



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
LICENCIATURA EN CIENCIAS COMPUTACIONALES

TESIS

**APLICACIÓN MÓVIL PARA GESTIÓN DE LÍNEA DE
TRANSPORTE DE CARGA (FREIGHTEM)**

**Para obtener el título de
Licenciado en Ciencias Computacionales**

PRESENTA

Miguel Ángel Pérez Munguía

Directora

Dra. Verónica Martínez Lazcano

Codirectora

M.C.C Iliana Castillo Pérez

Comité tutorial

Dra. Martha Idalid Rivera González (Presidente)

Dra. Claudia Isabel Martínez Alcalá (Secretaria)

Dra. Verónica Martínez Lazcano (Vocal)

M.C.C. Iliana Castillo Pérez (Suplente)

Pachuca de Soto, Hgo., México., enero 2025



Mineral de la Reforma, Hgo., a 10 de enero de 2025

Número de control: ICBI-D/049/2025

Asunto: Autorización de impresión.

**MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH**

Con fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio le comunico que el Jurado asignado al Egresado de la Licenciatura en Ciencias Computacionales **Miguel Ángel Pérez Munguía**, quien presenta el trabajo de titulación "**Aplicación Móvil para Gestión de Línea de Transporte de carga (Freightem)**", después de revisar el trabajo en reunión de Sinodales ha decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: Mtra. Martha Idalid Rivera González

Secretario: Dra. Claudia Isabel Martínez Alcalá

Vocal: Dra. Verónica Martínez Lazcano

Suplente: Mtra. Iliana Castillo Pérez

Sin otro particular por el momento, reciba un cordial saludo.

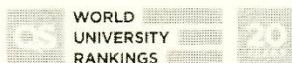
Atentamente
"Amor, Orden y Progreso"

Mtro. Gabriel Vergara Rodríguez
Director del ICBI



GVR/YCC

Ciudad del Conocimiento. Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 771 71 720 00 Ext. 40001
direccion_icbi@uaeh.edu.mx,
vergara@uaeh.edu.mx



Dedicatoria

A mis queridos padres. Este logro académico es un reflejo del incansable esfuerzo que han invertido para brindarme una educación sólida. Cada sacrificio que han hecho, cada día de trabajo duro y cada decisión que tomaron en mi nombre son el fundamento de mi éxito. Su dedicación y compromiso con mi educación son un regalo que valoro más allá de las palabras. Esta tesis es un testimonio de su sacrificio y amor, y me llena de orgullo honrarlos de esta manera. Gracias por ser los faros en mi vida, por iluminar el camino hacia el conocimiento y por inculcarme la importancia del trabajo duro y la educación. Los amo profundamente.

Agradecimientos

En primer lugar, deseo expresar mi más profundo agradecimiento a mis directores de tesis, cuya guía y apoyo han sido fundamentales para la realización de este trabajo. Su paciencia, conocimientos y disposición para orientarme en cada etapa de este proceso fueron claves para alcanzar este logro académico. Gracias por compartir su experiencia y sabiduría, que no solo enriquecieron este trabajo, sino también mi formación personal y profesional.

A mis padres, les agradezco de todo corazón por su amor incondicional y su respaldo constante a lo largo de mi estancia en la universidad. Su apoyo emocional, financiero y moral fue una fuente inagotable de fortaleza que me permitió superar los desafíos de este camino. Su confianza en mis capacidades ha sido el motor que me impulsó a seguir adelante, incluso en los momentos más difíciles.

A mis tíos, les expreso mi sincero agradecimiento por haber sido una segunda familia durante mi etapa universitaria. Su apoyo, consejos y cercanía hicieron que este trayecto fuera más llevadero y me brindaron un hogar fuera de casa. Su generosidad y cariño me acompañaron y motivaron durante esta etapa crucial de mi vida.

Asimismo, extendo mi gratitud a todas aquellas personas que, de manera directa o indirecta, contribuyeron al desarrollo de este trabajo. A mis amigos y compañeros, por su amistad y apoyo en los momentos de duda; a mis profesores, por transmitir sus conocimientos con dedicación y compromiso; y a todos quienes creyeron en mí y me impulsaron a alcanzar mis metas.

Este logro no habría sido posible sin cada uno de ustedes. Gracias por formar parte de este importante capítulo de mi vida.

Resumen

El presente proyecto nace de la observación de la problemática que dificulta la gestión de empresas dedicadas al autotransporte, siendo el sector de la logística de transporte uno de los sectores sociales que ha sido imprescindible para el desarrollo de la sociedad, debido a que este sector se centra en el transporte de bienes. De manera más específica, en el ámbito de la gestión, se enfrentan cotidianamente diversos desafíos que requieren tomar decisiones estratégicas, especialmente en lugares remotos como las carreteras. En este contexto, quienes gestionan las líneas de transporte necesitan acceder fácilmente a la información de forma remota.

Por esta razón, se desarrolló una aplicación que da solución a las dificultades mencionadas, mediante la aplicación de la metodología de desarrollo de software de prototipos evolutivos que se enfoca en la entrega constante de prototipos en periodos cortos con mejoras en cada prototipo. El resultado es una aplicación móvil que integra funciones beneficiosas tanto para el departamento de administración, el de mantenimiento, el de tráfico y el de ventas. Esta aplicación cuenta con una interfaz de usuario intuitiva y un acceso a los datos óptimo y comprensible, impactando así en la gestión de la información de empresas dedicadas a la logística.

Abstract

This project arises from the observation of the challenges that hinder the management of companies engaged in freight transportation. The transportation logistics sector is one of the essential social sectors for the development of society, as it focuses on the transport of goods. More specifically, in the realm of management, various challenges are faced daily that require strategic decision-making, especially in remote areas like highways. In this context, those managing transportation lines need easy remote access to information.

For this reason, an application was developed to address the mentioned difficulties through the application of an evolutionary prototype software development methodology that focuses on the constant delivery of prototypes in short periods with improvements in each version. The result is a mobile application that integrates beneficial functions for the administration, maintenance, traffic, and sales departments. This application features an intuitive user interface and optimal, comprehensible data access, thereby impacting the information management of companies dedicated to logistics.

Contenido

Introducción.....	1
Antecedentes.....	2
Planteamiento del problema	3
Solución propuesta	4
Justificación.....	5
Objetivo general	6
Objetivos específicos.....	6
Metodologías	7
Herramientas de desarrollo.....	11
Alcances.....	12
Limitaciones	13
Estructura del documento	13
Capítulo 1. Marco teórico.....	15
1.1 Términos técnicos.....	15
1.1.1 Bases de datos.....	15
1.1.2 Sistemas gestores de bases de datos	15
1.1.3 Modelo de datos	17
1.1.4 Modelo relacional	18
1.1.5 Reglas de integridad	20
1.1.6 Lenguaje SQL.....	22
1.1.7 Android.....	22
1.1.8 iOS.....	23
1.1.9 IDE	23

1.1.10 Framework.....	23
1.1.11 Flutter	24
1.1.12 Aplicaciones móviles.....	25
1.1.13 Aplicaciones híbridas	26
1.2 Marco conceptual	26
1.2.1 Tractocamión.....	26
1.2.2 Dolly Convertidor.....	27
1.2.3 Remolque.....	28
1.2.4 Logística	28
1.2.5 Activos.....	29
1.2.6 Línea	29
1.3 Estado del arte	29
1.3.1 Fleet Complete.....	30
1.3.2 Geotab.....	31
1.3.3 Sygic	33
Capítulo 2. Metodología.....	37
2.1 Fase 1. Investigación preliminar.....	37
2.2 Fase 2. Definición de los requerimientos del sistema	38
2.3 Fase 3. Diseño técnico	39
2.4 Fase 4. Desarrollo de pruebas.....	43
2.5 Fase 5. Operación y Mantenimiento.....	43
2.5.1 Caso uno	43
2.5.2 Caso dos.....	46
2.5.3 Caso tres	47
Capítulo 3. Desarrollo	50

3.1 Prototipo 1	50
3.1.1 Modelo relacional	51
3.1.2 Diseño de Base de Datos	52
3.1.3 Conexión con la Base de datos	55
3.1.4 Retroalimentación del prototipo	56
3.2 Prototipo 2	56
3.2.1 Retroalimentación del prototipo	58
3.3 Prototipo 3	59
3.3.1 Retroalimentación del prototipo	60
3.4 Prototipo 4	62
3.4.1 Retroalimentación del prototipo	63
3.5 Prototipo 5	64
3.5.1 Retroalimentación del prototipo	65
Prototipo 6	66
3.6.1 Retroalimentación del prototipo	68
Prototipo 7	70
3.7.1 Retroalimentación del prototipo	72
Prototipo 8	72
3.8.1 Retroalimentación del prototipo	76
Prototipo 9	77
3.9.1 Retroalimentación del prototipo	82
Prototipo 10	82
3.10.1 Retroalimentación del prototipo	89
Prototipo 11	89
3.11.1 Retroalimentación del prototipo	94

Prototipo 12	94
3.12.1 Retroalimentación del prototipo	101
Capítulo 4. Validación.....	103
Conclusiones.....	112
Trabajos futuros.....	114
Referencias	115
Anexo A. Manual de usuario.....	119

Índice de figuras

Figura 1 Ciclo de vida de la metodología de cascada.....	8
Figura 2 Flujo general del proceso Scrum.....	8
Figura 3 Ciclo de vida de un prototipo	10

Capítulo 1

Figura 1.1 Tractocamión Kenworth.....	27
Figura 1.2 Dolly Corpus Trailers.....	27
Figura 1.3 Plataforma estándar Corpus Trailers	28
Figura 1.4 Logo de Fleet Complete	30
Figura 1.5 Representación gráfica de la interfaz de Fleet Complete	31
Figura 1.6 Logo de Geotab	31
Figura 1.7 Representación de la interfaz gráfica de Geotab	32
Figura 1.8 Logo de Sygic.	33
Figura 1.9 Representación gráfica de Sygic	35

Capítulo 2

Figura 2.1 Boceto del diseño inicial de la aplicación	40
Figura 2.2 Propuesta del inicio de la aplicación	41
Figura 2.3 Propuesta del buscador de la aplicación.....	41
Figura 2.4 Compra de tractocamiones	44
Figura 2.5 Registro de las nuevas unidades mediante Freightem	44
Figura 2.6 Registro de nueva plataforma en Freightem.....	45
Figura 2.7 Registro de operador en Freightem	46
Figura 2.8 Consulta de datos para rescate de la unidad 25	47
Figura 2.9 Accidente de la unidad 44	48
Figura 2.10 Consulta de datos de la unidad 44.....	48

