" type="text" required></
ion-input>
    </ion-item>
    <div *ngIf="apellidos && apellidos.invalid && (apellidos.dirty || apell
idos.touched)">
        <div class="mensajeError" *ngIf=" !!apellidos.errors.required" style
="color: brown;">
            * Complete! el campo es obligatorio.
        </div>
        <div class="mensajeError" *ngIf=" !!apellidos.errors.minlength" sty
le="color: brown;">
            * Debe tener más de 3 caracteres.
        </div>
    </div>
    <ion-item lines="full">
        <ion-label position="floating">Correo</ion-label>
        <ion-input formControlName="correo" type="text" required></ion-
input>
    </ion-item>
    <div *ngIf="correo && correo.invalid && (correo.dirty || correo.touch
ed)">
        <div class="mensajeError" *ngIf=" !!correo.errors.required" style="
color: brown;">
            * Complete el campo es obligatorio
        </div>
        <div class="mensajeError" *ngIf=" !!correo.errors.minlength" style
="color: brown;">
            * Debe tener al menos 3 caracteres
        </div>
    </div>
    <ion-item lines="full">
        <ion-label position="floating">DNI</ion-label>
        <ion-
input formControlName="dni" type="number" maxLength="8" required></ion-
input>
    </ion-item>
    <div *ngIf="dni && dni.invalid && (dni.dirty || dni.touched)">
        <div class="mensajeError" *ngIf=" !!dni.errors.required" style="col
or: brown;">
            * Complete! el campo es obligatorio.
        </div>
        <div class="mensajeError" *ngIf=" !!dni.errors.minlength" style="c
olor: brown;">
            * El DNI debe tener 8 caracteres.
        </div>
    </div>
    </ion-item>

```

```

        <ion-label>Fecha de nacimiento:</ion-label>
        <ion-
datetime formControlName="fechanacimiento" displayFormat="DD MM YY
YY" placeholder="Select Date"></ion-datetime>
        </ion-item>
        <div *ngIf="fechanacimiento && fechanacimiento.invalid && (fechan
acimiento.dirty || fechanacimiento.touched)">
            <div class="mensajeError" *ngIf=" !!fechanacimiento.errors.require
ed" style="color: brown;">
                * Complete el campo es obligatorio
            </div>
        </div>
        <ion-item lines="full">
            <ion-label position="floating">Celular</ion-label>
            <ion-
input formControlName="celular" type="number" maxLength="9" required></
ion-input>
            </ion-item>
            <div *ngIf="celular && celular.invalid && (celular.dirty || celular.touc
hed)">
                <div class="mensajeError" *ngIf=" !!celular.errors.required" style=
"color: brown;">
                    * Complete! el campo es obligatorio.
                </div>
                <div class="mensajeError" *ngIf=" !!celular.errors.min" style="col
or: brown;">
                    * El Celular debe tener 9 caracteres.
                </div>
                <div class="mensajeError" *ngIf=" !!celular.errors.max" style="col
or: brown;">
                    * El Celular no debe superar los 9 caracteres.
                </div>
            </div>
            <ion-radio-group formControlName="licencia">
                <ion-item lines="full">
                    <ion-label>Licencia de conducir (BREVETE)</ion-label>
                </ion-item>
                <ion-item>
                    <ion-label>Activo</ion-label>
                    <ion-radio color="tertiary" slot="start" value="Activo"></ion-
radio>
                </ion-item>
                <ion-item>
                    <ion-label>Inactivo</ion-label>
                    <ion-radio color="tertiary" slot="start" value="Inactivo"></ion-
radio>
                </ion-item>
            </ion-radio-group>

```

```

        <div *ngIf="licencia && licencia.invalid && (licencia.dirty || licencia.
touched)">
            <div class="mensajeError" *ngIf=" !!licencia.errors.required" style
="color: brown;">
                * Complete el campo es obligatorio
            </div>
        </div>
    </ion-card-content>

    <ion-card-header>
        <ion-card-title><b>Datos del Vehículo</b></ion-card-title>
    </ion-card-header>

    <ion-card-content>
        <ion-item lines="full">
            <ion-label position="floating">Marca</ion-label>
            <ion-input formControlName="marca" type="text" required></ion-
input>
        </ion-item>
        <div *ngIf="marca && marca.invalid && (marca.dirty || marca.touche
d)">
            <div class="mensajeError" *ngIf=" !!marca.errors.required" style="
color: brown;">
                * Complete el campo es obligatorio
            </div>
            <div class="mensajeError" *ngIf=" !!marca.errors.minlength" style
="color: brown;">
                * Debe tener al menos 3 caracteres
            </div>
        </div>
        <ion-item lines="full">
            <ion-label position="floating">Modelo</ion-label>
            <ion-
input formControlName="modelo" type="text" required></ion-input>
        </ion-item>
        <div *ngIf="modelo && modelo.invalid && (modelo.dirty || modelo.t
ouched)">
            <div class="mensajeError" *ngIf=" !!modelo.errors.required" style
="color: brown;">
                * Complete el campo es obligatorio
            </div>
            <div class="mensajeError" *ngIf=" !!modelo.errors.minlength" styl
e="color: brown;">
                * Debe tener al menos 3 caracteres
            </div>
        </div>
        <ion-item lines="full">
            <ion-label position="floating">Placa del Vehículo</ion-label>

```

```

        <ion-input style="text-
transform: uppercase;" formControlName="placa" type="text" maxlength="7"
required"></ion-input>
    </ion-item>
    <div *ngIf="placa && placa.invalid && (placa.dirty || placa.touched)"
>
        <div class="mensajeError" *ngIf=" !!placa.errors.required" style="c
olor: brown;">
            * Complete! el campo es obligatorio.
        </div>
        <div class="mensajeError" *ngIf=" !!placa.errors.minlength" style="
"color: brown;">
            * El placa debe tener 7 caracteres.
        </div>
    </div>
    <ion-item lines="full">
        <ion-label position="floating">Aforo</ion-label>
        <ion-
input formControlName="aforo" type="number" required"></ion-input>
    </ion-item>
    <div *ngIf="aforo && aforo.invalid && (aforo.dirty || aforo.touched)"
>
        <div class="mensajeError" *ngIf=" !!aforo.errors.required" style="c
olor: brown;">
            * Complete el campo es obligatorio
        </div>
    </div>
    <ion-radio-group formControlName="soat">
        <ion-item lines="full">
            <ion-
label>Seguro Obligatorio de Accidentes de Tránsito (SOAT)</ion-label>
        </ion-item>
        <ion-item>
            <ion-label>Activo</ion-label>
            <ion-radio slot="start" value="Activo"></ion-radio>
        </ion-item>
        <ion-item>
            <ion-label>Inactivo</ion-label>
            <ion-radio slot="start" value="Inactivo"></ion-radio>
        </ion-item>
    </ion-radio-group>
    <div *ngIf="soat && soat.invalid && (soat.dirty || soat.touched)">
        <div class="mensajeError" *ngIf=" !!soat.errors.required" style="co
lor: brown;">
            * Complete el campo es obligatorio
        </div>
    </div>
</ion-card-content>

```

```

    <ion-row>
      <ion-col style="text-align: center;">
        <ion-
button color="tertiary" shape="round" fill="outline" type="submit" (click)="g
uardarConductor()">Crear</ion-button>
      </ion-col>
    </ion-row>
  -
</form>
</ion-content>

```

Figura 78: crear-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo crear-conductor.page.ts

```

import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';
import { ToastController } from '@ionic/angular';
import { Router } from '@angular/router';
import { AngularFireStore } from '@angular/fire/firestore';

@Component({
  selector: 'app-crear-conductor',
  templateUrl: './crear-conductor.page.html',
  styleUrls: ['./crear-conductor.page.scss'],
})
export class CrearConductorPage implements OnInit {
  agregarconductorForm: FormGroup;

  constructor(private afs: AngularFireStore,
    private router: Router,
    private ToastController: ToastController) {
    this.agregarconductorForm = this.createFormGroupAgregarConductor();
  }

  ngOnInit() {
  }

  createFormGroupAgregarConductor() {
    return new FormGroup({
      // tslint:disable-next-line:max-line-length
      clave: new FormControl("", [Validators.required, Validators.minLength(6),
Validators.maxLength(15)]),
      nombre: new FormControl("", [Validators.required, Validators.minLength(
3)]),
      lat: new FormControl(0),
      lng: new FormControl(0),

```



```

    apellidos: new FormControl("", [Validators.required, Validators.minLength(3)]),
    correo: new FormControl("", [Validators.required, Validators.minLength(3), Validators.email]),
    dni: new FormControl("", [Validators.required, Validators.minLength(8)]),
    fechanacimiento: new FormControl("", [Validators.required]),
    celular: new FormControl("", [Validators.required, Validators.min(900000000), Validators.max(999999999)]),
    licencia: new FormControl('Inactivo', [Validators.required]),
    marca: new FormControl("", [Validators.required, Validators.minLength(3)]),
    modelo: new FormControl("", [Validators.required, Validators.minLength(3)]),
    placa: new FormControl("", [Validators.required, Validators.minLength(7)]),
    aforo: new FormControl("", [Validators.required]),
    soat: new FormControl('Inactivo', [Validators.required]),
  });
}

get nombre() { return this.agregarconductorForm.get('nombre'); }
get clave() { return this.agregarconductorForm.get('clave'); }
get lat() { return this.agregarconductorForm.get('lat'); }
get lng() { return this.agregarconductorForm.get('lng'); }
get apellidos() { return this.agregarconductorForm.get('apellidos'); }
get correo() { return this.agregarconductorForm.get('correo'); }
get dni() { return this.agregarconductorForm.get('dni'); }
get fechanacimiento() { return this.agregarconductorForm.get('fechanacimiento'); }
get celular() { return this.agregarconductorForm.get('celular'); }
get licencia() { return this.agregarconductorForm.get('licencia'); }
get marca() { return this.agregarconductorForm.get('marca'); }
get modelo() { return this.agregarconductorForm.get('modelo'); }
get placa() { return this.agregarconductorForm.get('placa'); }
get aforo() { return this.agregarconductorForm.get('aforo'); }
get soat() { return this.agregarconductorForm.get('soat'); }

guardarConductor(){
  if (this.agregarconductorForm.valid){
    this.afs.collection('conductores').doc(this.agregarconductorForm.value.clave).set(this.agregarconductorForm.value);
    this.router.navigate(['lista-conductor']);
  }else{
    console.log('no se pudo guardar el registro');
    this.presentToast('Los campos estan incompletos!!', 'danger');
  }
}
}

```

```

async presentToast(mensaje, color1) {
  const toast = await this.ToastController.create({
    message: mensaje,
    duration: 3000,
    color: color1,
    buttons: [
      {
        text: 'Aceptar',
        role: 'cancel',
        handler: () => {
        }
      }
    ]
  });
  toast.present();
}
}

```

Figura 79: crear-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo lista-conductor.page.html

```

<ion-content>
  <ion-list>
    <ion-item-sliding *ngFor="let item of conductorcombi">
      <ion-item (click)="modalconductordetalle(item)">
        <ion-icon color="tertiary" name="person" slot="start"></ion-icon>
        <ion-label>
          <h2 style="text-
transform: uppercase;"><b>{{item.nombre}} &nbsp;&nbsp;&nbsp;{{item.apellidos}}
</b>
          </h2>
          <p>Licencia:&nbsp;&nbsp;&nbsp;{{item.licencia}}</p>
        </ion-label>
      </ion-item>
      <ion-item-options side="end">
        <ion-item-
option color="success" (click)="presentModalRecorrido(item.clave)">
          <ion-icon slot="icon-only" name="document-text"></ion-icon>
        </ion-item-option>
        <ion-item-option color="danger" (click)="borrarConductor(item)">
          <ion-icon slot="icon-only" name="trash"></ion-icon>
        </ion-item-option>
      </ion-item-options>
    </ion-item-sliding>
  </ion-list>

```

```

<ion-fab vertical="bottom" horizontal="end" slot="fixed">
  <ion-fab-
button color="tertiary" routerLink="/conductor" routerDirection="root">
  <ion-icon name="add-outline"></ion-icon>
  </ion-fab-button>
</ion-fab>
</ion-content>

```

Figura 80: lista-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo lista-conductor.page.ts

```

export class ListaConductorPage implements OnInit {
  conductorcombi: conductor[] = [];

  constructor(
    private afs: AngularFireStore,
    private modalCtrl: ModalController,
    private router: Router,
    public alertController: AlertController,
    private modalController: ModalController) {
    afs.collection('conductores').valueChanges()
      .subscribe((data: conductor[]) => {
        this.conductorcombi = data;
        console.log('conduc', data);
      });
  }

  ngOnInit() {
  }

  async modalconductordetalle(dato) {
    const modal = await this.modalCtrl.create({
      component: InfoConductorPage,
      cssClass: 'my-custom-class',
      componentProps: {infoconductor: dato}
    });
    return await modal.present();
  }

  async borrarConductor(item: conductor) {
    const alert = await this.alertController.create({
      cssClass: 'my-custom-class',
      header: 'Confirmar borrado',
      message: '¿Estás seguro de que deseas eliminar al conductor?',
      buttons: [
        {

```

```

text: 'Cancelar',
role: 'cancel',
cssClass: 'secondary',
handler: (blah) => {
  console.log('Confirm Cancel: blah');
}
}, {
text: 'Eliminar',
handler: () => {
  this.afs.doc(`/conductores/${item.clave}`).delete().then(() => {
    console.log(`conductor eliminado ="${item.nombre}"`);
    this.router.navigate(['listaconductor']);
  }).catch(err => {
    console.log(err);
  });
}
}
]);
await alert.present();
}

async presentModalRecorrido(clave1: string) {
  const modal = await this.modalController.create({
    component: RecorridoConductorPage,
    cssClass: 'my-custom-class',
    componentProps: {clave: clave1}
  });
  return await modal.present();
}

```

Figura 81: lista-conductor.page.ts

Fuente: Elaboración propia

Contenido del archivo login-admin.page.html