Capítulo 3

Figura 3.1 Inicio del primer prototipo	50
Figura 3.2 Ventana de búsqueda	51

Figura 3.3 Modelo relacional de la aplicación móvil Freightem	52
Figura 3.4 Muestra del identificador de las unidades físicas	54
Figura 3.5 Implementación de la dependencia Sqflite.....	55
Figura 3.6 Segundo prototipo del inicio	57
Figura 3.7 Acceso a las ventanas de edición en el prototipo dos.....	58
Figura 3.8 Ventana de edición de los tractos	59
Figura 3.9 Agregación de la dependencia Flutter_launcher_icons	59
Figura 3.10 Prototipo de la ventana de agregar operador	61
Figura 3.11 Prototipo de la ventana de eliminar operador.....	62
Figura 3.12 Prototipo cuatro con la implementación de la llamada al operador.....	63
Figura 3.13 Quinto prototipo en la ventana de iniciar sesión	64
Figura 3.14 Quinto prototipo en la ventana de cambio de aceite.....	65
Figura 3.15 Sexto prototipo en la ventana de crear usuario.....	66
Figura 3.16 Agregación de dependencias para generar reportes pdf	67
Figura 3.17 Prototipo seis en ventana de cambio de aceite con generar reporte	68
Figura 3.18 Prototipo seis en la ventana de compartir el reporte generado	69
Figura 3.19 Reporte generado de los cambios de aceite.....	69
Figura 3.20 Ventana de solicitar refacciones con historial de solicitudes	71
Figura 3.21 Ventana de reasignación de planas.....	72
Figura 3.22 Agregación de dependencias para exportar e importar la base de datos	73
Figura 3.23 Prototipo ocho en la ventana de exportar la base de datos	74
Figura 3.24 Prototipo ocho en la ventana de importar la base de datos.....	75
Figura 3.25 Prototipo ocho en la ventana de solicitudes de refacciones.....	76
Figura 3.26 Prototipo nueva en la ventana para asignar operador	78
Figura 3.27 Prototipo nueve en la ventana de los detalles del operador	79
Figura 3.28 Agregación de la dependencia timezone	80
Figura 3.29 Ventana de agendar viaje	81
Figura 3.30 Prototipo diez en la ventana de los estatus de las unidades.....	84
Figura 3.31 Prototipo diez en la ventana de viajes pendientes	85
Figura 3.32 Diálogo de asignar unidad a viaje	86
Figura 3.33 Prototipo diez en la ventana de viajes activos	87
Figura 3.34 Prototipo diez en la ventana de viajes terminados.....	88
Figura 3.35 Prototipo once en la ventana para registrar bitácoras de mantenimiento	90
Figura 3.36 Prototipo once en la ventana de bitácoras registradas	91
Figura 3.37 Dialogo de los detalles de la bitácora.....	92
Figura 3.38 Prototipo once en la ventana de los status de los operadores	93
Figura 3.39 Prototipo doce en la ventana de crear usuario actualizada	95
Figura 3.40 Prototipo doce en la ventana de descansos en la vista de asignar descanso	96

Figura 3.41 Prototipo doce en la ventana de descansos en vista de remover descanso	97
Figura 3.42 Prototipo doce en la ventana para agregar pólizas	98
Figura 3.43 Prototipo doce en la ventana de pólizas	99
Figura 3.44 Dialogo de los detalles de la póliza	100
Figura 3.45 Ventana emergente para asociar tractocamiones a póliza	100

Anexo A

Figura A.1 Inicio de sesión.....	120
Figura A.2 Pantalla para crear un nuevo usuario.....	121
Figura A.3 Pantalla principal de la aplicación Freightem.....	122
Figura A.4 Pantalla de elementos asociados al tractocamión	123
Figura A.5 Pantalla de las características específicas del operador	124
Figura A.6 Pantalla de edición de características específicas del operador	125
Figura A.7 Pantalla de cambio de aceite	126
Figura A.8 Pantalla para solicitar refacciones	128
Figura A.9 Pantalla para agregar operador	129
Figura A.10 Pantalla para eliminar operador.....	130
Figura A.11 Ventana emergente para modificar el operador asignado.....	131
Figura A.12 Pantalla para reasignar operador	132
Figura A.13 Pantalla para asignar planas	133
Figura A.14 Ventana emergente para modificar las planas asignadas.....	133
Figura A.15 Pantalla de pólizas registradas.....	134
Figura A.16 Pantalla para registrar póliza	135
Figura A.17 Pantalla para asignar descanso a los tractocamiones	136
Figura A.18 Pantalla de tractocamiones en descanso	137
Figura A.19 Pantalla para registrar bitácoras de mantenimiento	139
Figura A.20 Pantalla de los mantenimientos registrados.....	140
Figura A.21 Pantalla de las solicitudes de refacciones	141
Figura A.22 Pantalla del estado de disponibilidad de las unidades	142
Figura A.23 Pantalla de viajes pendientes	143
Figura A.24 Ventana emergente para asignar tractocamión al viaje	144
Figura A.25 Pantalla de viajes pendientes.....	145
Figura A.26 Pantalla del estado de la licencia de los operadores	146
Figura A.27 Pantalla de viajes terminados	147
Figura A.28 Pantalla para agendar viajes	148

Índice de gráficos

Capítulo 4

Gráfica 4.1 Respuestas de la pregunta uno	105
Gráfica 4.2 Respuestas de la pregunta dos	106
Gráfica 4.3 Respuestas de la pregunta tres	107
Gráfica 4.4 Respuestas de la pregunta cuatro	108
Gráfica 4.5 Respuestas de la pregunta cinco	109
Gráfica 4.6 Respuestas de la pregunta seis	111

Índice de tablas

Tabla 1 Ventajas y desventajas de las metodologías ágiles	11
---	----

Capítulo 2

Tabla 2.1 Comparación de aplicaciones respecto a la aplicación de la tesis	36
--	----

Introducción

Las empresas de transporte desde sus inicios han hallado una forma práctica de gestionar su información que solía ser funcional en el pasado, cuando aún la tecnología no era accesible a toda la población, pero con el avance de la tecnología en las últimas décadas esta ha revolucionado por completo múltiples sectores de la sociedad, debido a esto, se han logrado cumplir funciones que superan a la capacidad humana (Mejía García & Acosta Pérez, 2019). Dentro de la logística de transporte se aborda la forma de gestionar la información e incrementar la productividad y eficiencia de los trabajadores del transporte, este sector transporta alrededor del 83.9% de carga a nivel nacional (Jiménez Sánchez & Jiménez Castillo, 2016).

En las últimas décadas, las aplicaciones móviles han sido diseñadas por desarrolladores para sistemas operativos específicos, como Android e iOS, cada uno con sus respectivas tiendas digitales. En años recientes, se ha presenciado un desarrollo constante en el uso de tecnologías de la información y comunicación, particularmente a través de dispositivos móviles, lo que ha impulsado el acceso a aplicaciones web y móviles, abarcando ámbitos como el comercio electrónico y la productividad (Acosta Espinoza, Lenin León Yacelga, & Sanafria Michilena, 2022).

Al mismo tiempo, las aplicaciones híbridas se alzaron como puentes entre dos mundos, fusionando la magia de las aplicaciones con la esencia nativa y la capacidad de profundizar en lenguajes específicos. En este contexto, las aplicaciones móviles han evolucionado hasta convertirse en el principal canal de comunicación de la sociedad, destacándose por su versatilidad y eficiencia. Estas herramientas permiten acceder a información importante y confiable en cualquier momento del día o de la noche, manteniendo a los usuarios actualizados sobre cualquier tema de su interés (Martínez, Moreno, & Hernández, 2020).

En empresas medianas que poseen un potencial de crecimiento significativo, continúan operando de forma arcaica, utilizando métodos rudimentarios para gestionar su información, lo que tiende a ser una forma obsoleta de gestión. Como menciona Cardona (2024), a pesar de ser uno de los sectores económicos que más aportan a la logística siguen presentando resistencia ante las nuevas tecnologías. Esta práctica dificulta el acceso remoto a la

información y afecta significativamente la eficiencia operativa de las empresas en sus actividades diarias, como rectificó Cardona (2024), las empresas de transporte poseen dificultades en cuanto a la comunicación para las actividades que demanda el transporte.

La creciente demanda de servicios de transporte intensifica el impacto negativo de una gestión lenta, provocando la pérdida de oportunidades comerciales y, en última instancia, pérdidas económicas para las empresas. En este contexto, la versatilidad logística de algunas empresas se destaca como una ventaja crucial, su capacidad para adaptarse a cualquier tipo de carga, velocidad de respuesta, y su accesibilidad permiten ofrecer servicios “puerta a puerta” con eficiencia. Esta flexibilidad no solo optimiza la gestión del transporte, sino que también mitiga las pérdidas económicas y fortalece la posición en el mercado (Jiménez Sánchez & Jiménez Castillo, 2016).

En la presente tesis se plantea el desarrollo de una aplicación móvil que permitirá a los usuarios acceder a la información en el momento y lugar que sea solicitada, lo cual mejorará el desempeño laboral. Además de diversas funciones que aumentarán la productividad de la empresa en sus diferentes departamentos como el de administración, mantenimiento, tráfico y ventas. La aplicación se desarrollará mediante pequeñas tareas consecutivas, con base en actividades continuas para cubrir los máximos campos de mejora así como estándares de calidad.

Antecedentes

Las empresas dedicadas al sector de la logística, en especial al transporte de carga han evolucionado a la par de la tecnología, siendo esta última la que ha tenido el crecimiento más notorio, evolucionando a un ritmo frenético, de esa forma, brindando más herramientas con fines de mejorar la forma de trabajo de diversos sectores y gestionar una mayor cantidad de información (Roig & Castillo, 2022).

Algunas empresas medianas que pueden tener alrededor de 100 unidades propias han recurrido constantemente al uso de métodos rudimentarios pero efectivos de cierto modo para gestionar las empresas, de modo que la información se tiene en registros en forma de papel y en algunos casos en hojas de cálculo e incluso con la información centralizada en un único equipo de cómputo, lo cual no es seguro ni eficiente cuando se busca un crecimiento de la

empresa, además, se genera una dependencia de dicho equipo de cómputo causando vulnerabilidades (López Tulcanaza, 2010).

La información que suele ser importante para tener acceso de forma remota puede llegar a ser un extenso número de datos como los números de los tractocamiones, también conocidos en el sector como número económico, nombre de los operadores, línea o empresa para la que trabajan, tipo de tractocamión, configuración vehicular del tractocamión, tipo de remolques, placas del tractocamión y remolque, nombre de aseguradora, número de póliza, entre muchos otros datos que son relevantes para diversas áreas y situaciones.

Como menciona Pazmiño, Poma, Cárdenas, & Ordoñez (2020), en donde al ser empresas en crecimiento se demanda de la integración de diferentes departamentos para optimizar la carga de trabajo y controlar el proceso de un área en específico, los departamentos básicos de trabajo en una línea de transportes es el departamento de ventas, el cual es el encargado de negociar viajes con los clientes, el departamento de tráfico, es el encargado de asignar el tractocamión ideal para cada viaje, así mismo, dar seguimiento a cada uno de los viajes realizados en tiempo real, el departamento de administración que se encarga de mantener un orden en cuanto a las unidades con sus elementos relacionados, ya sea su remolque, su dolly y operador, el departamento de recursos humanos que es el encargada de los temas en general de los empleados, el departamento de mantenimiento que es la encargada de procurar mantener las unidades en su máxima operatividad posible y también los rescates en carretera a las unidades averiadas.

Todos los departamentos mencionados comparten información y en casos especiales dependen de la información de otro departamento que no se encuentran en un mismo lugar de trabajo por el entorno en que se desarrollan las actividades de cada departamento, lo que conlleva a demandar un servicio externo para manejar esta información que se necesita en tiempo real.

Planteamiento del problema

Las empresas de transporte que gestionan su información de manera centralizada en soportes físicos, como hojas de papel, o en menor medida, en hojas de cálculo de Excel, enfrentan serias limitaciones operativas. Al depender de una persona para el acceso a esta información,

se restringe la posibilidad de consulta remota, lo cual ralentiza la toma de decisiones estratégicas y afecta la agilidad de respuesta ante situaciones imprevistas. Esta forma de gestión ineficiente complica las operaciones diarias, reduce la flexibilidad y genera dependencias innecesarias.

Además, la comunicación y transmisión de información entre los diferentes departamentos se vuelve complicada, ya que no existe un sistema digital integrado que permita a los departamentos de administración, tráfico, mantenimiento y ventas acceder a la información de manera simultánea y eficiente. Esto da lugar a retrasos, duplicación de esfuerzos, errores humanos y una gestión rudimentaria de los procesos clave de la empresa, como el seguimiento del estado de las unidades, la programación de viajes, el mantenimiento preventivo y el control de pólizas.

En consecuencia, las empresas de transporte que no han adoptado herramientas tecnológicas modernas enfrentan mayores costos operativos, disminución de la eficiencia y una capacidad limitada para reaccionar ante las necesidades del mercado, lo que pone en riesgo su competitividad en el sector.

Solución propuesta

La presente tesis tiene como objetivo desarrollar una aplicación móvil que permita el acceso remoto a la información de los tractocamiones y operadores, así como diversas funciones que aportan una mejor gestión de las unidades, con el propósito de facilitar y mejorar las estrategias de trabajo diarias. Este desarrollo proporcionará beneficios directos a la empresa en su conjunto, así como un impacto positivo en los usuarios designados y al entorno general de la empresa.

En la base de datos se incluirán registros que describan los objetos de trabajo, como los tractocamiones. Entre los atributos descriptores se encuentran el número de tracto, marca, modelo, número de serie, motor, año del tractocamión, placas y la línea a la que pertenecen. Asimismo, se registrarán los atributos relacionados con los dolly y los remolques, estos últimos siendo plataformas estándar. Es especialmente relevante incluir la línea a la que pertenecen, en caso de que se trabaje en colaboración con otras líneas que es bastante común en el sector. Además de funciones que resulten útiles para diferentes departamentos de las

empresas, tal como el departamento de administración, el departamento de tráfico, el departamento de mantenimiento y el departamento de ventas.

Incluyendo funciones que mantengan en tiempo real los estatus de las unidades en cuanto a disponibilidad lo que significa que las unidades son aptas para operar sin necesidad de estar haciendo uso de aplicaciones externas para conocer dicho estatus.

Justificación

El sector de autotransporte enfrenta grandes desafíos relacionados con la gestión eficiente de sus flotas, especialmente en empresas medianas que no cuentan con herramientas tecnológicas avanzadas. La falta de acceso rápido a la información actualizada sobre unidades, operadores, y viajes afecta negativamente la toma de decisiones y genera ineficiencias en los procesos de operación, mantenimiento y administración.

El propósito de este proyecto es ofrecer una solución tecnológica que resuelva estos problemas, mejorando la accesibilidad y control de la información clave. Con la implementación de la aplicación móvil, las empresas podrán optimizar la gestión de sus unidades de transporte mediante:

Acceso remoto y centralización de la información: Muchas empresas de autotransporte operan sin un sistema centralizado que les permita acceder de manera rápida y remota a la información relevante de sus unidades. Esto genera demoras en la toma de decisiones y dificulta la supervisión del estado de las flotas. La aplicación ofrecerá un acceso remoto en tiempo real, permitiendo a los usuarios realizar consultas y tomar decisiones desde cualquier dispositivo móvil.

Mejora en la gestión del mantenimiento: La falta de un sistema adecuado de seguimiento del mantenimiento preventivo y correctivo puede causar fallas inesperadas y generar mayores costos. Con la implementación de un módulo de bitácoras de mantenimiento, se permitirá un mejor seguimiento y planificación de los servicios, evitando paros innecesarios y prolongando la vida útil de las unidades.

Optimización de la operación y planificación de viajes: Actualmente, la gestión de viajes y asignación de recursos (tractocamiones, planas, dollys y operadores) suele hacerse de manera

manual o desorganizada, lo que genera pérdidas de tiempo y recursos. La aplicación agilizará este proceso mediante un sistema que permitirá agendar viajes, asignar unidades y dar seguimiento detallado en tiempo real, mejorando así la eficiencia operativa.

Gestión de pólizas y cumplimiento legal: La gestión de pólizas es otro aspecto crítico, ya que es necesario que todas las unidades cuenten con seguros vigentes para operar legalmente. La aplicación permitirá un control eficiente de las pólizas, evitando sanciones y asegurando que todas las unidades estén cubiertas por seguros en regla.

Segmentación por departamentos y mejora en la seguridad: Al segmentar las funciones de la aplicación según algunos departamentos de las empresas de autotransporte, se garantiza que cada departamento (administración, tráfico, mantenimiento, ventas) tenga acceso solo a la información relevante para su operación. Esto mejora la seguridad de los datos y evita accesos no autorizados.

Finalmente, el desarrollo multiplataforma en Flutter garantiza que la solución sea accesible desde dispositivos Android e iOS, ampliando el alcance de su implementación. La aplicación está diseñada específicamente para empresas medianas, que normalmente no tienen acceso a herramientas de gestión de flotas, y se espera que tenga un impacto positivo al optimizar tiempos, reducir costos y mejorar el control operativo.

Con esta herramienta, se espera un impacto positivo en la operación de las empresas de autotransporte, reduciendo costos, tiempos muertos y mejorando la toma de decisiones en tiempo real, lo que fortalecerá su competitividad en el sector logístico.

Objetivo general

Desarrollar una aplicación móvil para aumentar la eficiencia operativa de las empresas medianas dedicadas al transporte de carga, mediante el uso del lenguaje de programación Dart, con una base de datos SQLite integrada.

Objetivos específicos

- Analizar las necesidades específicas de una línea de transporte para determinar los requisitos funcionales y no funcionales de la aplicación, realizando un estudio de

campo en conjunto con los encargados de los departamentos a los que va dirigida principalmente la aplicación.

- Diseñar el modelo relacional para la creación de una base de datos relacional mediante el uso del lenguaje SQLite, que contendrá la información previamente procesada para almacenar la información de la aplicación.
- Crear una interfaz de usuario intuitiva y fácil de usar que permita a los usuarios visualizar la información general de primera instancia, así como la información específica de cada unidad y una navegación óptima entre sus funciones mediante el lenguaje de programación Dart.
- Implementar la aplicación móvil dentro de una empresa de transporte de carga, con el fin de verificar su desempeño deseado mediante una prueba de uso.
- Aplicar una encuesta de usabilidad entre los usuarios finales para identificar mejoras en la aplicación móvil.

Metodologías

En esta sección se presentan diversas metodologías para el desarrollo de software las cuales han sido opciones a aplicar para el desarrollo de la aplicación móvil.

La metodología en cascada, también conocida como *waterfall*, es una de las metodologías tradicionales y por ende, una de las más usadas debido a su fácil implementación y de su secuencia lógica. La metodología cascada está orientada a la planeación, iniciando el proyecto con la definición de requerimientos con la idea de lograr una alta calidad en los resultados, el proyecto se estructura de forma que sigue una secuencia fija sin margen de retroceso. Dicha metodología consta de cinco etapas: Requerimientos, Diseño, Implementación, Pruebas y Mantenimiento (Patino, 2020).

En la Figura 1 se observa el ciclo de vida de la metodología de cascada, también conocido como el ciclo de vida clásico.

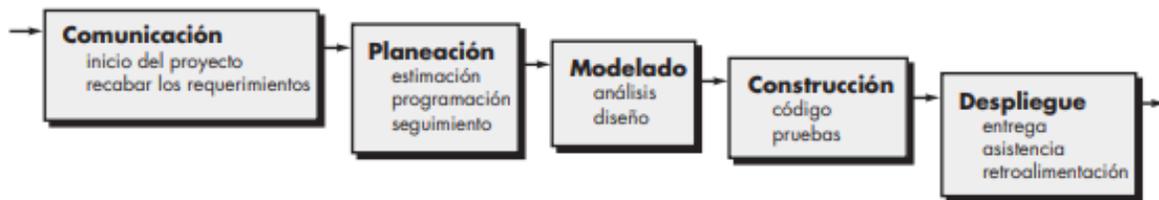
La metodología Scrum está basada en el control de proceso, esta permite abordar proyectos un tanto complejos y adaptarlos para que las entregas de productos tengan el máximo valor posible, a diferencia de la metodología en cascada, Scrum tiene un enfoque iterativo, es decir, permite retroceder de la fase de desarrollo con el fin de hacer cambios imprevistos, además,

demanda un equipo de trabajo a los que se les asigna su respectivo rol. La implementación de Scrum se basa en algunos aspectos tales como: Transparencia, Inspección y Adaptación (Barrera, 2020).

En la Figura 2 se observa el flujo de trabajo de la metodología Scrum.

Figura 1

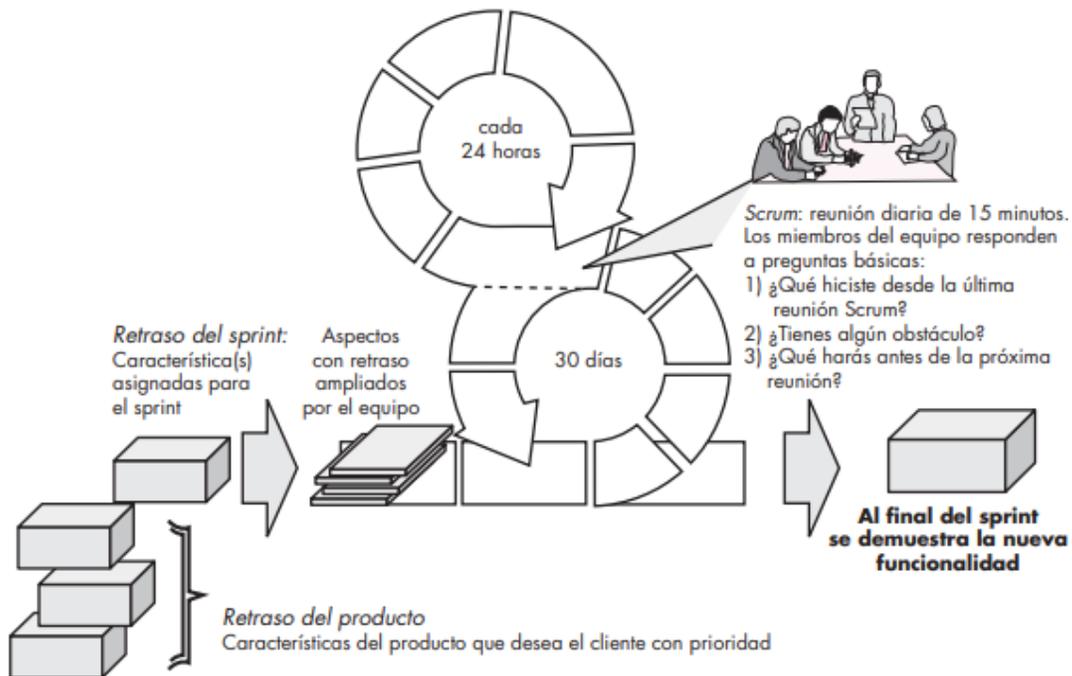
Ciclo de vida de la metodología de cascada



Fuente: (Pressman, 2010)

Figura 2

Flujo general del proceso Scrum



Fuente: (Pressman, 2010)

La metodología de prototipos evolutivos se basa en la creación e implementación de versiones parciales del sistema final, las cuales se desarrollan rápidamente para facilitar pruebas tempranas y obtener retroalimentación continua. Los prototipos se construyen en el menor tiempo posible, lo cual está influenciado por factores como la complejidad del proyecto y el conocimiento de las herramientas utilizadas.