```

<ion-content>
  <div style="width: 100%; text-align: center;">
    <p><b>Empresa de Transportes Cristo Morado Andahuaylas S.A</b></p>
  >
  
  </div>
  <br>
  <form [formGroup]="loginForm">
    <ion-item lines="full">
      <ion-icon color="tertiary" name="mail" slot="start"></ion-icon>
      <ion-label color="tertiary" position="floating">Email</ion-label>
    </ion-item>
  </form>

```

```

        <ion-input type="text" formControlName="email" required></ion-
input>
    </ion-item>
    <div *ngIf="email && email.invalid && (email.dirty || email.touched)">
        <div class="mensajeError" *ngIf=" !!email.errors.required">
            * Complete el campo es obligatorio
        </div>
        <div class="mensajeError" *ngIf=" !!email.errors.minlength">
            * Debe tener al menos 3 caracteres
        </div>
    </div>
    <ion-item lines="full">
        <ion-icon color="tertiary" name="lock-closed" slot="start"></ion-
icon>
        <ion-label color="tertiary" position="floating">Contraseña</ion-
label>
        <ion-
input type="password" formControlName="password" required></ion-input>
    </ion-item>
    <div *ngIf="password && password.invalid && (password.dirty || passw
ord.touched)">
        <div class="mensajeError" *ngIf=" !!password.errors.required">
            * Complete el campo es obligatorio
        </div>
        <div class="mensajeError" *ngIf=" !!password.errors.minlength">
            * Debe tener al menos 3 caracteres
        </div>
    </div>
    <ion-row>
        <ion-col style="text-align: center;">
            <ion-
button expand="full" color="tertiary" shape="round" type="submit" (click)="
onLogin()">Iniciar Sesion</ion-button>
        </ion-col>
    </ion-row>
</form>
</ion-content>

```

Figura 82: login-admin.page.html

Fuente: Elaboración propia

Contenido del archivo login-admin.page.ts

```

export class LoginAdminPage implements OnInit {
    loginForm: FormGroup;
    constructor(
        private authSvc: AuthadminService,

```

```

private router: Router,
public actionSheetController: ActionSheetController,
private ToastController: ToastController) {
  this.loginForm = this.createFormGroupLogin();
}

ngOnInit() {
}

createFormGroupLogin() {
  return new FormGroup({
    email: new FormControl("", [Validators.required, Validators.email]),
    password: new FormControl("", [Validators.required, Validators.minLength
(3)]),
  });
}

get email() { return this.loginForm.get('email'); }
get password() { return this.loginForm.get('password'); }

async onLogin(){
  if (this.loginForm.valid) {
    try{
      const user = await this.authSvc.login(this.loginForm.value.email, this.log
inForm.value.password);
      if (user){
        console.log('user->', user);
        this.router.navigate(['mapa-admin']);
        // this.redirectUser(isVerificar);
        // Todo: CheckEmail
        // this.router.navigate(['admin']);
      }
    }catch (error){
      console.log('Error', error);
    }
  }else{
    console.log('no se pudo registrar');
    this.presentToast('Complete los campos de manera correcta', 'danger');
  }
}

async presentToast(mensaje, color1) {
  const toast = await this.ToastController.create({
    message: mensaje,
    duration: 3000,
    color: color1,
    buttons: [
      {

```

```

    text: 'Aceptar',
    role: 'cancel',
    handler: () => {
    }
  }
]
});
toast.present();
}

async opcionPerfil() {
const actionSheet = await this.actionSheetController.create({
  buttons: [{
    text: 'Usuario',
    icon: 'person',
    handler: () => {
      this.router.navigate(['login-user']);
    }
  }, {
    text: 'Conductor',
    icon: 'person',
    handler: () => {
      this.router.navigate(['login-conductor']);
    }
  }, {
    text: 'Administrador',
    icon: 'person',
    handler: () => {
      this.router.navigate(['login-admin']);
    }
  }, {
    text: 'Cancelar',
    icon: 'close',
    role: 'cancel',
    handler: () => {
      console.log('Cancelado');
    }
  }
  ]
});
await actionSheet.present();
}
}

```

Figura 83: login-admin.page.ts

Fuente: Elaboración propia

Contenido del archivo mapa-admin.page.html

```

<ion-content [fullscreen]="true">
  <ion-fab vertical="bottom" horizontal="start" slot="fixed">
    <ion-fab-button color="tertiary" (click)="modalConductores()">
      <ion-icon name="people-outline"></ion-icon>
    </ion-fab-button>
  </ion-fab>
  <div id="map" #map></div>
</ion-content>

```

Figura 84: mapa-admin.page.html

Fuente: Elaboración propia

Contenido del archivo mapa-admin.page.ts

```

})
export class MapaAdminPage implements OnInit {

  lat = -13.66262190505499;
  lng = -73.38470607953863;
  init = false;
  conductorcombi: conductores[] = [];
  siguiendoa: string = null;
  siguiendonombre: string = null;
  constructor(db: AngularFireStore,
    private activatedRoute: ActivatedRoute,
    private modalController: ModalController) {
    db.collection('conductores').valueChanges()
      .subscribe((data: conductores[]) => {
        this.conductorcombi = data;
        this.loadMap(null);

        if (!this.init){
          this.lat = data[0].lat;
          this.lng = data[0].lng;
          this.init = true;
        }
      });
  }

  ngOnInit() {

  }

  loadMap(data){
    let distZoom;
    if (data) {
      this.lat = data.lat;
      this.lng = data.lng;
    }
  }
}

```



```

    distZoom = data.zoom;
  } else {
    distZoom = 14;
  }
  // console.log('llega', data);
  const myLatLng = {
    lat: this.lat,
    lng: this.lng
  };

  const mapEle: HTMLInputElement = document.getElementById('map');
  const map = new google.maps.Map(mapEle, {
    center: myLatLng,
    zoom: distZoom,
  });

  for (let m of this.conductorcombi) {
    // console.log('qqqq', m);
    const myLatLng = new google.maps.LatLng(m.lat , m.lng);
    const marker = new google.maps.Marker({
      position: myLatLng,
      map,
      label: m.nombre,
    });
  }
}

seguir(chofer: conductores){
  console.log(chofer);
  // this.siguiendoa = chofer.clave;
  this.siquiendonombre = chofer.nombre;
  const coords = {
    lat: chofer.lat,
    lng: chofer.lng,
    zoom: 17
  };
  this.loadMap(coords);
  // this.lat = chofer.lat;
  // this.lng = chofer.lng;
}

async modalConductores() {
  const modal = await this.modalController.create({
    component: ListaConductoresPage,
    cssClass: 'my-custom-class'
  });
  await modal.present();
}

```





```

}

getDatos(email) {
  this.dbData.obtenerDatosAdministrador(email).subscribe( data => {
    console.log('DATA:', data);
    if (data) {
      this.datosAdmi = data;
    }
  });
}

createFormGroupUpdate() {
  return new FormGroup({
    // tslint:disable-next-line:max-line-length
    direccion: new FormControl(),
    foto: new FormControl(""),
  });
}

get direccion() {return this.updateForm.get('direccion'); }

sacarFotoCamara(){

  let cameraOptions: CameraOptions = {
    quality: 50,
    encodingType: this.camera.EncodingType.JPEG,
    targetWidth: 500,
    targetHeight: 500,
    destinationType: this.camera.DestinationType.DATA_URL,
    sourceType: this.camera.PictureSourceType.CAMERA,
    correctOrientation: true
  };

  this.camera.getPicture(cameraOptions).then((imageData) => {
    // imageData is a base64 encoded string
    this.foto = 'data:image/jpeg;base64,' + imageData;
  }, (err) => {
    console.log(err);
  });
}

sacarFotoGaleria(){

  let cameraOptions: CameraOptions = {
    quality: 50,
    encodingType: this.camera.EncodingType.JPEG,
    targetWidth: 500,
    targetHeight: 500,

```

```

destinationType: this.camera.DestinationType.DATA_URL,
sourceType: this.camera.PictureSourceType.PHOTOLIBRARY,
correctOrientation: true
};

this.camera.getPicture(cameraOptions).then((imageData) => {
  // imageData is a base64 encoded string
  this.foto = 'data:image/jpeg;base64,' + imageData;
}, (err) => {
  console.log(err);
});
}

actualizarFoto() {
  console.log(this.updateForm.value);
  if (this.updateForm.valid) {
    if (isNullOrUndefined(this.foto)) {
      console.log('Selecciona una foto');
      this.updateForm.removeControl('foto');
    } else {
      this.updateForm.value.foto = this.foto;
    }
    this.dbData.actualizarDatos(this.datosAdmi.email, this.updateForm.value);
  } else {
    console.log('complete los datos de manera correcta');
  }
}

async opcionFoto() {
  const actionSheet = await this.actionSheetController.create({
    buttons: [{
      text: 'Cargr desde galería',
      icon: 'images',
      handler: () => {
        this.sacarFotoGaleria();
      }
    }, {
      text: 'Usar la cámara',
      icon: 'camera',
      handler: () => {
        this.sacarFotoCamara();
      }
    }, {
      text: 'Cancelar',
      icon: 'close',
      role: 'cancel',
      handler: () => {
        console.log('Cancelado');
      }
    }
  ]
});
}

```

```

    }
  }}
});
  await actionSheet.present();
}
}

```

Figura 87: perfil-admin.page.ts

Fuente: Elaboración propia

Contenido del archivo registro-admin.page.html

```

<ion-content>
  <form [formGroup]="registroForm">
    <ion-item lines="full">
      <ion-label position="floating">Nombre</ion-label>
      <ion-input formControlName="nombre" type="text" required</ion-
input>
    </ion-item>
    <ion-item lines="full">
      <ion-label position="floating">Apellidos</ion-label>
      <ion-input formControlName="apellidos" type="text" required</ion-
input>
    </ion-item>
    <ion-item lines="full">
      <ion-label position="floating">Dirección</ion-label>
      <ion-input formControlName="direccion" type="text" required</ion-
input>
    </ion-item>
    <ion-item lines="full">
      <ion-label position="floating">Celular</ion-label>
      <ion-
input formControlName="celular" type="number" required</ion-input>
    </ion-item>
    <ion-item lines="full">
      <ion-label position="floating">Email</ion-label>
      <ion-input type="text" formControlName="email" required</ion-
input>
    </ion-item>
    <ion-item lines="full">
      <ion-label position="floating">Contraseña</ion-label>
      <ion-
input type="password" formControlName="password" required</ion-input>
    </ion-item>
    <ion-row>
      <ion-col style="text-align: center;">

```

```

        <ion-
button color="tertiary" shape="round" type="submit" (click)="onRegistro()">
Registrar</ion-button>
        </ion-col>
    </ion-row>
</form>
</ion-content>

```

Figura 88: registro-admin.page.html

Fuente: Elaboración propia

Contenido del archivo registro-admin.page.ts

```

export class RegistroAdminPage implements OnInit {
registroForm: FormGroup; // slider 8

constructor(private authSvc: AuthadminService,
private afs: AngularFirestore,
private router: Router) {
this.registroForm = this.createFormGroupRegistro();
}

ngOnInit() {
}

createFormGroupRegistro() {
return new FormGroup({
// tslint:disable-next-line:max-line-length
nombre: new FormControl(""),
apellidos: new FormControl(""),
direccion: new FormControl(""),
celular: new FormControl(""),
email: new FormControl(""),
password: new FormControl(""),
foto: new FormControl(""),
});
}

get nombre() { return this.registroForm.get('nombre'); }
get apellidos() { return this.registroForm.get('apellidos'); }
get direccion() { return this.registroForm.get('direccion'); }
get celular() { return this.registroForm.get('celular'); }
get email() { return this.registroForm.get('email'); }
get password() { return this.registroForm.get('password'); }
get foto() { return this.registroForm.get('foto'); }

async onRegistro(){

```

```

console.log('infor', this.registroForm.value);

if (this.registroForm.valid) {
  try{
    const user = await this.authSvc.register(this.registroForm.value.email, thi
s.registroForm.value.password);
    if (user){
      this.guardarUsuario();
      // console.log('user->', user);
      // this.router.navigate(['admin']);
    }
  }catch (error){
    console.log('Error', error);
  }
  }else{
    console.log('no se pudo registrar');
  }
}

private guardarUsuario(){
  if (this.registroForm.valid){
    this.afs.collection('administradores').doc(this.registroForm.value.email).set
(this.registroForm.value);
  }else{
    console.log('no se pudo guardar el registro');
  }
}

```

Figura 89: registro-admin.page.ts

Fuente: Elaboración propia

## Modulo conductor

Contenido del archivo login-conductor.page.html

```

<ion-content padding>
  <ion-slides pager="true" #slidePrincipal>
    <ion-slide>
      <div style="width: 100%; text-align: center;">
        <p><b>Empresa de Transportes Cristo Morado Andahuaylas S.A</
b></p>
        
      </div>
      <h2 class="slide-title">Ingrese su código</h2>
      <p>Debe de ingresar su código de conductor</p>
      <ion-button shape="round" color="tertiary" (click)="mostrarAlerta()">
        Ingresar

```



```

</ion-button>
<br>
<br>
<br>
<br>
<br>
</ion-slide>
</ion-slides>
</ion-content>

```

Figura 90: login-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo login-conductor.page.ts