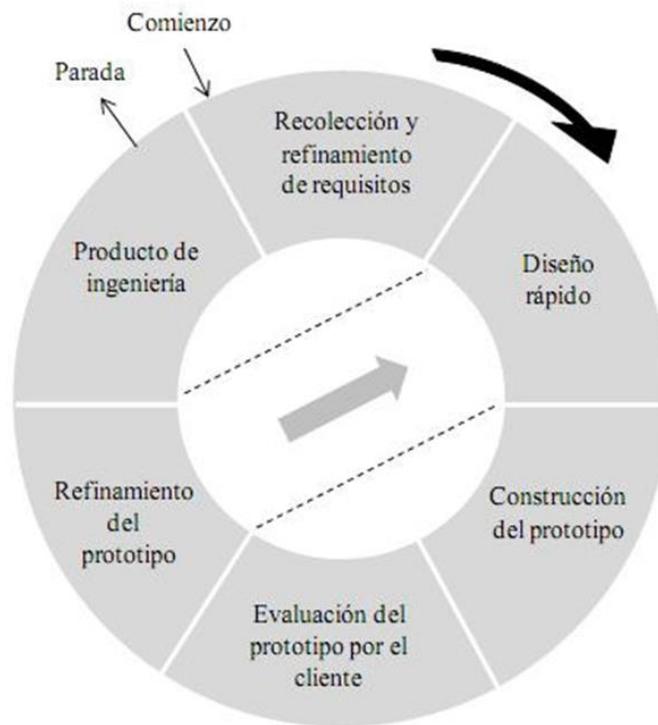
En este contexto, dichos prototipos se presentan a los usuarios para su evaluación, y después de un período de prueba, se recopila retroalimentación para realizar los ajustes necesarios. Esta metodología se articula a través de varias fases: Investigación preliminar, definición de requerimientos del sistema, diseño técnico, desarrollo de pruebas y por último, la operación y mantenimiento.

La metodología de prototipos evolutivos es una parte integral del enfoque RAD (Desarrollo Rápido de Aplicaciones), que busca acelerar el ciclo de vida del desarrollo de software al utilizar técnicas como la iteración rápida y la participación continua del usuario. En RAD, los prototipos evolutivos permiten que el software se desarrolle en ciclos cortos y repetidos, facilitando ajustes y mejoras basadas en la retroalimentación del usuario. Este enfoque no solo optimiza el tiempo de desarrollo, sino que también asegura que el producto final esté alineado con las necesidades y expectativas del usuario. Al integrar prototipos evolutivos dentro de RAD, se potencia la flexibilidad y la capacidad de respuesta del proceso de desarrollo, ofreciendo una solución ágil y adaptativa a los requisitos cambiantes (Maida & Pacienza, 2015).

Cabe mencionar que la metodología de prototipos evolutivos tiene como ventajas que es un diseño bastante flexible y adaptable a las necesidades que surjan por parte de los clientes, tiene mucho margen de mejora entre las entregas de los prototipos y no demanda la necesidad de brindar capacitaciones a los usuarios finales dado que ya han utilizado el sistema en una fase de prototipo. En la Figura 3 se observa el ciclo de vida de un prototipo, en donde se muestra que es un proceso iterativo en las fases intermedias del ciclo hasta que el prototipo sea capaz de continuar con el ciclo de vida.

Figura 3

Ciclo de vida de un prototipo



Fuente: (Maida & Pacienza, 2015)

En la Tabla 1 se mencionan las ventajas y desventajas del uso de estas metodologías ágiles.

Una vez presentadas una serie de metodologías se determina que para este caso, se va a desarrollar bajo los fundamentos de la metodología de RAD con el enfoque de prototipos evolutivos, debido a que brinda la posibilidad de tener diseños funcionales del sistema sin necesidad de que sea el final, también porque evita una capacitación futura del uso del sistema y, la última razón, es que se adapta a nuevos requerimientos que surjan durante el desarrollo, lo cual es un posible escenario durante el desarrollo de la aplicación, dado que a las empresas que va dirigida la aplicación no dimensionan de forma inicial la capacidad que tiene el desarrollo de una herramienta de esta índole.

Tabla 1

Ventajas y desventajas de las metodologías ágiles

Ventajas	Desventajas
<ul style="list-style-type: none">• El cliente (persona para quien se desarrolla el proyecto) es considerado como parte del equipo, por ende, se le debe de informar de los avances de forma continua.• Permite obtener comentarios y sugerencias de los usuarios desde las etapas iniciales del desarrollo, lo que ayuda a identificar y corregir posibles problemas antes de que el producto final esté completamente desarrollado.	<ul style="list-style-type: none">• La implicación continua del cliente llega a ser complicada en algún punto del proyecto.• El proceso de desarrollo iterativo y la evolución constante de los prototipos pueden dificultar la gestión del proyecto, especialmente en términos de planificación de plazos y asignación de recursos.

Herramientas de desarrollo

A continuación, se presentan las herramientas que se han considerado necesarias para lograr los objetivos desde un entorno de desarrollo como Visual Studio, Adobe Illustrator para los recursos gráficos, *SQLite* como lenguaje de base de datos, *DB Browser for SQLite* como sistema gestor de base de datos, y *Flutter* como *framework* de desarrollo.

Visual Studio Code es un editor de código altamente personalizable y ligero desarrollado por Microsoft, el cual, facilita la escritura de código mediante una interfaz intuitiva y personalizable, además de integrar herramientas como control de versiones para una experiencia de desarrollo fluida (Microsoft, 2024). Como herramienta de diseño de imágenes se optó por Illustrator, dado que permite a los diseñadores crear gráficos vectoriales y manipular texto con precisión para producir ilustraciones y diseños gráficos profesionales (Adobe, 2024).

Dentro del desarrollo de programación se usará *Flutter*, este framework brinda un *kit* de desarrollo de interfaz de usuario de código abierto desarrollado por *Google*. Facilita la creación de interfaces de usuario interactivas y atractivas para aplicaciones móviles y web mediante la composición de *widgets*, y agiliza el proceso de desarrollo con su función de *Hot Reload* (Flutter, 2024). Como SGBD se designó a *DB Browser for SQLite*, dado que es una base de datos ligera y sin servidor que se integra directamente en las aplicaciones móviles, programado mediante el lenguaje *SQLite*. Además, proporciona almacenamiento de datos local de manera eficiente y segura, adecuado para aplicaciones móviles y de escritorio, sin necesidad de configuraciones complejas de servidor (SQLite, 2024).

Alcances

El presente proyecto tiene como alcance el desarrollo de una aplicación móvil diseñada específicamente para empresas de autotransporte, enfocándose en la gestión integral de sus unidades. A continuación, se describen los principales alcances:

1. Gestión de la información de las unidades: La aplicación permitirá administrar de manera eficiente la información detallada de cada unidad, incluyendo datos sobre tractocamiones, planas y dollys, asegurando que toda la información esté centralizada y accesible desde una sola plataforma.
2. Acceso remoto a la información: Los usuarios podrán consultar de forma remota, en tiempo real, el estado y detalles de cada unidad desde cualquier dispositivo móvil, brindando mayor flexibilidad y mejor toma de decisiones.
3. Funciones para diferentes departamentos: La aplicación ofrecerá funcionalidades específicas para distintos departamentos, como administración, tráfico, mantenimiento y ventas. Cada departamento tendrá acceso a las herramientas necesarias para cumplir con sus responsabilidades de manera eficiente.
4. Bitácoras de mantenimiento: Se implementará un sistema de bitácoras de mantenimiento donde se registrará el historial de servicios y reparaciones de cada unidad. Esto facilitará el seguimiento de los mantenimientos preventivos y correctivos, asegurando la disponibilidad óptima de las unidades.

5. Agendar y dar seguimiento a viajes: La aplicación contará con funcionalidades para agendar viajes, permitiendo que el usuario asigne unidades y operadores. Además, se podrá monitorear el estado de cada viaje, desde su programación hasta su finalización.
6. Estado de disponibilidad de las unidades: La herramienta proporcionará información actualizada sobre la disponibilidad de las unidades, permitiendo visualizar si una unidad se encuentra en tránsito, vacía, en descanso o en mantenimiento.
7. Gestión de pólizas: Se incluirá un módulo para administrar las pólizas de seguro de cada unidad, facilitando el seguimiento de vigencias y renovaciones, y asegurando que todas las unidades cumplan con los requisitos legales.
8. Público objetivo: Estas empresas podrán mejorar significativamente la operación y administración de sus unidades mediante la implementación de esta solución.
9. Desarrollo multiplataforma: Al estar desarrollada en Flutter, la aplicación será multiplataforma, es decir, funcionará tanto en dispositivos Android como iOS, asegurando mayor accesibilidad y cobertura.
10. Implementación piloto: Inicialmente, la aplicación será implementada en una empresa que cumpla con las características descritas, con el fin de validar su correcto funcionamiento y ajustarla a las necesidades reales del sector.

Este proyecto busca dotar a las empresas de autotransporte de una herramienta integral que les permita mejorar su operación y gestión, optimizando tiempos, costos y recursos.

Limitaciones

Una de las principales situaciones que la aplicación móvil no abordará es el seguimiento de las rutas en tiempo real, debido a que eso demanda el uso externo de dispositivos de rastreo que no han sido considerados para este desarrollo.

Estructura del documento

El presente documento consta de cinco capítulos más el protocolo, a continuación se menciona brevemente en que consiste cada uno de estos.

En el protocolo se introduce al lector sobre lo que se desarrollará el proyecto, contextualizando el desarrollo del mismo.

En el Capítulo 1, se presenta todo el conocimiento teórico con el que se fundamenta el desarrollo del proyecto, además de presentar fundamentos necesarios para comprender los aspectos a abordar.

En el Capítulo 2, se aplica por completo la metodología de desarrollo de software elegida en donde se aclara cómo fue realizado el desarrollo de la aplicación móvil.

En el Capítulo 3, se demuestra el desarrollo de cómo fue realizada la aplicación.

En el Capítulo 4, se muestra la validación de la aplicación, en donde se muestra una encuesta de usabilidad a los usuarios que usaron la aplicación Freightem.

Continuando con las Conclusiones a las que se llegaron una vez finalizada la presente tesis.

En seguida, los Trabajos futuros, en donde se hace mención de alternativas que pueden surgir como cambios a la aplicación móvil posterior al desarrollo de la presente tesis.

Por último, en el Anexo A, se muestra al lector la forma de cómo usar la aplicación desarrollada, abarcando todas las funciones posibles por la aplicación móvil.

Capítulo 1. Marco teórico

En esta sección se presenta un conjunto estructurado de teorías, conceptos, modelos y antecedentes relevantes que proporcionan el fundamento intelectual y contextual necesario para comprender, analizar y abordar el problema y área en la que se desarrolla esta tesis.

Todos los temas definidos en el Capítulo 1 se enmarcan en la tesis ya que cada uno aborda aspectos cruciales para el desarrollo, implementación y operabilidad de la aplicación móvil para la gestión de unidades de transporte de carga.

1.1 Términos técnicos

En esta sección se abordan diversos conceptos que son necesarios conocer para entender el aspecto de desarrollo por la parte técnica.

1.1.1 Bases de datos

Una base de datos es un conjunto de datos almacenados en tablas y registros. Toda base de datos es diseñada para cumplir las necesidades de una empresa u organización específica. Las bases de datos no solo contienen la información que sea ingresada, sino también una descripción de dichos datos, a dicha descripción se le conoce como los metadatos los cuales se almacenan en un diccionario de datos permitiendo así una independencia de datos lógica-física (Menéndez, 2022).