```

import { Component, OnInit, ViewChild } from '@angular/core';
import { AlertController, IonSlides, LoadingController, ActionSheetController
} from '@ionic/angular';
import { AuthconductorService } from '../services/authconductor.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-login-conductor',
  templateUrl: './login-conductor.page.html',
  styleUrls: ['./login-conductor.page.scss'],
})
export class LoginConductorPage implements OnInit {
  @ViewChild('slidePrincipal', {static: false}) slider: IonSlides;

  constructor(
    public alertController: AlertController,
    public loadingController: LoadingController,
    public usuarioservice: AuthconductorService,
    private router: Router,
    public actionSheetController: ActionSheetController) {}

  ionViewDidEnter(){
    this.slider.lockSwipeToNext(true);
  }

  ngOnInit() { }

  async mostrarAlerta(){
    const alert = await this.alertController.create({
      header: 'ingrese usuario',
      inputs: [{
        name: 'username',
        placeholder: 'username'

```

```

    }],
    buttons: [{
      text: 'Cancel',
      role: 'cancel',
    }, {
      text: 'ingresar',
      handler: (data) => {
        this.verificarUsurio(data.username);
      }
    }
  ]
});
await alert.present();
}

async verificarUsurio( clave: string ) {
  const loading = await this.loadingController.create({
    cssClass: 'my-custom-class',
    message: 'Cargando...',
    duration: 1000
  });
  this.usuarioservice.verificaUsuario( clave )
  .then(existe => {
    // console.log(existe);
    if (existe){
      this.slider.lockSwipeToNext(false);
      this.slider.slideNext();
      this.router.navigate(['mapa-conductor']);

    }else{
      // ---
      this.alertaError();
      // ---
    }
  });

  await loading.present();
}

async alertaError(){
  const alert = await this.alertController.create({
    header: 'Contactece con el administrador o intente de nuevo',
    buttons: [{
      text: 'okey',
    }
  ]
  });
  await alert.present();
}

```

```

}

async opcionPerfil() {
  const actionSheet = await this.actionSheetController.create({
    buttons: [{
      text: 'Usuario',
      icon: 'person',
      handler: () => {
        this.router.navigate(['login-user']);
      }
    }, {
      text: 'Conductor',
      icon: 'person',
      handler: () => {
        this.router.navigate(['login-conductor']);
      }
    }, {
      text: 'Administrador',
      icon: 'person',
      handler: () => {
        this.router.navigate(['login-admin']);
      }
    }, {
      text: 'Cancelar',
      icon: 'close',
      role: 'cancel',
      handler: () => {
        console.log('Cancelado');
      }
    }
  ]
  });
  await actionSheet.present();
}
}

```

Figura 91: login-conductor.page.ts

Fuente: Elaboración propia

Contenido del archivo mapa-conductor.page.html

```

<ion-content>
  <div id="map" #map></div>
  <ion-fab horizontal="end" vertical="bottom" slot="fixed">
    <ion-fab-button color="dark">
      <ion-icon name="caret-back-outline"></ion-icon>
    </ion-fab-button>
  <ion-fab-list side="start">

```

```

        <ion-
button color="tertiary" (click)="presentAlertConfirm('ENTRADA')" shape="rou
nd">Entrada</ion-button>
        <ion-
button color="tertiary" (click)="presentAlertConfirm('SALIDA')" shape="rou
nd">Salida</ion-button>
        </ion-fab-list>
    </ion-fab>
</ion-content>

```

Figura 92: mapa-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo mapa-conductor.page.ts

```

export class MapaConductorPage implements OnInit {
    user: User;
    conductores = [];
    infowindow: any;
    map: any;
    geocoderr: any;
    address: string;
    constructor(public ubicationservice: UbicacionConductorService,
        public usuarioservice: AuthconductorService,
        private router: Router,
        private db: AngularFirestore,
        private toastController: ToastController,
        private alertController: AlertController,
        private Zone: NgZone,
        private dbDataUser: DbDatauserService,
        private modalController: ModalController) {
        this.ubicationservice.iniciarGeolocalizacion();
        this.ubicationservice.inicializarTaxista();

        // this.user = {};
        this.ubicationservice.taxista.valueChanges().subscribe(data => {
            this.user = data;
            this.loadMap(this.user);
            // console.log('ADRIAN',data)
        });
    }
    ngOnInit(){
        console.log('FECHA', formatDate(new Date(), 'dd/MM/yyyy', 'en'));
    }

    obtenerDireccionApiGoogle(LatLng) {

```

```

// tslint:disable-next-line:object-literal-key-quotes
this.geocoderr.geocode({ 'location': LatLng }, (results, status) => {
  if (status === 'OK') {
    if (results[0]) {
      // alert(results[0].formatted_address);
      // console.log(results[0].formatted_address);
      // console.log('country' , results[0]);
      this.Zone.run(() => {this.address = results[0].formatted_address; });
      // this.address = results[0].formatted_address; // TIENES K DECLARA
R LA VARIABLE ADDRES
      console.log(this.address);
    } else {
      window.alert('No results found');
    }
  } else {
    window.alert('Geocoder failed due to: ' + status);
  }
});
}

loadMap(data){
  this.geocoderr = new google.maps.Geocoder(); // solo agregas esto Y DECL
ARAR VARIABLE GEOCODERR
  // console.log('llega', data);
  const myLatLng = {
    lat: data.lat,
    lng: data.lng
  };

  const mapEle: HTMLInputElement = document.getElementById('map');
  this.map = new google.maps.Map(mapEle, {
    center: myLatLng,
    zoom: 18
  });

  this.obtenerDireccionApiGoogle(myLatLng);

// tslint:disable-next-line: no-unused-expression
const marker = new google.maps.Marker({
  position: myLatLng,
  map: this.map,
  title: 'Hello World!',
  label: 'YO',
});
// AGREGA UN CIRCULO ROJO
const cityCircle1 = new google.maps.Circle({
  strokeColor: '#FF0000',
  strokeOpacity: 0.8,

```

```

strokeWeight: 2,
fillColor: '#FF0000',
fillOpacity: 0.35,
map: this.map,
center: myLatLng,
radius: 100,
});
// MUESTRA INFORMACION AL HACER CLICK AL MARKER
google.maps.event.addListener(marker, 'click', () => {
  if (!this.infowindow) {
    this.infowindow = new google.maps.InfoWindow();
  }
  const contentStringconductor =
    '<div>' +
    '<h3 style="text-align: center; text-
transform: uppercase;">' + this.user.nombre + '</h3>' +
    '<h6>Placa:' + this.user.placa + '</h6>' +
    '</div>';
  // tslint:disable-next-line:max-line-length
  this.infowindow.setContent(contentStringconductor);
  this.infowindow.open(this.map, marker);
});
// LA LISTA DE CONDUCTORES, VAMOS A COMPARAR CON EL USU
ARIO ACTUAL
// OBTENGO LA LISTA DE CONDUCTORES
this.db.collection('conductores').valueChanges()
.subscribe(lista => {
  lista.forEach((element: User) => {
    console.log(element, this.user);
    if (this.user.clave !== element.clave) {
      if (this.getDistance(this.user.lat, this.user.lng, element.lat, element.lng) <
100) {
        this.presentToast('El conductor ' + element.nombre + ' esta cerca');
        const myLatLng = {
          lat: element.lat,
          lng: element.lng
        };
        // tslint:disable-next-line: no-unused-expression
        const marker = new google.maps.Marker({
          position: myLatLng,
          map: this.map,
          title: 'Conductor',
          //label: element.nombre,
          icon: '../assets/img/combimarker.png'
        });
        google.maps.event.addListener(marker, 'click', () => {
          if (!this.infowindow) {
            this.infowindow = new google.maps.InfoWindow();

```

```

    }
    const contentString =
      '<div>' +
      '<h3 style="text-align: center; text-
transform: uppercase;">' + element.nombre + '</h3>' +
      '<h6>Placa:' + element.placa + '</h6>' +
      '</div>';
    // tslint:disable-next-line:max-line-length
    this.infowindow.setContent(contentString);
    this.infowindow.open(this.map, marker);
  });
} else {
  console.log('el conductor sta lejos de 100 metros');
}
}
});
});
}

rad(x) {
  return x * Math.PI / 180;
}

getDistance(lat1, lng1, lat2, lng2) {
  const R = 6378137; // Earth's mean radius in meter
  const dLat = this.rad(lat2 - lat1);
  const dLong = this.rad(lng2 - lng1);
  const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(this.rad(lat1)) * Math.cos(this.rad(lat2)) *
    Math.sin(dLong / 2) * Math.sin(dLong / 2);
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  const d = R * c;
  console.log('Distancia', d);
  // if (d < 100) {
  //   alert('Usted se encuentra en una distancia menor a 100 metros');
  // }
  return d; // returns the distance in meter retorna distancia en metros
}

async presentToast(mensaje: string) {
  const toast = await this.toastController.create({
    message: mensaje,
    duration: 3000,
    color: 'danger',
    position: 'top',
    buttons: [
      {
        side: 'start',
        icon: 'alert-circle-outline',

```

```

    }
  ]
});
toast.present();
}

async presentAlertConfirm(estado: string) {
  const alert = await this.alertController.create({
    cssClass: 'my-custom-class',
    header: 'Confirme',
    message: 'Marcar <strong>' + estado + '</strong>?',
    buttons: [
      {
        text: 'Cancelar',
        role: 'cancel',
        cssClass: 'secondary',
        handler: (blah) => {
          console.log('Confirm Cancel: blah');
        }
      }, {
        text: 'Aceptar',
        handler: () => {
          console.log('Confirm Okay');
          this.guardarEntradaSalida(estado);
        }
      }
    ]
  });
  await alert.present();
}

guardarEntradaSalida(estado1: string) {
  const data = {
    direccion: this.address,
    fecha: new Date(),
    estado: estado1
  };
  this.dbDataUser.guardarEntradaSalida(this.user.clave, data);
  console.log('se guardo', data);
}

async presentModalRecorrido() {
  const modal = await this.modalController.create({
    component: RecorridoConductorPage,
    cssClass: 'my-custom-class',
    componentProps: {clave: this.user.clave}
  });
  return await modal.present();
}

```



```

}

salir(){
  this.ubicacionservice.detenerUbicacion();
  this.usuarioservice.borrarUsuario();
  this.router.navigate(['/login-conductor']);
}

```

Figura 93: mapa-conductor.page.ts

Fuente: Elaboración propia

Contenido del archivo perfil-conductor.page.html

```

<ion-content>
  <ion-fab vertical="bottom" horizontal="start" slot="fixed">
    <ion-button color="tertiary" shape="round" routerLink="/mapa-
conductor" routerDirection="root">
      <p>Posición actual</p>
    </ion-button>
  </ion-fab>
  <div id="map" #map></div>
</ion-content>

```

Figura 94: perfil-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo perfil-conductor.page.ts

```

export class VehiculosConductorPage implements OnInit {
  conductorcombi: choferes[] = [];
  lat = -13.6619891;
  lng = -73.3790506;
  constructor(public ubicacionservice: UbicacionConductorService,
    public usuarioservice: AuthconductorService,
    private router: Router,
    db: AngularFirestore) {
    // initialize the plugin

    this.ubicacionservice.iniciarGeolocalizacion();
    this.ubicacionservice.inicializarTaxista();
    db.collection('conductores').valueChanges()
      .subscribe((data: choferes[]) => {
        this.conductorcombi = data;
        this.loadMap(this.conductorcombi);
      });
  }
  ngOnInit(){

```

```

}

loadMap(data){
  const myLatLng = {
    lat: this.lat,
    lng: this.lng
  };
  console.log('me muero', data.lat);
  const mapEle: HTMLElement = document.getElementById('map');
  const map = new google.maps.Map(mapEle, {
    center: myLatLng,
    zoom: 13
  });

  for (let m of this.conductorcombi) {
    // console.log('zulma muestrame todas las claves de la coleccion conductor
es', m.clave);
    const myLatLng = new google.maps.LatLng(m.lat , m.lng);
    const marker = new google.maps.Marker({
      position: myLatLng,
      map,
      label: m.nombre,
      title: m.placa,
      icon: '../assets/img/combimarker.png'
    });
  }
}
}

```

Figura 95: perfil-conductor.page.ts

Fuente: Elaboración propia

Contenido del archivo vehiculos-conductor.page.html

```

<ion-content>
  <ion-fab vertical="bottom" horizontal="start" slot="fixed">
    <ion-button color="tertiary" shape="round" routerLink="/mapa-
conductor" routerDirection="root">
      <p>Posición actual</p>
    </ion-button>
  </ion-fab>
  <div id="map" #map></div>
</ion-content>

```

Figura 96: vehiculos-conductor.page.html

Fuente: Elaboración propia

Contenido del archivo vehiculos-conductor.page.ts

```
export class VehiculosConductorPage implements OnInit {
  conductorcombi: choferes[] = [];
  lat = -13.6619891;
  lng = -73.3790506;
  constructor(public ubicationservice: UbicacionConductorService,
    public usuarioservice: AuthconductorService,
    private router: Router,
    db: AngularFirestore) {
    // initialize the plugin

    this.ubicationservice.iniciarGeolocalizacion();
    this.ubicationservice.inicializarTaxista();
    db.collection('conductores').valueChanges()
      .subscribe((data: choferes[]) => {
        this.conductorcombi = data;
        this.loadMap(this.conductorcombi);
      });
  }
  ngOnInit(){
  }

  loadMap(data){
    const myLatLng = {
      lat: this.lat,
      lng: this.lng
    };
    console.log('me muero', data.lat);
    const mapEle: HTMLElement = document.getElementById('map');
    const map = new google.maps.Map(mapEle, {
      center: myLatLng,
      zoom: 13
    });

    for (let m of this.conductorcombi) {
      // console.log('zulma muestrame todas las claves de la coleccion conductor
es', m.clave);
      const myLatLng = new google.maps.LatLng(m.lat , m.lng);
      const marker = new google.maps.Marker({
        position: myLatLng,
        map,
        label: m.nombre,
        title: m.placa,
        icon: '../assets/img/combimarker.png'
      });
    }
  }
}
```

```
}
```

Figura 97: vehiculos-conductor.page.ts

Fuente: Elaboración propia

## Modulo usuario

Contenido del archivo login-usuario.page.html

```
<ion-content>
  <div style="width: 100%; text-align: center;">
    <p><b>Empresa de Transportes Cristo Morado Andahuaylas S.A</b></p>
  >
  
</div>
<form [formGroup]="loginForm">
  <ion-item lines="full">
    <ion-icon color="tertiary" name="mail" slot="start"></ion-icon>
    <ion-label color="tertiary" position="floating">Email</ion-label>
    <ion-input type="text" formControlName="email" required></ion-
  input>
  </ion-item>
  <div *ngIf="email && email.invalid && (email.dirty || email.touched)">
    <div class="mensajeError" *ngIf=" !!email.errors.required">
      * Complete el campo es obligatorio
    </div>
    <div class="mensajeError" *ngIf=" !!email.errors.minlength">
      * Debe tener al menos 3 caracteres
    </div>
  </div>
  <ion-item lines="full">
    <ion-icon color="tertiary" name="lock-closed" slot="start"></ion-
  icon>
    <ion-label color="tertiary" position="floating">Contraseña</ion-
  label>
    <ion-
  input type="password" formControlName="password" required></ion-input>
  </ion-item>
  <div *ngIf="password && password.invalid && (password.dirty || passw
  ord.touched)">
    <div class="mensajeError" *ngIf=" !!password.errors.required">
      * Complete el campo es obligatorio
    </div>
    <div class="mensajeError" *ngIf=" !!password.errors.minlength">
      * Debe tener al menos 3 caracteres
```

```

        </div>
    </div>
    <ion-row>
        <ion-col style="text-align: center;">
            <ion-
button expand="full" color="tertiary" shape="round" type="submit" (click)="
onLogin()">Iniciar Sesion</ion-button>
            <p class="ion-text-center"><a routerLink="/recuperar-
contrasena" routerDirection="root">¿Olvidaste tu contraseña?</a></p>
        </ion-col>
    </ion-row>
    <div style="text-align: center; width: 100%;">
        <span class="divider line one-line"> or </span>
    </div>

    <ion-row>
        <ion-col style="text-align: center;">

            <ion-
button expand="full" shape="round" type="submit" color="danger" (click)="o
nLoginGoogle()">
                <ion-icon name="logo-google"></ion-
icon>&nbsp;&nbsp;&nbsp;Iniciar Sesion con Google</ion-button>
                <p class="ion-text-center"><a routerLink="/registro-
user" routerDirection="root">Registrate</a></p>
            </ion-col>
        </ion-row>
    </form>
</ion-content>

```

Figura 98: login-usuario.page.html

Fuente: Elaboración propia

Contenido del archivo login-usuario.page.ts

```

export class LoginUserPage implements OnInit {
    loginForm: FormGroup;
    constructor(
        private authSvc: AuthuserService,
        private router: Router,
        private ToastController: ToastController,
        private actionSheetController: ActionSheetController) {
        this.loginForm = this.createFormGroupLogin();
    }

    ngOnInit() {
    }
}

```

```

createFormGroupLogin() {
  return new FormGroup({
    email: new FormControl("", [Validators.required, Validators.email]),
    password: new FormControl("", [Validators.required, Validators.minLength
(3)]),
  });
}

get email() { return this.loginForm.get('email'); }
get password() {return this.loginForm.get('password'); }

async onLogin(){
  if (this.loginForm.valid) {
    try{
      const user = await this.authSvc.login(this.loginForm.value.email, this.log
inForm.value.password);
      if (user){
        const isVerified = this.authSvc.verificarEmail(user);
        this.redirectUser(isVerified);
        // console.log('user->', user);
        // this.router.navigate(['admin']);
      }
    }catch (error){
      console.log('Error', error);
    }
  }else{
    console.log('no se pudo registrar');
    this.presentToast('Complete los campos de manera correcta', 'danger');
  }
}

async onLoginGoogle(){
  try{
    const user = await this.authSvc.logonGoogle();
    if (user){
      const isVerificar = this.authSvc.verificarEmail(user);
      console.log('verificar->', isVerificar);
      this.redirectUser(isVerificar);
      // Todo: CheckEmail
      // this.router.navigate(['admin']);
    }
  }catch (error){
    console.log('Error', error);
  }
}

private redirectUser(isVerified: boolean): void {

```

```

if (isVerified) {
  this.router.navigate(['mapa-user']);
} else {
  this.router.navigate(['verificar-email']);
}
}

async presentToast(mensaje, color1) {
  const toast = await this.ToastController.create({
    message: mensaje,
    duration: 3000,
    color: color1,
    buttons: [
      {
        text: 'Aceptar',
        role: 'cancel',
        handler: () => {
        }
      }
    ]
  });
  toast.present();
}

async opcionPerfil() {
  const actionSheet = await this.actionSheetController.create({
    buttons: [{
      text: 'Usuario',
      icon: 'person',
      handler: () => {
        this.router.navigate(['login-user']);
      }
    }, {
      text: 'Conductor',
      icon: 'person',
      handler: () => {
        this.router.navigate(['login-conductor']);
      }
    }, {
      text: 'Administrador',
      icon: 'person',
      handler: () => {
        this.router.navigate(['login-admin']);
      }
    }, {
      text: 'Cancelar',
      icon: 'close',
      role: 'cancel',

```

```

    handler: () => {
      console.log('Cancelado');
    }
  ]
});
await actionSheet.present();
}
}

```

Figura 99: login-usuario.page.ts

Fuente: Elaboración propia

Contenido del archivo mapa-usuario.page.html

```

<ion-header>
  <ion-toolbar color="tertiary">
    <ion-title>admin</ion-title>
    <ion-buttons slot="primary">
      <ion-button routerLink="/perfil-user" routerDirection="root">
        <ion-icon slot="icon-only" name="person-circle-outline"></ion-
icon>
      </ion-button>
      <ion-button (click)="cerrarsesion()">
        <ion-icon slot="icon-only" name="close-circle-outline"></ion-
icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div id="map" #map></div>

</ion-content>

```

Figura 100: mapa-usuario.page.html

Fuente: Elaboración propia

Contenido del archivo mapa-usuario.page.ts

```

export class MapaUserPage implements OnInit {
  lat = -13.6619891;
  lng = -73.3790506;
  conductorcombi: choferes[] = [];
  constructor(db: AngularFireStore,
    private authSvc: AuthuserService,
    private router: Router) {
    db.collection('conductores').valueChanges()

```



```

.subscribe((data: choferes[]) => {
  this.conductorcombi = data;
  this.loadMap(this.conductorcombi);
  console.log('gdfhgjjgk', data);
});
}

ngOnInit() {

}

loadMap(data){
  // console.log('llega', data);
  const myLatLng = {
    lat: this.lat,
    lng: this.lng
  };

  const mapEle: HTMLInputElement = document.getElementById('map');
  const map = new google.maps.Map(mapEle, {
    center: myLatLng,
    zoom: 16
  });
  for (let m of this.conductorcombi) {
    console.log('qqqq', m);
    const myLatLng = new google.maps.LatLng(m.lat , m.lng);
    const marker = new google.maps.Marker({
      position: myLatLng,
      map,
      label: m.nombre,
    });
    console.log('hola', m.lng);

  }

  new google.maps.Marker({
    position: myLatLng,
    map,
    title: "Hello World!"
  });
}

cerrarsesion(){
  console.log('Cerrar Sesión');
  this.authService.cerrarSesion();
}
}

```





```

// tslint:disable-next-line:max-line-length
direccion: new FormControl(),
foto: new FormControl(""),
});
}

subifotoCamara(){
let cameraOptions: CameraOptions = {
quality: 50,
encodingType: this.camera.EncodingType.JPEG,
targetWidth: 500,
targetHeight: 500,
destinationType: this.camera.DestinationType.DATA_URL,
sourceType: this.camera.PictureSourceType.CAMERA,
correctOrientation: true
};

this.camera.getPicture(cameraOptions).then((imageData) => {
// imageData is a base64 encoded string
this.foto = 'data:image/jpeg;base64,' + imageData;
}, (err) => {
console.log(err);
});
}

subirFotoGaleria(){

let cameraOptions: CameraOptions = {
quality: 50,
encodingType: this.camera.EncodingType.JPEG,
targetWidth: 500,
targetHeight: 500,
destinationType: this.camera.DestinationType.DATA_URL,
sourceType: this.camera.PictureSourceType.PHOTOLIBRARY,
correctOrientation: true
};

this.camera.getPicture(cameraOptions).then((imageData) => {
// imageData is a base64 encoded string
this.foto = 'data:image/jpeg;base64,' + imageData;
}, (err) => {
console.log(err);
});
}

actualizarFoto() {
console.log(this.updateForm.value);
}

```

```

if (this.updateForm.valid) {
  if (isNullOrUndefined(this.foto)) {
    console.log('Selecciona una foto');
    this.updateForm.removeControl('foto');
  } else {
    this.updateForm.value.foto = this.foto;
  }
  this.dbData.actualizarFoto(this.datosUser.email, this.updateForm.value);
} else {
  console.log('complete los datos de manera correcta');
}
}

```

```

async borrarConductor(foto) {
  const alert = await this.alertController.create({
    cssClass: 'my-custom-class',
    header: 'Confirmar borrado',
    message: '¿Estás seguro de que deseas eliminar al conductor?',
    buttons: [
      {
        text: 'Cancelar',
        role: 'cancel',
        cssClass: 'secondary',
        handler: (blah) => {
          console.log('Confirm Cancel: blah');
        }
      }, {
        text: 'Eliminar',
        handler: () => {
          this.afs.doc(`/conductores/${foto.email}`).delete().then(() => {
            console.log(`conductor eliminado ="${foto.email}"`);
          }).catch(err => {
            console.log(err);
          });
        }
      }
    ]
  });
  await alert.present();
}

```

```

async opcionFoto() {
  const actionSheet = await this.actionSheetController.create({
    cssClass: 'my-custom-class',
    buttons: [{
      text: 'Galería',
      icon: 'images',
      handler: () => {

```

```

    this.subifotoCamara();
  }
}, {
  text: 'Cámara',
  icon: 'camera',
  handler: () => {
    this.subirFotoGaleria();
  }
},
{
  text: 'Eliminar',
  icon: 'trash',
  handler: () => {
    console.log('Eliminar');
  }
}, {
  text: 'Cancel',
  icon: 'close',
  role: 'cancel',
  handler: () => {
    console.log('Cancel clicked');
  }
}]
});
await actionSheet.present();
}

```

Figura 103: perfil-usuario.page.ts

Fuente: Elaboración propia

Contenido del archivo registro-usuario.page.html

```

<ion-content>
  <form [formGroup]="registroForm">
    <ion-item lines="full">
      <ion-label position="floating">Nombre:</ion-label>
      <ion-input formControlName="nombre" type="text" required></ion-
input>
    </ion-item>
    <div *ngIf="nombre && nombre.invalid && (nombre.dirty || nombre.tou
ched)">
      <div class="mensajeError" *ngIf=" !!nombre.errors.required">
        * Complete el campo es obligatorio
      </div>
      <div class="mensajeError" *ngIf=" !!nombre.errors.minlength">
        * Debe tener al menos 3 caracteres
      </div>
    </div>
  </form>

```

```

<ion-item lines="full">
  <ion-label position="floating">Apellidos:</ion-label>
  <ion-input formControlName="apellidos" type="text" required></ion-
input>
</ion-item>
<div *ngIf="apellidos && apellidos.invalid && (apellidos.dirty || apellid
os.touched)">
  <div class="mensajeError" *ngIf=" !!apellidos.errors.required">
    * Complete el campo es obligatorio
  </div>
  <div class="mensajeError" *ngIf=" !!apellidos.errors.minlength">
    * Debe tener al menos 3 caracteres
  </div>
</div>
<ion-item lines="full">
  <ion-label position="floating">DNI:</ion-label>
  <ion-
input formControlName="dni" minlength="8" maxlength="8" required></ion-
input>
</ion-item>
<div *ngIf="dni && dni.invalid && (dni.dirty || dni.touched)">
  <div class="mensajeError" *ngIf=" !!dni.errors.required">
    * Complete el campo es obligatorio
  </div>
  <div class="mensajeError" *ngIf=" !!dni.errors.minlength">
    * Debe tener 8 caracteres
  </div>
</div>
<ion-item lines="full">
  <ion-label position="floating">Dirección:</ion-label>
  <ion-input formControlName="direccion" type="text" required></ion-
input>
</ion-item>
<div *ngIf="direccion && direccion.invalid && (direccion.dirty || direcc
ion.touched)">
  <div class="mensajeError" *ngIf=" !!direccion.errors.required">
    * Complete el campo es obligatorio
  </div>
  <div class="mensajeError" *ngIf=" !!direccion.errors.minlength">
    * Debe tener al menos 3 caracteres
  </div>
</div>
<ion-item lines="full">
  <ion-label position="floating">Celular:</ion-label>
  <ion-
input formControlName="celular" minlength="9" maxlength="9" required></i
on-input>
</ion-item>

```