1.1.2 Sistemas gestores de bases de datos

Los Sistemas Gestores de Bases de Datos (en adelante SGBD) son aplicaciones que permiten definir, crear y administrar una base de datos. Los SGBD siguen un modelo muy similar al que se utiliza en los lenguajes de programación orientados a objetos, donde existe una implementación interna de un objeto y una especificación externa separada. Los usuarios de un objeto sólo ven la especificación externa y no tienen que preocuparse por cómo se implementa el objeto internamente. La ventaja de este modelo (llamado abstracción de datos) es que la implementación interna de un objeto se puede cambiar sin afectar a sus usuarios, ya que la especificación externa permanece igual. De manera similar, un sistema de base de datos separa la definición de la estructura física de los datos de su estructura lógica y

almacena esta definición en la base de datos. El SGBD actúa como intermediario entre la base de datos y la aplicación. Un SGBD proporciona los siguientes servicios:

- Permite definir la base de datos mediante un lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
- Permite las acciones de insertar, actualizar, eliminar y consulta de los datos mediante un lenguaje de manejo de datos. Existen dos tipos de lenguajes de manejo de datos, los procedurales y no procedurales, la manera principal en que se distingue uno de otro es el modo en que acceden a los datos. Los lenguajes procedurales manipulan la base de datos registro por registro, mientras que los lenguajes no procedurales procesan un conjunto de registros. Además, los lenguajes procedurales deben especificar qué operaciones realizar para obtener un resultado, mientras que los no procedurales se especifica que dato se obtiene sin especificar cómo hacerlo.
- Proporciona acceso controlado a la base de datos mediante sistemas de seguridad, sistemas de integridad que mantienen la integridad y consistencia de los datos, sistema de control de concurrencia que permite el acceso compartido a la base de datos, un sistema de control de recuperación que restablece la base de datos después de algún fallo y un diccionario de datos que contiene la descripción de la base de datos.

Una herramienta de los SGBD son las vistas, que permiten que cada usuario tenga una visualización especial de la base de datos. El lenguaje de definición de datos permite definir vistas para una manejabilidad más fácil a los usuarios (Abelló, Rodríguez, & Rollón, 2021).

Como menciona Abelló et al. (2021), existen ventajas del uso de estos SGBD:

1.1.2.1 Ventajas de los SGBD

- Mejora la disponibilidad de los datos para los usuarios. Aunque en una empresa el propietario de los datos es como tal la empresa u organización, los empleados deben tener acceso a la base de datos que se comparte entre las diferentes áreas, existiendo una mayor disponibilidad y transparencia,

- Mejora y aumenta la normalización de la documentación. Los SGBD ofrecen herramientas para reflejar el contenido semántico de los datos, bien conocidos como la metadata.
- Mejora la eficiencia en la captura de los registros. Al ser un sistema centralizado de captura solo se hace una vez, lo que reduce la redundancia de datos.
- Reducción de almacenamiento. Al estar la información albergada en un mismo equipo evita tener en cada equipo toda la información que muchas veces puede ser inútil en ciertas áreas de trabajo.

De igual forma, Abelló et al. (2021) mencionan que también existen desventajas del uso de los SGBD:

1.1.2.2 Desventajas de los SGBD

- Dependiendo de los alcances puede demandar equipos potentes. Algunos SGBD requieren mejores equipos para su funcionamiento lo que demanda
- Personal especializado. Es imprescindible algún conocedor del sistema, conocido como Administrador de la Base de Datos (DBA), diseñadores y programadores.
- En un principio la implementación puede ser tardada, dependiendo del tipo de información que maneja la organización.
- La rentabilidad de una base de datos suele reflejarse a mediano y largo plazo.

Una vez presentadas las ventajas y desventajas de la implementación de un sistema de bases de datos junto con el SGBD se llega a la conclusión que es necesario dicha implementación, principalmente por un tema de modernización y pensamiento a futuro de la empresa.

1.1.3 Modelo de datos

Los modelos de datos es la forma de ofrecer una abstracción necesaria de los datos a través de la jerarquía de niveles. Los modelos de datos son un conjunto de conceptos que en unión sirven para describir una base de datos, es decir, su estructura, los datos, relaciones entre los datos y restricciones de ellos. Estos modelos tienen una serie de operaciones básicas para realizar consultas y actualizaciones sobre los datos. Los modelos de datos se pueden clasificar de acuerdo a conceptos que ofrece para describir una estructura de una base de datos, por medio de una jerarquía de niveles. Los modelos de alto nivel, también conocidos como

modelos conceptuales, disponen de conceptos bastante parecidos a cómo los usuarios perciben los datos. Por otro lado, los modelos de bajo nivel, también conocidos como modelos físicos, proporcionan detalles de cómo se almacenan los datos. Los modelos conceptuales utilizan eso mismo, conceptos, tales como entidades, atributos y relaciones (Menéndez, 2022).

Una entidad representa un objeto del mundo real, un atributo representa alguna propiedad descriptora y una relación describe una interacción entre dos o más entidades. Cada SGBD soporta un modelo lógico siendo el más común el relacional, red y jerárquico. Para esta tesis, se utilizará el modelo relacional.

1.1.4 Modelo relacional

Tal como señala Menéndez (2022) es un modelo de datos en el que se basan la mayoría de SGBD en la actualidad. Inicialmente se presenta una estructura de datos relacional, seguido de ciertas reglas de integridad que deben cumplirse en las bases de datos. El modelo relacional se basa en el concepto matemático de la relación, que en este contexto se refiere a la relación entre tablas compuestas por columnas y filas. En un SGBD el usuario solo necesita entender la base de datos como un conjunto de tablas, un atributo es el nombre de una columna de una relación, en este modelo las relaciones se utilizan para guardar información sobre los registros de una base de datos. Se puede entender de una forma bidimensional en la que las filas corresponden a registros individuales (también conocidos como tuplas) y las columnas a atributos de esos mismos registros, donde particularmente los atributos pueden aparecer en la relación de cualquier orden.

De acuerdo con Marqués (2011), un dominio es el conjunto de valores permitidos de cada uno de los atributos, la cual es una característica muy importante en este modelo. Cada uno de los atributos tienen su dominio, principalmente delimitado por el tipo de datos con el que haya sido definido, también por la cantidad de caracteres. Una tupla es una fila de una relación, o bien cada registro de la tabla. Los elementos de una relación son las tuplas y filas de la tabla.

El grado de una relación es la cantidad de atributos que contiene, una vez definida la relación es inusual que cambie su grado. La cardinalidad de una relación es el número de tuplas que contiene, dado que en las relaciones se van insertando y borrando tuplas con frecuencia.

Las relaciones o bien tablas tienen sus características propias, entre las que destacan las siguientes:

- Cada relación tiene un nombre que no se puede repetir en toda la base de datos.
- No hay atributos que se llamen igual.
- El orden de los atributos no importa.
- Cada tupla es distinta a las demás, no hay tuplas duplicadas.
- El orden de las tuplas no importa.

Existen dos tipos de relaciones, las relaciones base y las vistas. Las relaciones base son las relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada. Se dice que las relaciones base son relaciones autónomas.

Las vistas, también conocidas como relaciones virtuales, son relaciones con nombre y derivadas, es decir, que se obtienen a partir de otras relaciones. Las vistas no poseen datos almacenados propios; los datos que muestran corresponden a la base de datos en la relación base.

Claves: Dado que en una relación no existen tuplas repetidas, estas se distinguen unas de otras, es decir, se identifican con un identificador único. Se le conoce como super clave o clave primaria a aquel atributo que identifica específicamente a cada tupla de una relación. Por otro lado, existen también las claves foráneas, la cual es una columna o un conjunto de columnas de una relación que hace referencia a la clave primaria de otra relación. Para las claves foráneas también existen reglas que se deben cumplir, y una de ellas es fundamental: la regla de integridad referencial (Menéndez, 2022).

1.1.4.1 Esquema de una base de datos relacional

Tiene la función de plasmar de una forma abstracta la base de datos, dándole nombre a las relaciones y los atributos que constituyen dichas relaciones, los dominios que tienen los atributos, claves primarias y claves foráneas.

En el esquema los nombres de las relaciones o tablas van seguidos de los atributos que contiene cada relación marcados por un paréntesis, las claves primarias vienen subrayadas en cada relación, por otro lado, las tablas que se relacionan entre sí se debe agregar la clave primaria en una de las tablas de la relación según la cardinalidad (Menéndez, 2022).

1.1.5 Reglas de integridad

Una vez definida la estructura del modelo relacional, se deben implementar reglas de integridad de los datos que se almacenarán en la base de datos. Una vez definido el dominio de los atributos se impone automáticamente una restricción sobre los valores permitidos en cada atributo, a esto se le conoce como restricciones de dominio. Además de esta, existen otras dos reglas de integridad que son relevantes y las deben de cumplir todas las bases de datos relacionales, la regla de integridad y la regla de integridad referencial. Junto con este tipo de restricciones vienen los valores nulos, cuando en una tupla existen algún atributo desconocido, es decir, que no ha sido introducido, se dice que es nulo. Un valor nulo no representa un valor de cero, en caso de ser un atributo de tipo cadena no representa una cadena vacía, lo que significa es ausencia de información. Como los nulos no son un valor como tal, deben de tratarse de una manera especial, lo que ocasiona problemas en el uso de la base de datos.

Volviendo a la mencionada regla de integridad de entidades, esta se aplica a las claves primarias de las relaciones en donde ninguno de los atributos que componen clave primaria puede ser nulo. Este atributo es utilizado para identificar de manera única a las tuplas, en caso de que esta regla no se cumpla se da por hecho que no es necesario identificar cada tupla como única, lo cual no cumple uno de los propósitos de las bases de datos. Esta regla solo se aplica a las relaciones base y a sus claves primarias, no a las claves alternativas.

1.1.5.1 Regla de la integridad referencial

La segunda regla de integridad se aplica a las claves foráneas, si en una relación existe alguna clave foránea sus valores deben de coincidir con los valores de la relación a la que hace referencia, en el caso de que no se requiera deben de ser nulos, siempre y cuando cumpla con las definiciones de la relación. Para mantener la integridad en cuanto a claves foráneas es útil plantear estas preguntas para determinar el comportamiento de dicha clave:

- Regla de los nulos. ¿Tiene sentido que la clave ajena acepte nulos?
- Regla de borrado. ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave foránea?
 - Restringir. NO se permite eliminar la tupla referenciada.
 - Propagar. Se borra la tupla referenciada y se propaga el borrado a las tuplas que las referencian mediante las claves foráneas
 - Anular. Se elimina la tupla referenciada y las tuplas que la referenciaban se cambia a nulo en la clave foránea, en dado caso de que ese atributo acepte valores nulos.
 - Valor por defecto. Se borra la tupla referenciada y las tuplas que la referenciaban ponen en la clave ajena el valor por defecto establecido para la misma, en caso de haberse definido un valor definido.
- Regla de modificación. ¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave foránea?
 - Restringir. NO se permiten modificar el valor de la clave primaria de la tupla referenciada.
 - Propagar. Se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que referencian mediante la clave foránea.
 - Anular. Se modifica la tupla referenciada y las tuplas que la referencian ponen nulo a la clave foránea, en caso de permitir el uso de valores nulos.
 - Valor por defecto. Se modifica la tupla referenciada y las tuplas que la referenciaban ponen en la clave foránea el valor por defecto establecido en la misma.