```

<div *ngIf="celular && celular.invalid && (celular.dirty || celular.touche
d)">
  <div class="mensajeError" *ngIf=" !!celular.errors.required">
    * Complete el campo es obligatorio
  </div>
  <div class="mensajeError" *ngIf=" !!celular.errors.minlength">
    * Debe tener 9 caracteres
  </div>
</div>
<ion-item lines="full">
  <ion-label position="floating">Email:</ion-label>
  <ion-input type="text" formControlName="email" required></ion-
input>
</ion-item>
<div *ngIf="email && email.invalid && (email.dirty || email.touched)">
  <div class="mensajeError" *ngIf=" !!email.errors.required">
    * Complete el campo es obligatorio
  </div>
  <div class="mensajeError" *ngIf=" !!email.errors.minlength">
    * Debe tener al menos 3 caracteres
  </div>
</div>
<ion-item lines="full">
  <ion-label position="floating">Contraseña:</ion-label>
  <ion-
input type="password" formControlName="password" required></ion-input>
</ion-item>
<div *ngIf="password && password.invalid && (password.dirty || passw
ord.touched)">
  <div class="mensajeError" *ngIf=" !!password.errors.required">
    * Complete el campo es obligatorio
  </div>
  <div class="mensajeError" *ngIf=" !!password.errors.minlength">
    * Debe tener al menos 3 caracteres
  </div>
</div>
<ion-row>
  <ion-col style="text-align: center;">
    <ion-
button expand="full" color="tertiary" shape="round" type="submit" (click)="
onRegistro()">Registrar</ion-button>
  </ion-col>
</ion-row>
</form>
</ion-content>

```

Figura 104: registro-usuario.page.html

Fuente: Elaboración propia



Contenido del archivo registro-usuario.page. ts

```
export class RegistroUserPage implements OnInit {
  registroForm: FormGroup; // slider 8
  public user$: Observable<User>;

  constructor(private authSvc: AuthuserService,
    private afs: AngularFirestore,
    private toastcontroller: ToastController,
    private router: Router) {
    this.registroForm = this.createFormGroupRegistro();
  }

  ngOnInit() {
  }

  createFormGroupRegistro() {
    return new FormGroup({
      // tslint:disable-next-line:max-line-length
      nombre: new FormControl("", [Validators.required, Validators.minLength(
3)]),
      apellidos: new FormControl("", [Validators.required, Validators.minLength
(3)]),
      dni: new FormControl("", [Validators.required, Validators.minLength(8)]),
      direccion: new FormControl("", [Validators.required, Validators.minLength
(3)]),
      celular: new FormControl("", [Validators.required, Validators.minLength(9
)]),
      email: new FormControl("", [Validators.required, Validators.email]),
      password: new FormControl("", [Validators.required, Validators.minLength
(3)]),
      foto: new FormControl(""),
    });
  }

  get nombre() { return this.registroForm.get('nombre'); }
  get apellidos() { return this.registroForm.get('apellidos'); }
  get dni() { return this.registroForm.get('dni'); }
  get direccion() { return this.registroForm.get('direccion'); }
  get celular() { return this.registroForm.get('celular'); }
  get email() { return this.registroForm.get('email'); }
  get password() { return this.registroForm.get('password'); }
  get foto() { return this.registroForm.get('foto'); }

  async onRegistro(){
    console.log('infor', this.registroForm.value);
  }
}
```

```

if (this.registroForm.valid) {
  try{
    const user = await this.authSvc.register(this.registroForm.value.email, thi
s.registroForm.value.password);
    if (user){
      this.guardarUsuario();
      const isVerified = this.authSvc.verificarEmail(user);
      this.redirectUser(isVerified);
      // console.log('user->', user);
      // this.router.navigate(['admin']);
    }
  }catch (error){
    console.log('Error', error);
  }
}else{
  console.log('no se pudo registrar');
  this.presentToast('Complete los campos de manera correcta', 'danger');
}
}

private redirectUser(isVerified: boolean): void {
  if (isVerified) {
    this.router.navigate(['mapa-user']);
  } else {
    this.router.navigate(['verificar-email']);
  }
}

private guardarUsuario(){
  if (this.registroForm.valid){
    this.afs.collection('usuarios').doc(this.registroForm.value.email).set(this.re
gistroForm.value);
  }else{
    console.log('no se pudo guardar el registro');
  }
}

async presentToast(mensaje, color1) {
  const toast = await this.toastcontroller.create({
    message: mensaje,
    duration: 3000,
    color: color1,
    buttons: [
      {
        text: 'Aceptar',
        role: 'cancel',
        handler: () => {

```

```

    }
  }
]
});
toast.present();
}
}

```

Figura 105: registro-usuario.page.ts

Fuente: Elaboración propia

Contenido del archivo verificar-email.page.html

```

<ion-content *ngIf="user$ | async as user">
  <ion-row>
    <ion-col>
      <h2>¡Gracias por registrarse! </h2>
      <p>Enviamos un correo electrónico de confirmación a <strong>{{user.
email}}</strong></p>
      <p>Revise su correo electrónico y haga clic en el enlace para verificar
su dirección de correo electrónico.</p>
    </ion-col>
  </ion-row>

  <ion-row>
    <ion-col>
      <ion-
button type="submit" color="tertiary" expand="block" (click)="onSendEmail(
)">Reenviar la verification</ion-button>
    </ion-col>
  </ion-row>
</ion-content>

```

Figura 106: verificar-email.page.html

Fuente: Elaboración propia

Contenido del archivo verificar-email.page.ts

```

export class VerificarEmailPage implements OnInit {
  user$: Observable<User> = this.authSvc.afAuth.user;

  constructor(private authSvc: AuthuserService) {}

  ngOnInit() {
  }
}

```

```
async onSendEmail(): Promise<void> {
  try {
    await this.authSvc.sentVerificationEmail();
  } catch (error) {
    console.log('Error->', error);
  }
}