Estas reglas se deben de definir en todos los atributos que sean definidos como claves foráneas, en caso de que no sean definidas tomaran automáticamente el NO ACTION, en donde simplemente los cambios tendrán efecto en la tabla donde sean modificados. Existe una regla denominada de negocio, esta es dependiendo del funcionamiento de la empresa u organización a la que se le diseñe la base de datos, por decir, en caso de que una empresa no permita más de 10 registros de empleados no dejará ingresar nuevos registros hasta que se

cumpla la regla definida, pero no todos los SGBD permiten este tipo de restricciones (Menéndez, 2022).

1.1.6 Lenguaje SQL

Structured Query Language es un lenguaje estándar que permite manejar los datos de una base de datos relacional. Una mayoría de los SGBD utilizan este lenguaje debido a que permite el acceso a la base de datos, dicho acceso se realiza mediante sentencias, también conocidas como query, existen tres tipos de estas sentencias:

- Sentencias de definición de datos. Son las sentencias que permiten crear, alterar o eliminar las tablas, dicha sentencia también es aplicable a las vistas, índices y triggers.
- Sentencias de manejo de datos. Son las sentencias que permiten insertar, consultar, modificar y borrar datos en las tablas.
- Sentencias de control. Son las sentencias que utilizan los DBA (Database Administrator).

Dentro de las definiciones de las bases de datos es necesario elegir el tipo de dato adecuado para el atributo, depende muchas veces del contexto, es recomendable consultar el manual del SGBD que se esté utilizando para no cometer errores en las definiciones (Menéndez, 2022).

Los tipos de datos que se utilizan en esta tesis son los estándares de SQL:

- VARCHAR. Cadena de caracteres.
- INT Número de n dígitos.

Cabe mencionar que también se usan los NULL que ya se describió no como un valor, si no, como la ausencia de información.

1.1.7 Android

Android es un popular sistema operativo móvil basado en Linux desarrollado por Google y lanzado por primera vez en 2007. Android es un sistema de código abierto muy extendido. Google desarrolla activamente la plataforma Android, pero ofrece parte de ella de forma gratuita para los fabricantes de hardware y proveedores de servicios que quieran utilizar Android en sus dispositivos. Este sistema permite el desarrollo de aplicaciones móviles

utilizando los lenguajes Java y Kotlin. Además, proporciona una variedad de herramientas en el kit de desarrollo de software (SDK) para facilitar el desarrollo, como depuradores, bibliotecas, emuladores y documentación. Asimismo, Android facilita la interacción con el hardware del dispositivo móvil para proporcionar acceso a sus componentes, como la cámara, el teléfono o el control GPS. Gracias a su naturaleza de código abierto y a las herramientas de desarrollo gratuitas disponibles, desarrollar aplicaciones en Android es muy fácil y no requiere ningún hardware especializado (Osuna, 2020).

1.1.8 iOS

iOS es un sistema operativo móvil lanzado oficialmente por Apple el 29 de junio de 2007. Se deriva de macOS y macOS se deriva de Darwin BSD, por lo que es un sistema operativo similar a Unix. iOS tiene cuatro capas de abstracción: la capa central del sistema operativo, la capa de "servicios principales", la capa "medios" y la capa "Cocoa Touch". A diferencia de Android, este sistema operativo sólo funciona en dispositivos desarrollados por la compañía, como el iPhone, iPod, iPad o Apple Watch. Además, el kit de desarrollo del sistema operativo sólo está disponible a través del fabricante del dispositivo, lo que puede aumentar los costos de desarrollo al requerir hardware específico (Osuna, 2020).

1.1.9 IDE

Un entorno de desarrollo integrado (IDE) proporciona un conjunto de herramientas que le ayudan a crear software. La complejidad de los programas desarrollados hoy y los paradigmas de programación dominantes hacen que el uso de estas herramientas sea casi indispensable. Por otro lado, el desarrollo de los dispositivos móviles y el software que contienen ha permitido migrar diversas tareas que habitualmente se realizan en las computadoras a estos dispositivos. Sin embargo, los programas disponibles para programar desde dispositivos móviles no brindan las mismas herramientas que ofrecen los IDE actualmente (Palumbo, 2023).

1.1.10 Framework

Un framework es aquel que proporciona ciertas herramientas para hacer el trabajo más fácil y rápido, es decir, proporciona herramientas muy poderosas que los desarrolladores pueden usar sin perder mucho tiempo construyéndolas. Si se desea utilizar un framework

determinado, es importante saber si es un framework confiable y si tiene buen mantenimiento y actualizaciones periódicas. Un framework de aplicaciones híbridas brinda las herramientas que necesita para desarrollar aplicaciones híbridas de manera rápida y eficiente para que pueda evolucionar la funcionalidad y el estilo visual de las aplicaciones desarrolladas usando los frameworks. También proporciona herramientas para compilar y empaquetar aplicaciones que se pueden utilizar en todos los dispositivos (Forcada, 2020).

1.1.11 Flutter

Flutter es un popular framework de desarrollo de código abierto creado por Google para construir aplicaciones móviles, web y de escritorio desde una sola base de código, es decir, el mismo código se puede ejecutar tanto en Android como en iOS. Fue lanzado por primera vez en 2017, pero desde 2018 fue en alza teniendo una comunidad más grande que los frameworks del mismo objetivo.

Flutter utiliza el lenguaje de programación Dart, que también fue desarrollado por Google. Dart es conocido por ser rápido y eficiente, y ofrece características de programación orientada a objetos y funcional, Dart compila a código nativo, es un lenguaje JIT (Just In Time), es decir, es amigable con Scripting como JavaScript o Python y no es necesario realizar el debug una y otra vez, además de ser de nivel alto. Flutter como es un lenguaje moderno posee una característica llamada Garbage Collector, esta permite hacer un uso eficiente de la memoria RAM, esta va liberando memoria conforme va dejando de ser usada, y reserva memoria que se va a usar, esto optimiza el desarrollo de las aplicaciones móviles.

Flutter se destaca por su enfoque en la creación de interfaces de usuario atractivas y personalizables, debido a su característica de hard reload. Flutter tiene un rendimiento nativo, es decir, es compilado a código nativo, por lo tanto, todas las instrucciones se ejecutan directamente por el procesador, logrando un mejor rendimiento. Utiliza un conjunto de widgets personalizables que permiten a los desarrolladores diseñar interfaces de usuario de alta calidad y rendimiento (Flutter, 2024).

Como indica (Collaguazo, Venegas, Guerrero, & Beltrán, 2022), una de las características más atractivas de *Flutter* es que no utiliza *widgets* nativos de *Android* o de *iOS*, sino usando

su propio código se crean los *widgets*, logrando integrar el código relacionado a los *widgets* dentro de la aplicación.

De acuerdo con Collaguazo et al. (2022), resume las funcionalidades más importantes de *Flutter*:

- Experiencia de usuario: *Flutter* incluye *Material Design* de *Google* y *Cupertino* de *Apple*, obteniendo interfaces de usuario idénticas a las de las aplicaciones nativas de cada sistema operativo desarrolladas por las propias compañías.
- Tiempo de carga: Una de las principales razones por las que los usuarios abandonan una aplicación es el tiempo de carga. Con *Flutter*, se logran tiempos de carga de menos de un segundo en dispositivos *iOS* y *Android*.
- Desarrollo ágil: Gracias a la característica *hot-reload*, se puede programar y observar los cambios en tiempo real en el emulador.

1.1.12 Aplicaciones móviles

Con la llegada de los teléfonos inteligentes, se ha impulsado el desarrollo de aplicaciones móviles que, aunque suelen ser breves, cumplen funciones esenciales para los usuarios (Bhardwaj, Saunders, Juneja, & Gavrilovska, 2019). En la era de la digitalización, estas aplicaciones han adquirido un papel crucial en la sociedad, ya que permiten realizar procesos cotidianos de manera eficiente (McIlroy, Ali, & Hassan, 2016)

Las aplicaciones móviles están diseñadas para funcionar en teléfonos inteligentes, tabletas y otros dispositivos móviles, y permiten a los usuarios llevar a cabo diversas tareas que facilitan la ejecución de programas y actividades. Por lo general, se distribuyen a través de plataformas específicas o empresas que poseen sistemas operativos móviles, como *Android*, *iOS*, *BlackBerry OS* y *Windows Phone* (Osuna, 2020).

Para su desarrollo, es fundamental contar con un framework adecuado que guíe el proceso y asegure la incorporación de los mejores elementos en términos de calidad (Ríos, Tapia, Souto, & León, 2021).

1.1.13 Aplicaciones híbridas

Las aplicaciones híbridas aprovechan la versatilidad del desarrollo web y se adaptan al dispositivo al igual que las aplicaciones nativas. Actualmente existen varias herramientas o bien *frameworks* que permiten desarrollar aplicaciones híbridas, incluido *React Native* usando *JavaScript* y *Flutter* usando *Dart*. Este tipo de aplicaciones son cada vez más utilizadas por los desarrolladores, lo que se traduce en mejoras en el proceso de desarrollo y en su rendimiento. A diferencia de las aplicaciones nativas, estas aplicaciones no están escritas en el idioma nativo del sistema, pero lo que suele suceder es que el código fuente se convierte a un código nativo o una vista simulada para ejecutarse sin causar inconvenientes (Osuna, 2020).

De acuerdo con Collaguazo et al. (2022), una de las principales complicaciones al crear aplicaciones móviles es la diversidad de dispositivos y sistemas operativos, lo que obliga a desarrollar una versión distinta para cada uno, utilizando diferentes lenguajes y herramientas. Debido al constante avance de la tecnología en especial de los dispositivos móviles, se ha requerido de manera urgente una herramienta que permita el desarrollo para adaptarse a estos sistemas operativos.

Según Nahuel Delía (2017), afirma que el principal reto de los desarrolladores de aplicaciones móviles es proporcionar soluciones para todas las plataformas porque conlleva un alto costo, debido a que no es posible reusar el código fuente de una aplicación para distintos sistemas operativos, lo que demanda un mayor esfuerzo y con ello se elevan los costos del desarrollo.

1.2 Marco conceptual

En esta sección se abordan los conceptos que son utilizados dentro del contexto de la aplicación, es decir, dentro del entorno de los transportes de carga.

1.2.1 Tractocamión

Un tractocamión o tracto camión es un vehículo comercial especializado para llevar mercancías pesadas y de gran tamaño. Se encuentra dentro de la categoría de vehículos de

grandes dimensiones y cuentan generalmente con varios ejes. Básicamente, un tracto camión es un tráiler sin caja. En la Figura 1.1 se observa un tractocamión.

Figura 1.1

Tractocamión Kenworth



Nota: La figura muestra un tractocamión T680 sin caja 2024 marca Kenworth. Fuente: PACCAR (2024).

1.2.2 Dolly Convertidor

Es un sistema de acoplamiento o eslabón que permite unir 2 semirremolques de arrastre, logrando desplazar mayor volumen de carga, reduciendo considerablemente costos, ya que sólo se emplea un tractocamión en vez de dos, obteniendo mayor eficiencia en tiempos en la industria del transporte (Gamma Corporation , 2024). En la Figura 1.2 se observa un dolly.

Figura 1.2

Dolly Corpus Trailers



Nota: La figura muestra un dolly tipo A de dos ejes marca Corpus Trailers. Fuente: (Gamma Corporation , 2024).