ngOnDestroy(): void {
  this.authSvc.logout();
}
}
```

Figura 107: verificar-email.page.ts

Fuente: Elaboración propia

### 3.6.5. Fase pruebas

En esta última fase de la metodología Mobile-D, se realiza un conjunto de pruebas de la aplicación móvil, se identifica y se soluciona algunos errores, esta fase busca entregar una versión completamente estable y funcional de la aplicación móvil. Una vez que se solucionaron todos los errores identificados, recién se puede obtener el entregable final estable. Al terminar el desarrollo de los módulos de la aplicación móvil se pasó a esta fase de pruebas, donde se iteró, para obtener un módulo funcional y al final de todo se llegó a la versión estable.

## CONCLUSIONES

El objetivo de esta tesis era desarrollar un sistema de geolocalización en tiempo real para determinar la ubicación de todos vehículos de la empresa de transporte público urbano Cristo Morado de Andahuaylas, de esta manera solucionar los problemas de tráfico en la ciudad de Andahuaylas, logrando cumplir todos los objetivos planteados:

- Se ha diseñado y desarrollado 3 módulos para el administrador, conductor y usuario con interfaces amigables y sencillas de usar, cuya funcionalidad es permitir a los vehículos de la empresa de transporte público urbano Cristo Morado, el orden de tránsito en sus respectivas rutas.
- Se ha desarrollado el módulo de ubicación en tiempo real para todos los vehículos en servicio de la empresa de transporte público urbano Cristo Morado, de esa manera tanto los usuarios y administrador podrá monitorear o visualizar en tiempo real a todos vehículos en servicio.
- Implementar y desarrollo del módulo de reorganización de todos los vehículos en servicio, enviando alertas cuando un conductor este cerca de su posición actual.
- Se desarrolló el módulo de reporte de salida y llegada de los vehículos en los paraderos iniciales y finales en las rutas A, B y C en la empresa de trasportes cristo morado, esto ayudara a responsable a verificar el cumplimiento del recorrido por las rutas a cada vehículo.

## RECOMENDACIONES

A lo largo de la investigación se logró identificar varios puntos clave que se deben tomar en cuenta para este tipo de investigación.

- En los módulos del proyecto, el uso de cada plugins y dependencias instaladas tienen que ser probadas y solucionadas en el transcurso del desarrollo, ya que estos generan errores y conflictos que pueda presentar a lo largo del desarrollo, la cual se recomienda solucionarlo para no tener problemas en la exportación final del proyecto.
- Para habilitar los permisos eficientes en el uso de la geolocalización en tiempo real en los sistemas operativos Android y IOS aparte del plugin (Geolocation), se recomienda la instalación de dos plugin alternativos (AndroidPermissions, LocationAccuracy), los plugin mencionados permiten que los smartphones activen su GPS de manera automática, esto ayudará que los conductores no dificulten al momento de que el sistema operativo solicite el uso del GPS.
- Se recomienda realizar una copia de seguridad del código fuente del proyecto para posteriores actualizaciones o si ha estado añadiendo o eliminando plugins y esta genere errores permanentes de la aplicación ya sea por conflictos internos de los plugin y dependencias.
- Mantener actualizado las versiones de IONIC, Angular, Firebase. Debido a que las versiones están disponibles y constantemente actualizados por los desarrolladores de los mismos.

## Bibliografía

- Altamirano Ochicua, C. (2018). Aplicación móvil de gestión de infracciones de tránsito georeferenciadas para inspectores de tránsito de la Municipalidad Provincial de Andahuaylas.
- Amazon Web Services. (22 de julio de 2020). *¿Qué es NoSQL?* Obtenido de aws: <https://aws.amazon.com/es/nosql/>
- Baldoceda Chavez, J. C. (2017). Desarrollo de un aplicativo móvil basado en la metodología mobile-D para la gestión de reservas del hotel Caribe de Huaral. 46-53.
- Chicaiza Palate, D. B. (2020). Desarrollo de una aplicación móvil híbrida e-commerce para la gestión de ventas de la empresa Calzado Anabel (Bachelor's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera Ingeniería en Sistemas Co. *google académico*. Ambato, Ecuador.
- Claudia, G. (24 de Noviembre de 2017). *¿Aplicación híbrida o nativa? Usos, ventajas y características.* Obtenido de MEGAWEB: <http://www.megaweb.com.mx/Noticia/Aplicacion-hibrida-o-nativa-Usos-ventajas-y-caracteristicas>
- colaboradores de Wikipedia. (25 de mayo de 2020). *Angular (framework)*. Obtenido de Wikipedia, La enciclopedia libre: [https://es.wikipedia.org/w/index.php?title=Angular\\_\(framework\)&oldid=126367011](https://es.wikipedia.org/w/index.php?title=Angular_(framework)&oldid=126367011)
- colaboradores de Wikipedia. (1 de septiembre de 2020). *Wikipedia*. (L. e. Wikipedia, Editor) Obtenido de <https://es.wikipedia.org/w/index.php?title=Transporte&oldid=128926995>
- Content, R. (24 de julio de 2019). *¿Qué es CSS y cuáles son sus funciones en Internet?* Obtenido de <https://rockcontent.com/es/blog/que-es-css/>
- Delía, L. N. (2017). Delía, L. N. (2017). Desarrollo de aplicaciones móviles multiplataforma (Doctoral dissertation, Facultad de Informática). *google académico*, (pág. 84). La plata.
- Desarrolladores web. (22 de julio de 2020). *Descubriendo HTML*. Obtenido de Desarrolladores web: <https://desarrolladoresweb.org/html/que-es-html/>
- Espinoza Choque, C. A. (2019). Espinoza Choque, C. A. (2019). Desarrollo de un aplicativo móvil para el monitoreo de trabajadores utilizando la tecnología GPS móvil en la Universidad Peruana Unión Filial–Juliaca. (pág. 82). Juliaca: google académico.
- Griffith, C. (22 de julio de 2020). *¿Qué es Apache Cordova?* doi:<https://ionicframework.com/resources/articles/what-is-apache-cordova#summary>

- Guzman Dias, N. R., & Chaparro Arizaga, M. L. (2017). Desarrollo de Aplicación Móvil de Transporte entre la Comunidad Universitaria con Capacidad de Geolocalización para el Proyecto Ud Sobre Ruedas.
- Hernandez Vasquez, A. (01 de junio de 2018). *Mobile-D*. Obtenido de Prezi: <https://prezi.com/p/ilesdkrksv5e/mobile-d/>
- Lucas, J. (04 de septiembre de 2019). *Qué es NodeJS y para qué sirve*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-nodejs/>