1.2.3 Remolque

El remolque se trata de un vehículo independiente, no motorizado, que ofrece un espacio extra para almacenar la carga. Este debe ser remolcado o arrastrado por el vehículo remolcador (tractocamión). También se puede conocer como remolque de enganche, y su uso está muy extendido entre los tractocamiones. Existe una amplia variedad de remolques, por ejemplo, de caja cerrada, cuello de cisne, cama plana, low boy, plataforma estándar, entre otros. En la Figura 1.3 se observa un remolque tipo plataforma estándar.

Figura 1.3

Plataforma estándar Corpus Trailers



Nota: La figura muestra una plataforma estándar de dos ejes Marca Corpus Trailers. Fuente: (Gamma Corporation , 2024).

1.2.4 Logística

En la literatura, el concepto de logística se define de muchas maneras y continúa evolucionando. Esto se debe a que la logística está teniendo una aplicación cada vez mayor en diversos sectores, incluyendo la logística del transporte.

De acuerdo con Lambert, Cooper, & Pagh (1998) indican a la logística de esta forma “es la parte de la gestión de la cadena de suministro (supply chain management, scm) que planifica, implementa y controla el flujo eficiente y efectivo de materiales y el almacenamiento de productos, así como la información asociada desde el punto de origen hasta el de consumo, con el objeto de satisfacer las necesidades de los clientes”.

Por otro lado el autor Ballou (2003), da un enfoque a la logística industrial “todo movimiento y almacenamiento que facilite el flujo de productos desde el punto de compra de los materiales hasta el punto de consumo, así como los flujos de información que se ponen en marcha, con el fin de dar al consumidor el nivel de servicio adecuado a un costo razonable”.

Como se observa en estas definiciones, los autores caracterizan la logística como el flujo continuo de productos desde su origen hasta el punto de consumo. Para alcanzar este objetivo, es crucial una adecuada organización de los procesos, lo que lleva a mejoras en la ejecución de actividades y, en consecuencia, a la reducción de costos para mantener la competitividad.

En este contexto, una cadena logística se compone de todas las actividades empresariales integradas funcionalmente, incluyendo el transporte, la gestión de almacenes e inventarios, los sistemas de información, y otras etapas como el despacho aduanero y los trámites bancarios. Esta integración funcional es clave para optimizar el flujo de productos y garantizar una logística eficiente.

1.2.5 Activos

Recursos y bienes que una empresa usa para mover y gestionar productos. Esto incluye cosas como vehículos, almacenes, equipos, inventario y tecnología. Estos activos ayudan a que los productos lleguen a los clientes de manera eficiente.

1.2.6 Línea

Dentro del sector de la logística de transporte una línea se les conoce a las líneas de transporte que también pueden ser comprendidas como las empresas dedicadas al transporte.

1.3 Estado del arte

El estado del arte representa una revisión de la literatura y las soluciones existentes en el área en que se desarrolla este sistema, además expone las tendencias actuales, los avances tecnológicos, las aplicaciones existentes, así como los desafíos y las oportunidades identificadas en el campo. Es una estrategia para guiar al lector hacia el contexto más amplio en el que se inscribe al trabajo.

1.3.1 Fleet Complete

Fleet Complete es una empresa que ofrece soluciones de gestión de flotas y seguimiento de activos para empresas y organizaciones que desean administrar eficazmente sus vehículos y activos móviles, fleet complete fue creada por Tony Lourakis. Esta empresa ofrece diversos servicios tales como: gestión de flotas, rastreo, resguardo de unidades, gestión de seguridad y gestión de entrega y envíos. En la Figura 1.4 se observa el logo de la empresa Fleet Complete.

Figura 1.4

Logo de Fleet Complete



Nota: Logo de empresa Fleet Complete. Fuente: (Fleet Complete, 2024).

Fleet complete brinda una plataforma en donde se puede conectar todo lo implicado en una empresa de transportes, desde empleados hasta activo, una empresa reconocida en toda norte América, siendo el proveedor de diversas empresas tales como Maple Leaf, Minto, Toronto, L&L Oilfield. Esta empresa ha sido durante años una de las más utilizadas por empresas con flotas medianas y grandes debido a la cantidad de servicios que ofrece, que sin duda alguna es la más completa del mercado. Algunas de las oportunidades que cubre Fleet Complete, es la fundamental en una flota, que es el seguimiento de la ubicación en tiempo real, permite ver la ubicación de todas las unidades desde cualquier dispositivo con conexión a internet, además, una de sus cualidades es que interviene con el mantenimiento de las unidades, esto mediante la obtención de alertas de fallas que permiten planificar su mantenimiento lo más rápido posible, lo que mantiene a las unidades en el estado óptimo para el trabajo. El único conveniente existente con Fleet Complete es que, al ser una empresa con raíces canadienses, la adquisición a países como México limita un poco la comunicación con asesores debido a que es un mercado menor. En la Figura 1.5 se observa la aplicación que brinda la empresa Fleet Complete para dispositivos móviles.

Figura 1.5

Representación gráfica de la interfaz de Fleet Complete



Nota: Dispositivos móviles con la aplicación ofrecida por Fleet Complete en ejecución.

Fuente: (Dianes, 2019).

1.3.2 Geotab

Geotab es una reconocida empresa de telemetría y gestión de flotas, con una sólida presencia a nivel mundial, habiendo sido nombrada tres veces el proveedor mundial número uno en telemática comercial. En la Figura 1.6 se observa el logo de la empresa Geotab.

Figura 1.6

Logo de Geotab



Nota: Logo de la empresa de Geotab. Fuente: (Geotab, 2024).

Su plataforma de gestión de flotas es utilizada por empresas alrededor del mundo para supervisar en todo momento sus unidades, así como mejorar la eficiencia de sus flotas y vehículos comerciales. Esta plataforma se destaca por ofrecer varias herramientas enfocadas en las flotas de vehículos, las funciones que ofrece son las siguientes:

- Seguimiento en tiempo real. Geotab permite rastrear la ubicación y estado de los vehículos, garantizando la seguridad de los operadores, optimizar rutas y tener una respuesta más rápida a situaciones impredecibles de carretera.
- La plataforma recopila datos sobre el comportamiento de los conductores y unidades, como la velocidad, frenado brusco o aceleración agresiva, dichos informes llegan a las personas encargadas para promover una conducción segura.
- Ofrece herramientas para programar mantenimientos, lo que reduce fallas dentro de su operabilidad, reduciendo así imprevistos de arreglo. Promueve el cumplimiento de la normatividad de la industria, como el tiempo de conducción y los registros de horas de servicio.

Como se visualiza en la Figura 1.7 las interfaces que ofrece Geotab son discretas y especializadas, lo que implica la instalación especial para un mejor rendimiento del dispositivo.

Figura 1.7

Representación de la interfaz gráfica de Geotab

Name	Odometer	Rate plan	Serial number
Truck ABC Best Group 1, Vehicle + Driver Group 05/09/19 13:57:32	45089 mi	Pro	09000000000
Dispatch 456 Admin 2 Only, Vehicle + Driver Group	124779 mi	ProPlus	09000000000
John Company group	13720 mi	ProPlus	09000000000
Dispatch 432 Vehicle + Driver Group 05/15/20 11:36:52	48917 mi	Pro	09000000000
Dispatch 123 Company group	120365 mi	ProPlus	09000000000
Dispatch 234 Company group	203654 mi	ProPlus	09000000000

Nota: Software que ofrece Geotab ejecutándose en un ordenador. Fuente: (Geotab, 2024).

Dentro de los casos de éxito de Geotab se pueden encontrar diversas empresas y organizaciones, tales como: Synergy, Ayuntamiento de Madrid, Volkswagen Truck & Bus México, Flota de la policía del estado de Utah y Southern Wine & Spirits of America, Inc., como se aprecia son distintos sectores de trabajo, lo cual no es un obstáculo para la

implementación de Geotab, además, la cantidad de unidades en las flotas no es un problema, administrando flotas desde 12 unidades hasta una de más de 649000 vehículos, de la empresa Enterprise Fleet Management, la cual se dedica a la renta y alquiler de vehículos.

Por último, Geotab no tiene barreras de los dispositivos en los que se desee tener la gestión de la flota, Geotab es un software multiplataforma, es decir, se puede ver en dispositivos móviles como en ordenadores, siempre y cuando se tenga el acceso a internet.

1.3.3 Sygic

Sygic es una destacada aplicación de navegación GPS que se ha ganado una sólida reputación en la industria de la navegación. Ofrece una amplia gama de funciones de navegación para conductores y peatones, lo que la convierte en una opción popular para usuarios de todo el mundo. Fue fundada en 2004 convirtiéndose en la primera empresa de navegación para iPhone y la segunda para Android. El logo de Sygic se muestra en la Figura 1.8.

Figura 1.8

Logo de Sygic.



Nota: Logo de la aplicación Sygic. Fuente: (Wisdom Factory, 2024).

Actualmente es conocida como la aplicación #1 en la navegación sin conexión del mundo, la aplicación número dos del mundo en la categoría de Navegación, con más de 200 millones de usuarios, la primera aplicación de navegación de terceros en CarPlay, Android Auto y Google Play Store para Android Automotive OS. Dentro de las principales características de Sygic destacan:

- **Navegación sin Conexión:** Una de las características más destacadas de Sygic es su capacidad de navegación sin conexión. Los usuarios pueden descargar mapas completos de regiones o países para usarlos sin necesidad de una conexión a Internet, lo que es útil en áreas con mala conectividad o cuando se viaja al extranjero.

- Mapas Precisos y Actualizados: Sygic utiliza mapas de alta calidad y precisión que se actualizan regularmente para reflejar cambios en las carreteras y la infraestructura. Esto garantiza rutas precisas y actualizadas.
- Información de Tráfico en Tiempo Real: La aplicación proporciona información de tráfico en tiempo real, lo que permite a los conductores evitar atascos y tomar rutas alternativas para llegar a su destino de manera más eficiente.
- Alertas de Límites de Velocidad y Radares: Sygic ofrece alertas de límites de velocidad actuales y radares de velocidad, ayudando a los conductores a mantenerse dentro de los límites legales y evitar multas por exceso de velocidad.
- Planificación de Rutas Avanzada: Los usuarios pueden planificar rutas personalizadas, evitar carreteras específicas, peajes o áreas congestionadas, y obtener estimaciones precisas de tiempo de llegada (ETA).
- Puntos de Interés (POI): La aplicación muestra puntos de interés cercanos, como restaurantes, estaciones de servicio, cajeros automáticos y más, lo que facilita la búsqueda de servicios esenciales durante los viajes.
- Navegación Peatonal: Además de la navegación en automóvil, Sygic ofrece instrucciones de navegación peatonal para explorar ciudades a pie.
- Realidad Aumentada (AR): Sygic incorpora una función de realidad aumentada que utiliza la cámara del teléfono para superponer direcciones y señales de navegación.