- Méndez, D. (17 de abril de 2020). *Numdea*. Obtenido de <https://numdea.com/servicios.html>
- Meza Romero, J. C., & Leño Pariona, V. G. (2018). Sistema de monitoreo de una red de buses de transporte público e información para usuarios empleando transceptores GPS/GSM. 23. Obtenido de [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/9781/MEZA\\_JAVIER\\_MONITOREO\\_BUSES\\_TRANSPORTE\\_P%3%9ABLICO\\_TRANSCEPTORES.pdf?sequence=1](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/9781/MEZA_JAVIER_MONITOREO_BUSES_TRANSPORTE_P%3%9ABLICO_TRANSCEPTORES.pdf?sequence=1)
- Milenium. (22 de julio de 2020). *Internet*. Obtenido de Milenium: <https://www.informaticamilenium.com.mx/es/temas/que-es-internet.html>
- mobincude. (23 de junio de 2020). *App Nativas y Web Apps*. Obtenido de mobincude Support: <https://mobincube.zendesk.com/hc/es/articles/206839903-App-Nativas-y-Web-Apps>
- Müller, M., Charypar, D., & Gross, M. (2003). Particle Based Fluid Simulation for Interactive Applications. *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*.
- mySolutions. (20 de julio de 2018). *¿Qué es Ionic?* Obtenido de mySolutions Digital Factory: <https://mysolutions.cl/que-es-ionic/>
- Ordoñez Marín, C. A. (2019). Desarrollo de una aplicación web para el control y monitoreo en tiempo real que permita mejorar el servicio del transporte interno de la Universidad Nacional de Cajamarca. 18.
- Orlando, Y. (27 de 10 de 2016). *Apps Nativas VS Híbridas: Ventajas y Desventajas*. Obtenido de VEX SOLUCIONES: <https://www.vexsoluciones.com/apps-moviles/apps-nativas-vs-hibridas/>
- Pérez cardona, M. (14 de Octubre de 2016). *Firestore, qué es y para qué sirve la plataforma de Google*. Obtenido de IEBS: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Pino Reyes, J. (13 de junio de 2018). *¿Qué es HTML?* Obtenido de DevCode: <https://devcode.la/blog/que-es-html/>



- QUALITY DEVS. (16 de septiembre de 2019). *¿Qué es Angular y para qué sirve?* Obtenido de QUALITY DEVS: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>
- Quintero Caicedo, R. G. (2015). Aplicación web para la gestión de ventas en la empresa line technology de la ciudad de Ibarra (Bachelor's thesis). (pág. 86). Ibarra, Ecuador: google académico.
- Ramírez Vique, R. (2019). Métodos para el desarrollo de aplicaciones móviles. *Universidad Oberta de Catalunya*. Obtenido de [http://190.57.147.202:90/xmlui/bitstream/handle/123456789/464/Tecnologia\\_y\\_desarrollo\\_en\\_dispositivos\\_moviles.pdf?sequence=1](http://190.57.147.202:90/xmlui/bitstream/handle/123456789/464/Tecnologia_y_desarrollo_en_dispositivos_moviles.pdf?sequence=1)
- Rodríguez, T. (30 de septiembre de 2011). *Métodos aplicables para el desarrollo de aplicaciones móviles*. Obtenido de GENBETA: <https://www.genbeta.com/desarrollo/metodos-aplicables-para-el-desarrollo-de-aplicaciones-moviles>
- ROMERO, J. C., VALERIO, O. L., & FERREIRA, G. S. (2019). APLICACIONES HÍBRIDAS PARA DISPOSITIVOS MÓVILES COMO HERRAMIENTA EN EL SECTOR EMPRESARIAL. IMPLEMENTACIÓN DE SIFÓN HIDRÁULICO SAN MATEO EN EL SISTEMA DE RIEGO TEPETITLÁN ESTADO DE MÉXICO, GENERANDO SUSTENTABILIDAD Y APROVECHAMIENTO DEL CICLO PRODUCTIVO.,. *CIENCIA ADMINISTRATIV*, 48-49.
- Sabini, J. L. (13 de octubre de 2019). *Estudio desde Casa*. Obtenido de ¿Qué es un teléfono celular inteligente?: [https://estudiodesdecasa.com.ar/que-es-un-telefono-celular-inteligente/#Definicion\\_de\\_Telefono\\_Inteligente\\_o\\_Smartphone](https://estudiodesdecasa.com.ar/que-es-un-telefono-celular-inteligente/#Definicion_de_Telefono_Inteligente_o_Smartphone)
- SendPulse. (17 de 12 de 2018). *Que es Geolocalización - Significado*. Obtenido de SendPulse: <https://sendpulse.com/latam/support/glossary/geolocation>
- Solenthaler, B., Schläfli, J., & Pajarola, R. (2007). A Unified Particle Model for Fluid–Solid Interactions: Research Article. *Computer Animation and Virtual Worlds*.
- Takahiro, H., Seiichi, K., & Yoichiro, K. (2007). Smoothed Particle Hydrodynamics in Complex Shapes. *Proceedings of the 23rd Spring Conference on Computer Graphics*. Budmerice, Slovakia.
- Van Dokkum, K., Koomen, K., Ten Katen, H., & Pinkster, J. (2008). *Ship Stability, 3rd edition*. DokMar: Enkhuizen.

Wikipedia. (24 de febrero de 2020). *Tiempo real*. Obtenido de Wikipedia, La enciclopedia libre.: [https://es.wikipedia.org/w/index.php?title=Tiempo\\_real&oldid=123787400](https://es.wikipedia.org/w/index.php?title=Tiempo_real&oldid=123787400)

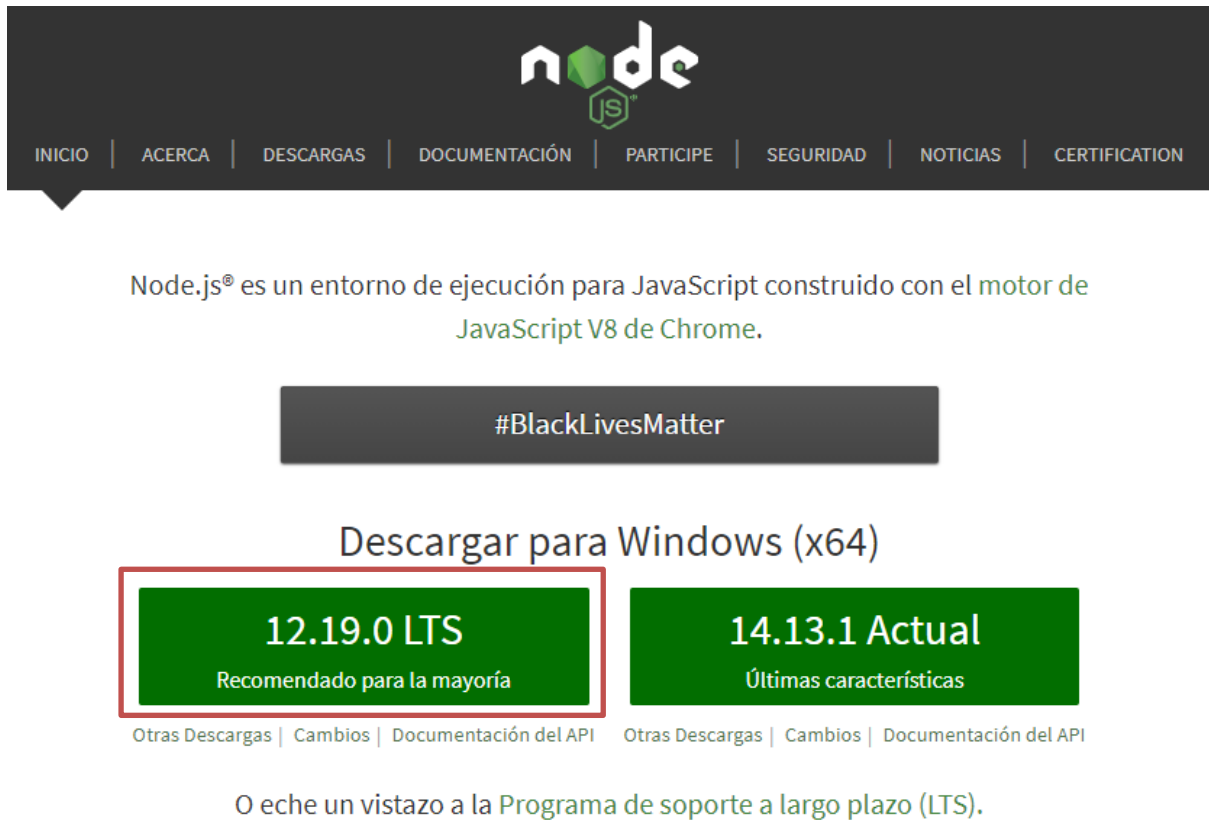
Yago Marco , E. (2015). Aplicación Web y Móvil para el seguimiento de autobuses escolares (Doctoral dissertation). 3.

## ANEXOS

### Manual del instalación y configuración de IONIC:

Descargar Node.js del siguiente link de acuerdo al tipo de Sistema Operativo:

<https://nodejs.org/es/>



node JS

INICIO | ACERCA | DESCARGAS | DOCUMENTACIÓN | PARTICIPE | SEGURIDAD | NOTICIAS | CERTIFICATION

Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

#BlackLivesMatter

Descargar para Windows (x64)

12.19.0 LTS  
Recomendado para la mayoría

14.13.1 Actual  
Últimas características

Otras Descargas | Cambios | Documentación del API

O eche un vistazo a la Programa de soporte a largo plazo (LTS).

Figura 108: Pagina de descarga de Node.Js

Fuente: Página oficial de Node.Js

Como siguiente paso procedemos a instalar Node.Js

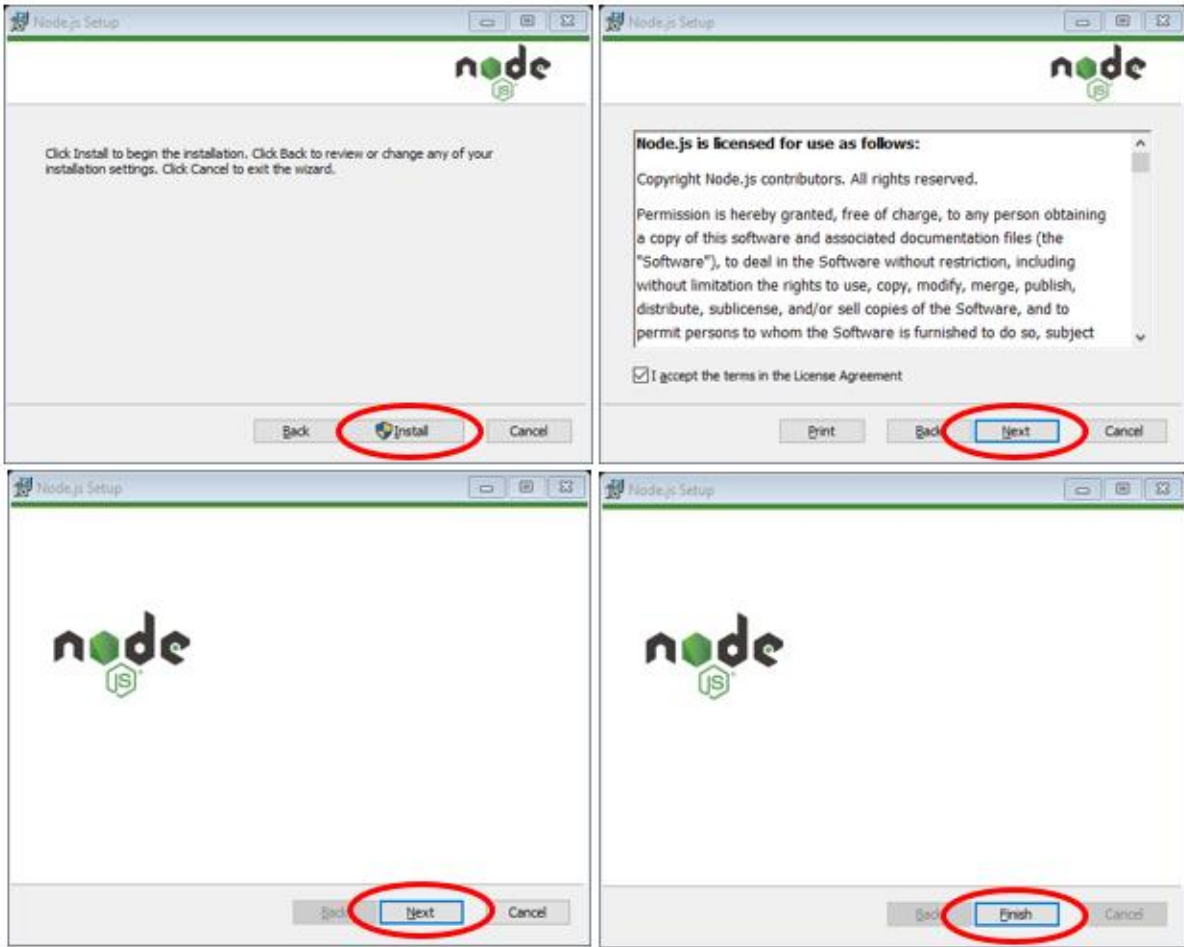


Figura 109: Pagina de descarga de Node.Js

Fuente: Elaboración propia

### Seguidamente instalamos Visual Code.

Descargamos el instalador desde la página oficial de visual code.

<https://code.visualstudio.com/download>

# Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

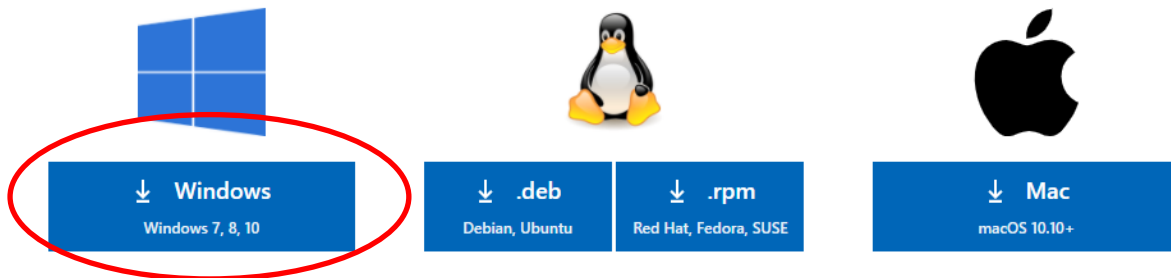


Figura 110: Descarga de Visual Code

Fuente: Página oficial de Visual Code.

Como siguiente paso instalaremos Visual Code.

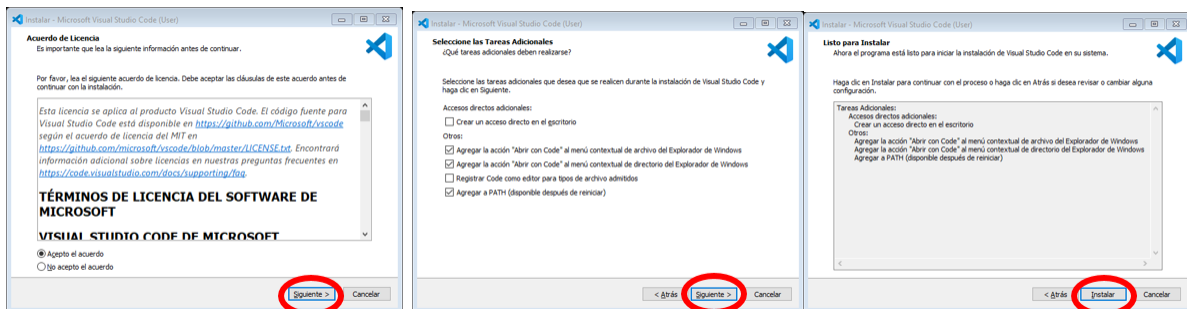


Figura 111: Instalación de visual code.

Fuente: Elaboración propia.

Una vez instalado ejecutaremos visual code.



Figura 112: Icono de Visual code.

Fuente: Elaboración propia.

Tendremos una ventana de Visual code y presionaremos Ctrl + ñ para abrir el terminal.

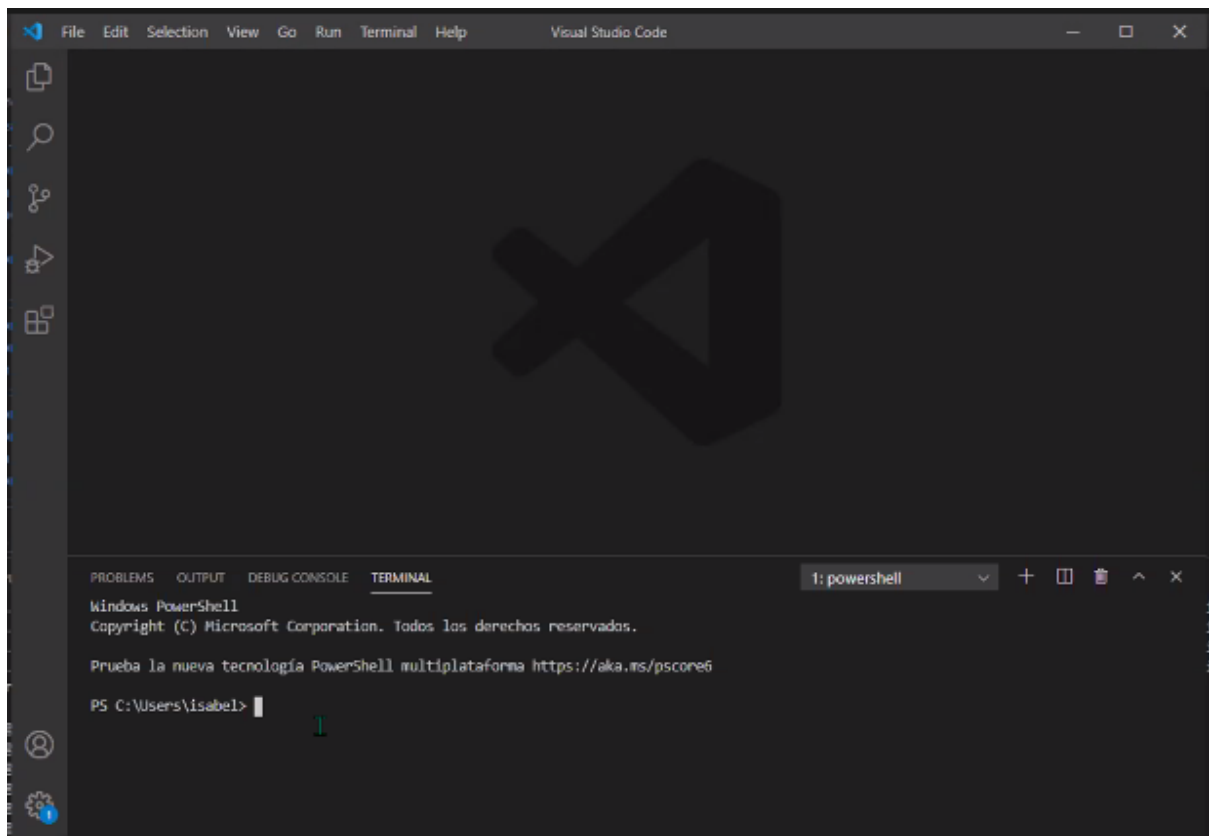


Figura 113: terminal en Visual code.

Fuente: Elaboración propia.

A continuación, ejecutamos los siguientes comandos para la instalación de cordova:

```
npm install -g cordova
PS C:\Users\isabel> npm install -g cordova
[.....] \ fetchMetadata: WARN remove@obsoleteDep removing server@6.3.0 from the tree as its been replaced by a newer ver[
```

Figura 114: instalación en el terminal de Cordova

Fuente: Elaboración propia.

También ejecutamos el siguiente comando ara la instalación de IONIC.

```
npm install -g ionic
PS C:\Users\isabel> npm install -g ionic
npm WARN deprecated ionic@5.4.16: The Ionic CLI now uses +@ionic/cli + for its package name!
status/1223268498362851338
[.....] | rollbackFailedOptional: WARN deprecated ionic@5.4.16: The Ionic CLI now
```

Figura 115: instalación en el terminal de ionic

Fuente: Elaboración propia.

En seguida iniciaremos un nuevo proyecto en IONIC

```
ionic start appcristomorado cidemenu
```

Figura 116: iniciamos un nuevo proyecto en ionic

Fuente: Elaboración propia.

A continuación, se muestran los manuales de los 3 módulos de la aplicación UBIMOBILE.

- **Manual del Administrador:**

**Iniciar sesión:** Permite al administrador autenticarse para posteriormente pueda visualizar en tiempo real a los vehículos en servicio.

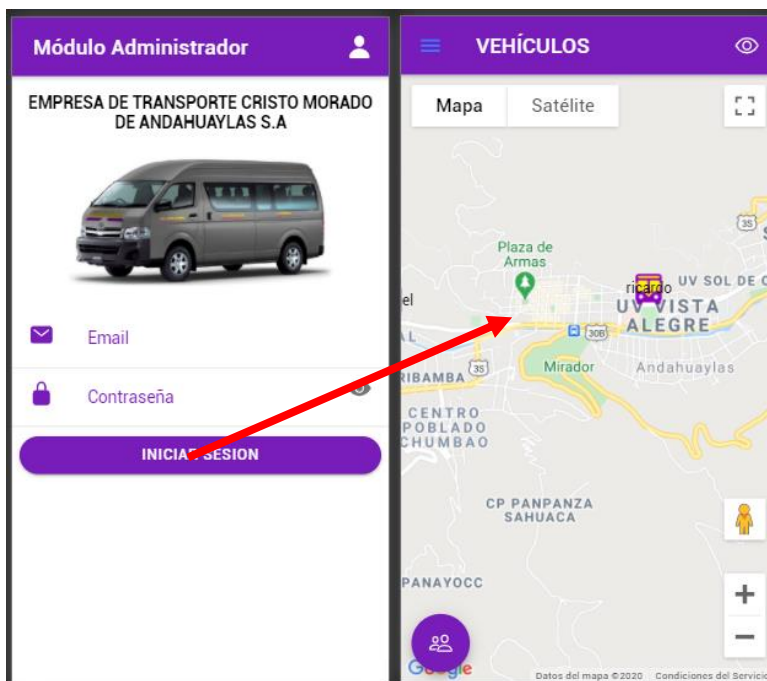


Figura 117: inicio de sesión de la aplicación

Fuente: Elaboración propia.

**Menu:** Permite al administrador acceder a las diferentes páginas en la aplicación.

**Perfil:** Permite mostrar los datos personales del administrador y actualizar la foto de perfil del administrador.

**Registro administrador:** Permite crear a un nuevo administrador.

**Agregar conductor:** Permite crear un nuevo conductor.

**Lista de conductor:** Permite mostrar a todos los conductores en una lista.

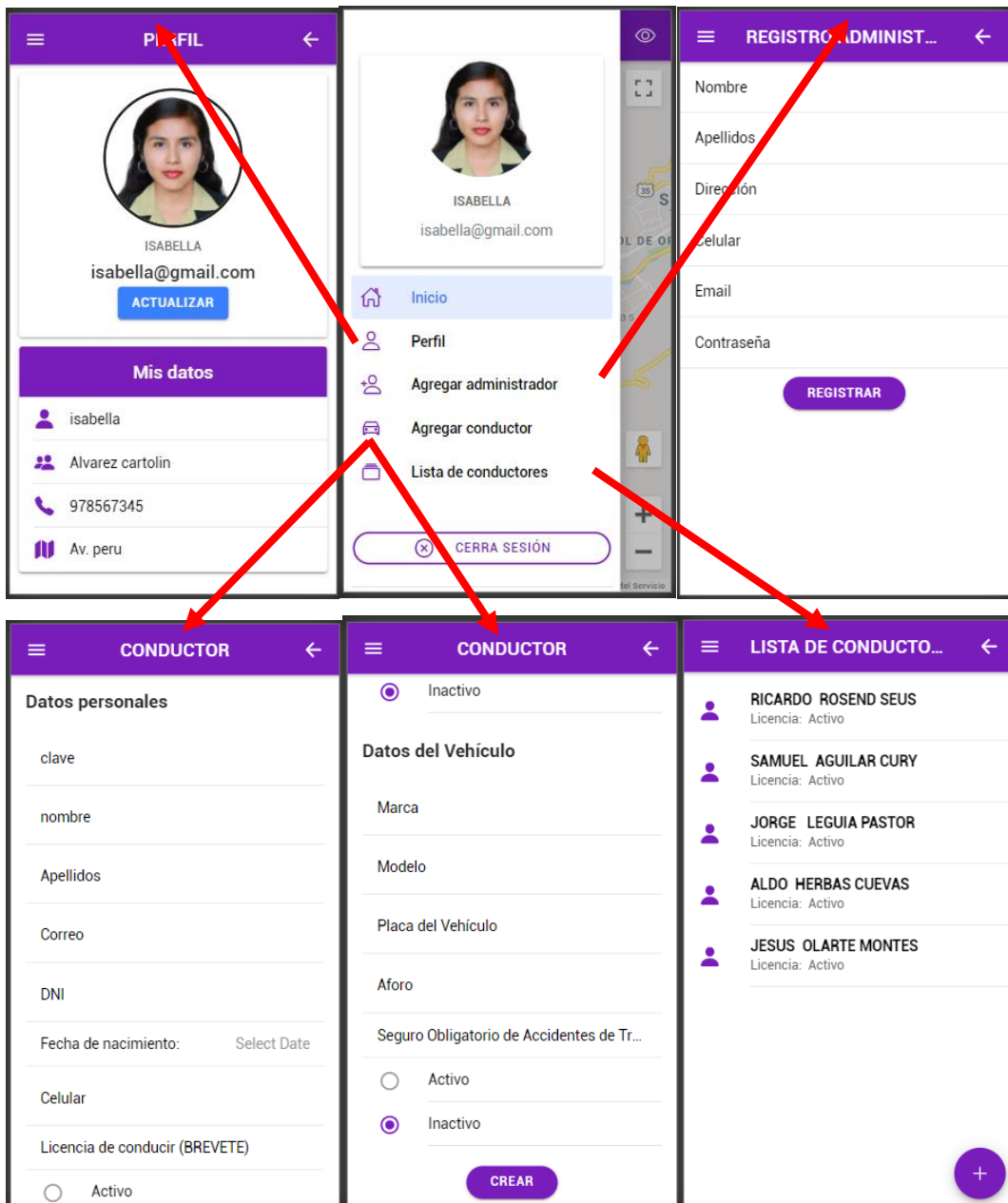


Figura 118: menú y sus respectivas rutas.

Fuente: Elaboración propia.

**Actualizar conductores:** Permite actualizar los datos de los conductores.



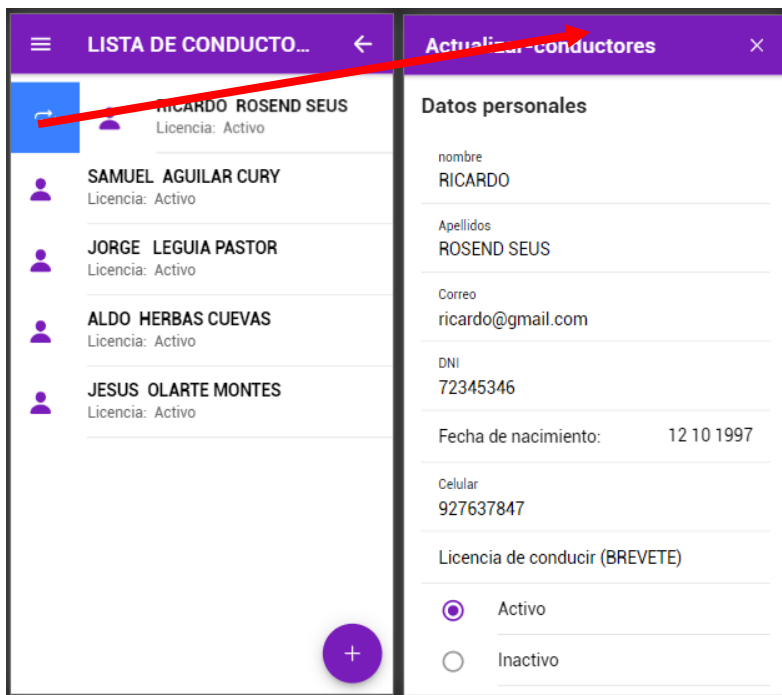


Figura 119: Pagina actualizar conductor.

Fuente: Elaboración propia.

**Informe recorrido:** Permite al administrador visualizar la cantidad de salidas y llegas en sus paraderos finales de los vehículos.

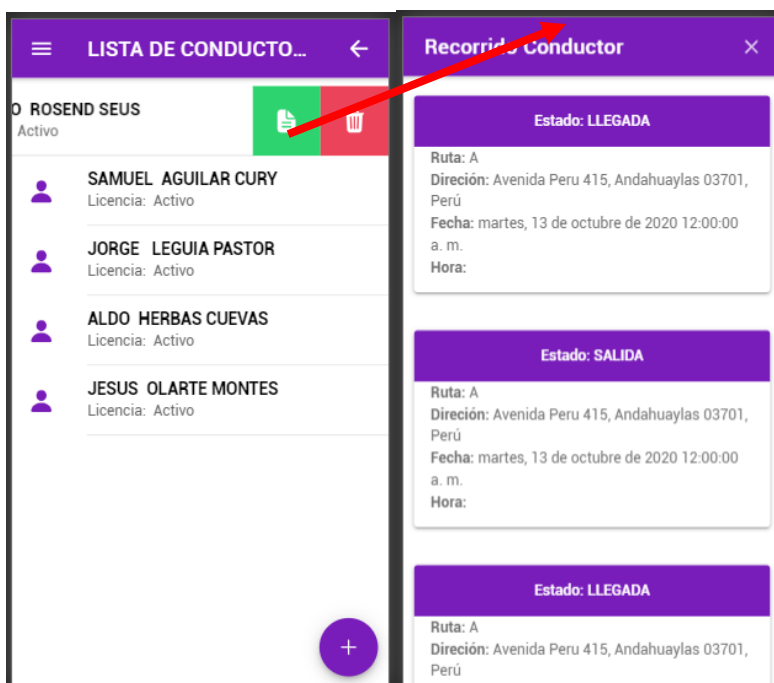
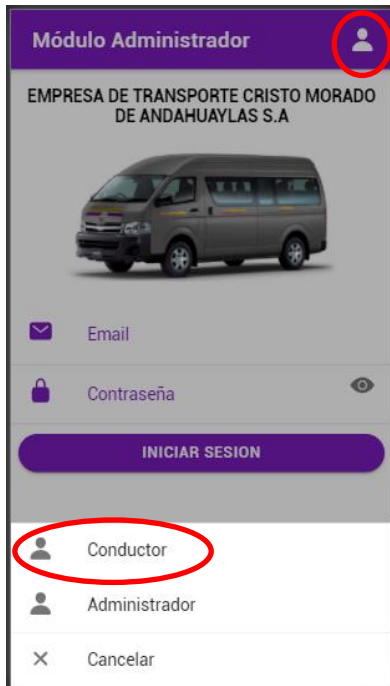


Figura 120: Pagina reporte de salida y llegada de vehículos en os paraderos iniciales y finales.

Fuente: Elaboración propia.

- **Manual de Conductor:**

Acceso del módulo administrados al módulo conductor.



*Figura 121: página de intercambio de pginas inicio sesión.*

Fuente: Elaboración propia.

**Iniciar sesión:** Permite al conductor autenticarse mediante una clave acceder a la geolocalización.

**Geolocalización en tiempo real del vehículo:** visualiza su posición, así enviando sus coordenadas actuales.

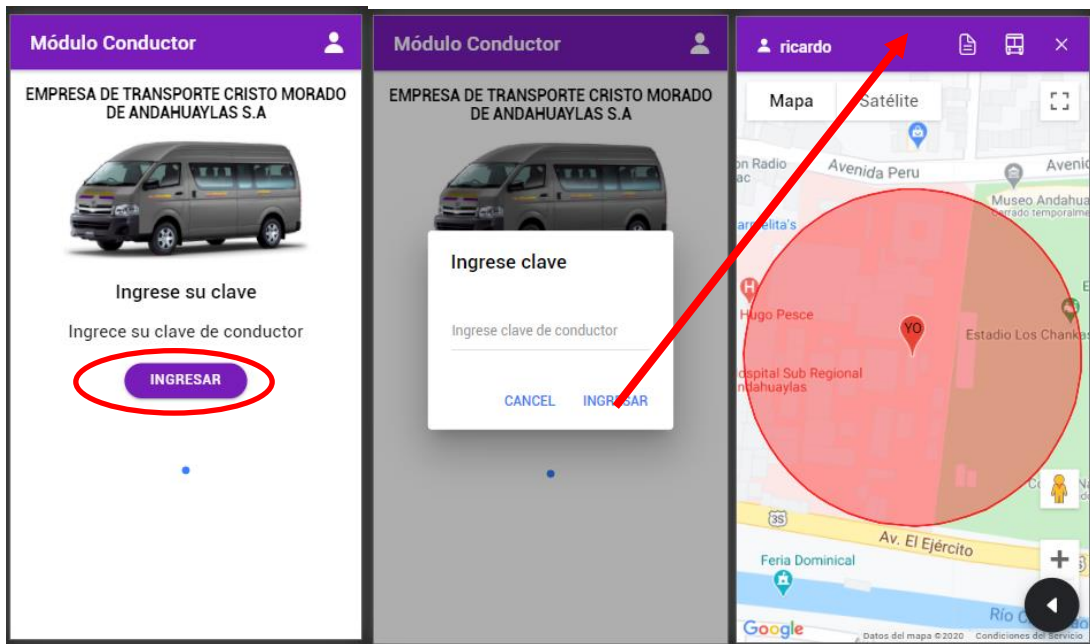


Figura 122: inicio sesión del conductor mediante una clave.

Fuente: Elaboración propia.

**Reporte de salida y llegada:** Permite al conductor visualizar sus reportes de salida y llegada a los paraderos iniciales y finales.

**Reporte de salida:** opción para reportar la salida de vehículo del paradero inicial.

**Reporte de llegada:** opción para reportar la llega de vehículo del paradero final.

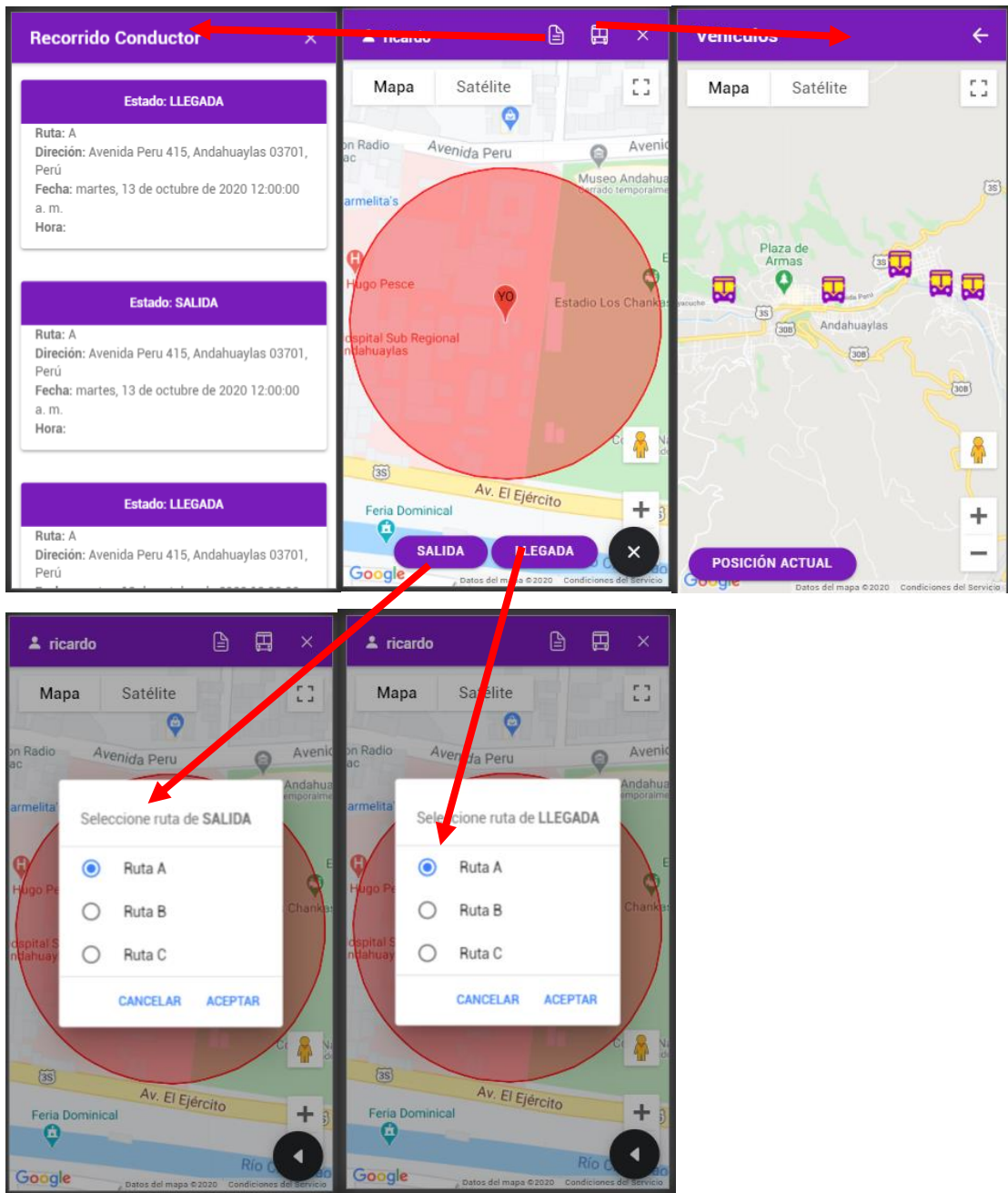


Figura 123: páginas de reporte de entrada de salida y llegada en los paraderos iniciales y finales de cada vehículos en servicio. .

Fuente: Elaboración propia.

**Orden de vehículos en servicio:** orden de vehículos mediante alerta cuando otros vehículos estén dentro de su radio de 100 metros.



Figura 124: orden de vehículos con alerta.

Fuente: Elaboración propia.

- **Manual de Usuario:**

**Iniciar sesión:** Permite al usuario autenticarse y acceder a la página principal donde visualizara a los conductores en tiempo real.

**Recuperar contraseña:** permite al usuario recuperar su contraseña en caso de perderlo.

**Registro:** Permite al usuario registrarse para el uso del servicio.

**Verificar correo:** Permite al usuario verificar su correo mediante una nota de verificación en correo.

**Mapa:** Permite al usuario visualizar la geolocalización en tiempo real a todos los vehículos en servicio.

**Perfil:** muestra los datos personales del usuario.

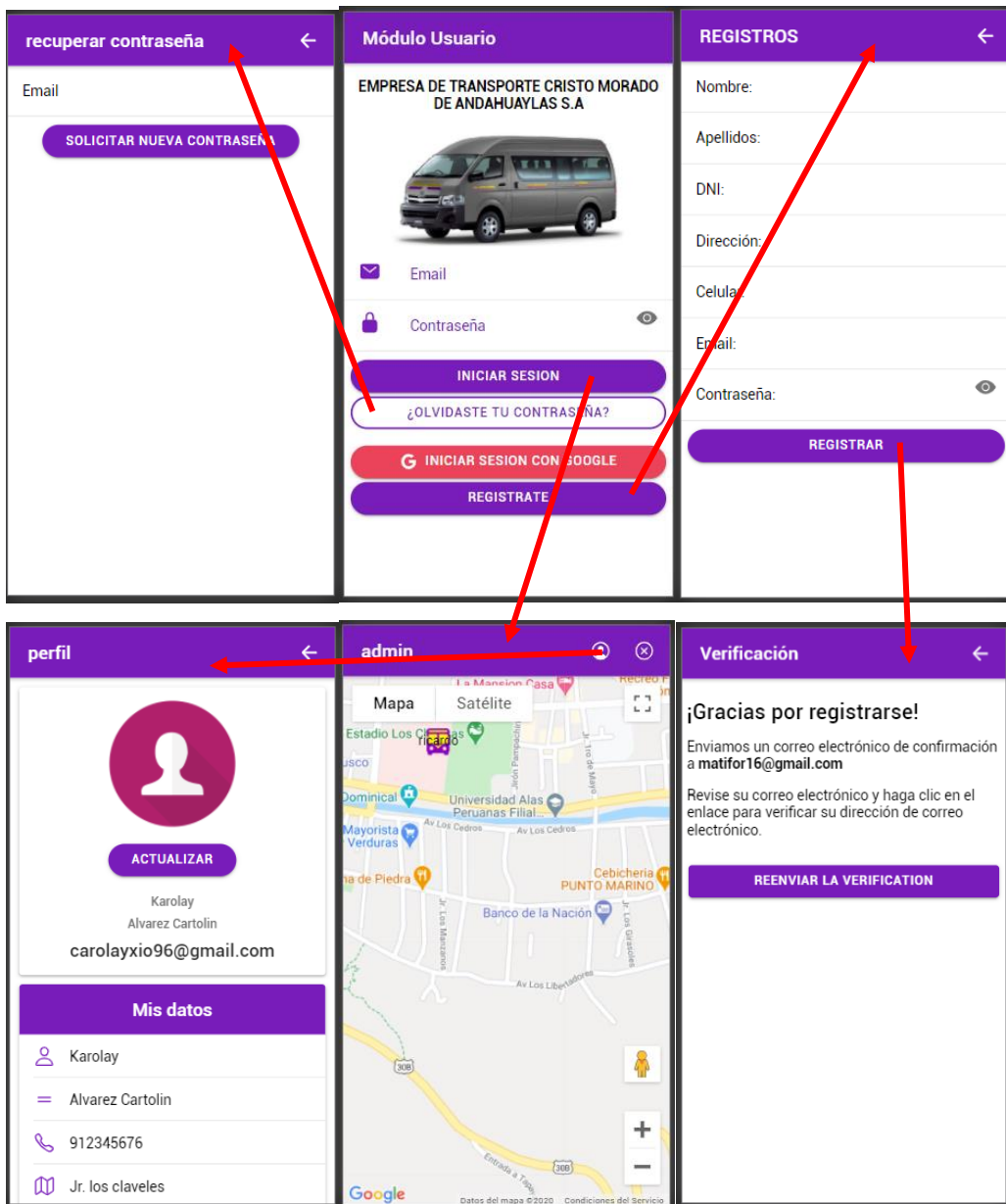


Figura 125: manual del usuario.

Fuente: Elaboración propia.