Se observa una aplicación bastante completa, dentro de sus ventajas destacan:

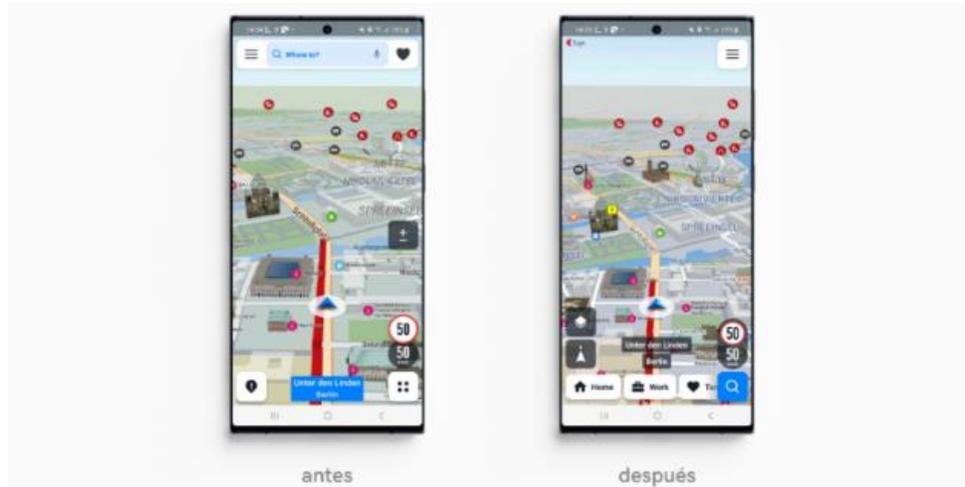
- Navegación sin Conexión
- Mapas Precisos y Actualizados
- Interfaz Intuitiva

Sygic es una aplicación de navegación confiable y versátil que ha ganado una gran base de usuarios debido a su enfoque en la navegación sin conexión, mapas precisos y un conjunto completo de características de navegación. Ya sea para uso personal o comercial, Sygic es una opción sólida para aquellos que buscan una solución de navegación GPS confiable en sus dispositivos móviles (Wisdom Factory, 2024).

En la Figura 1.9 se muestra la interfaz de usuario en una ruta.

Figura 1.9

Representación gráfica de Sygic.



Nota: La figura muestra cómo se ve Sygic ejecutándose en dispositivos móviles. Fuente: (Wisdom Factory, 2024).

En la Tabla 2.1 se presenta una serie de características de las aplicaciones presentadas respecto a la que se desarrolla en el presente documento, en donde, se puede apreciar que se abarcan diversas funciones de deferentes ámbitos que demanda el transporte, destacando la segmentación de las funciones por departamentos, por lo que, se determina que existe una oportunidad de competencia frente a las aplicaciones que dominan el mercado para la gestión de líneas de transporte pequeñas o medianas que carecen de una herramienta especializada.

A partir de la comparación de las aplicaciones mencionadas con la que se desarrolla en el presente proyecto, se determina que la aplicación planteada es competente para ser una herramienta en el sector de la logística de transporte, ofreciendo ciertas funcionalidades que no poseen las demás aplicaciones del mercado.

Tabla 2.1*Comparación de aplicaciones respecto a la aplicación de la tesis*

Característica	Sygie	Geotab	Fleet Complete	Freightem
Gestión de unidades de transporte.	✓	✓	✓	✓
Base de Datos local.	(Depende de plan)	(Depende de plan)	(Depende de plan)	✓
Registro de números de serie, placas y números de motor.	✓	✓	✓	✓
Asociación de unidades con operadores.	✓	✓	✓	✓
Registro de cambios de aceite y fechas.	✗	✓	✓	✓
Trabajo segmentando por departamentos.	✗	✓	✗	✓
Funciones para mantenimientos.	✗	✓	✓	✓
Funciones para viajes.	✓	✓	✓	✓
Acceso gratuito.	(Depende de plan)	(Depende de plan)	(Depende de plan)	✓
Inicio de sesión.	✓	✓	✓	✓
Llamadas directas a operadores desde la Aplicación.	✗	✗	✓	✓
Funcionamiento sin internet.	(Depende de plan)	(Depende de plan)	(Depende de plan)	✓
Interfaz de usuario intuitiva.	✓	✓	✓	✓

Capítulo 2. Metodología

Como ya fue presentado previamente la metodología seleccionada para basar el desarrollo de la aplicación móvil es la metodología RAD con el enfoque en prototipos evolutivos, de forma general esta metodología se enfoca en crear diseños funcionales de forma parcial, con el fin de que prototipo tras prototipo se vaya mejorando hasta llegar a satisfacer todas las necesidades y problemáticas identificadas.

2.1 Fase 1. Investigación preliminar

En la fase de investigación preliminar se determina el problema, importancia y efectos a la organización, se identifica una solución general para analizar qué tan factible es crear un software. Consta de tres actividades principales: La clasificación de requerimientos como actividad principal, se debe observar de forma analítica, identificando el área de oportunidad de una empresa, en donde se admite la intervención del usuario para completar el análisis.

Como segunda actividad el estudio de factibilidad, se determina si realizar el proyecto es viable tomando en cuenta aspectos como si es posible la realización con equipo actual, si refleja ser algo que valga la pena económicamente, es decir, que no refleje un gasto grande para las funciones que va a facilitar el desarrollo y si el software si será usado por los usuarios. Como tercera actividad la aprobación de requerimientos, se espera la aprobación del proyecto, esta decisión pertenece a los jefes de área habitualmente de las empresas.

Recordando que el problema a solucionar es la dificultad en la gestión de la información en situaciones remotas, es decir, que los empleados y encargados de departamentos que no se encuentren dentro de una oficina de trabajo, por lo tanto, no tienen fácil acceso a la información y se complica significativamente poder resolver las situaciones que conlleva el día a día del transporte, esto representa una oportunidad de mejora que es crear una aplicación móvil, a la que en adelante se le denominará como *Freightem*.

Para el desarrollo de la aplicación móvil *Freightem* se solicitará el apoyo a una empresa de transporte privada que tiene las siguientes características: tener una flota de 50 a 100 unidades, tener su información centralizada en un equipo de cómputo, no contar con una

herramienta especializada para la gestión de sus unidades y la más importante, estar dispuesta a trabajar en una modalidad cara a cara constante con el desarrollador.

Con una empresa para apoyar el desarrollo se hace hincapié por parte del departamento de administración de la empresa que esta aplicación debe de ser de fácil acceso, tanto a la aplicación como a la información, es decir, una aplicación intuitiva que no represente problema alguno en el aprendizaje para su manejo.

La propuesta planteada es una aplicación móvil en donde sea posible ver la información útil de las unidades de la empresa, es decir, de toda su flota, esta información comprende desde el número de serie de un tractocamión hasta el contacto telefónico de su operador, por lo que se concluye por la parte administrativa de la empresa que el desarrollo de la aplicación es una adquisición que mejorará la forma en que se gestiona la información, por lo tanto, el desarrollo de la misma es factible.

2.2 Fase 2. Definición de los requerimientos del sistema

La fase de definición de los requerimientos del sistema. Tomando en cuenta el análisis realizado previamente se definen todos los requerimientos del sistema y el desarrollador determina de forma final los requisitos mediante la construcción y demostración del primer prototipo, en este proceso los usuarios deben de brindar el contexto de sus necesidades al desarrollador, para que este comprenda la situación del usuario a la perfección. Esta fase comprende cinco etapas distintas:

En primer lugar, se lleva a cabo el análisis preliminar y la especificación, definiendo los requisitos incales del sistema, además, se diseña un esquema básico del primer prototipo que debe ser aprobado por el cliente, en este caso una persona de la empresa con la que se trabajará el desarrollo.

Una vez analizando las actividades que desempeña el campo laboral de los transportes se observan diversas situaciones que pueden ser facilitadas para su desempeño con el uso de una aplicación móvil. En este punto del análisis se hace una lista de los requerimientos del cliente para identificar cuáles son viables para satisfacer por medio de la aplicación.

Dentro de los principales requerimientos planteados se encuentra que la aplicación *Freightem* debe ser:

- Aplicación intuitiva.
- No contener un bombardeo gráfico.
- Ser capaz de buscar las unidades por número de tracto.
- Modificar los operadores de los tractos.

Estos requerimientos iniciales serán un punto clave para iniciar con el prototipado de la aplicación móvil.

2.3 Fase 3. Diseño técnico

A continuación, en la etapa de diseño y construcción, se desarrolla el prototipo con un enfoque prioritario en la funcionalidad básica y, particularmente, en la interfaz de usuario de forma inicial.

Posteriormente, en la etapa de evaluación, se verifica si el prototipo cumple con los requisitos establecidos. Este proceso se realiza con la participación de los usuarios, quienes prueban el sistema. Si se identifican fallos o deficiencias, el desarrollador debe ajustar el prototipo y entregarlo nuevamente. Este ciclo de prueba y ajuste se repite hasta que el prototipo sea considerado aceptable.

En la etapa de modificación, se realizan ajustes adicionales en el prototipo según la retroalimentación obtenida durante la evaluación, en caso de que el prototipo inicial no haya sido aceptado.

Finalmente, en la etapa de término, se considera que el prototipo ha alcanzado el estado deseado y cumple con los estándares requeridos y es capaz de proceder a la última etapa del diseño final.

Iniciando con las actividades previamente descritas:

Basándose en los requerimientos de la fase previa, se diseña con un boceto de interfaz de usuario, dicho boceto se observa en la Figura 2.1 para proceder a materializarlo en un software de diseño, por lo que, se entrega un prototipo de diseño que fue creado en el software

Adobe Illustrator, dichos diseños iniciales se representan en la Figura 2.2 y Figura 2.3. Los bocetos de la interfaz de usuario (UI) fueron aprobados por la empresa, por lo que se dio inicio a la creación de la aplicación por medio del framework Flutter.

- a) Diseño y construcción: Esta etapa aborda el primer prototipado, este se encuentra en el Capítulo 3 del presente documento.
- b) Evaluación: Dichas retroalimentaciones de cada prototipo realizado se encuentran en el Capítulo 3 del presente documento.
- c) Modificación: Los prototipos con los cambios solicitados se encuentran en el Capítulo 3 del presente documento como nuevos prototipos.
- d) Terminó: El prototipo final se encuentra en el Capítulo 3 del presente documento.

Figura 2.1

Boceto del diseño inicial de la aplicación



Figura 2.2

Propuesta del inicio de la aplicación.

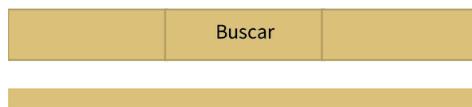
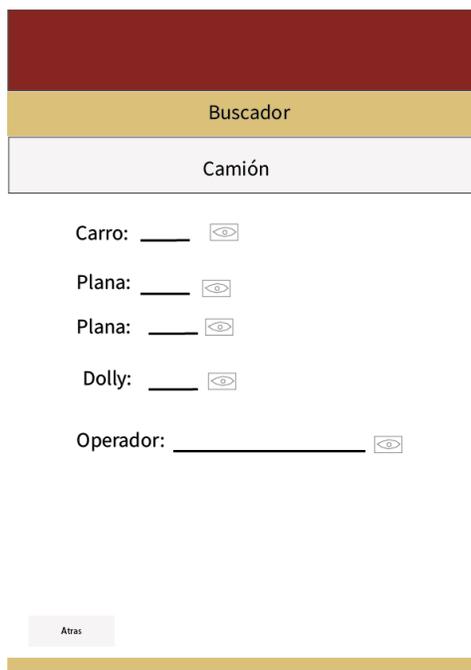


Figura 2.3

Propuesta del buscador de la aplicación



Como última actividad de esta fase, se dirige a rediseñar el sistema conforme a los estándares de la organización. En esta etapa, se produce la documentación necesaria tanto para el diseño como para los mantenimientos futuros del sistema.

En este punto del desarrollo de la aplicación móvil se ha enfocado en que los prototipos se desarrollen en la parte lógica con todos los requerimientos funcionales solicitados, por lo que, usando el designado como prototipo final se continua con hacer el diseño final, ya sin la necesidad de modificar la lógica de la aplicación, lo que conlleva a diseñar la aplicación con base en los estándares definidos por parte de la empresa que ha apoyado en el desarrollo de la aplicación móvil, que en este caso ha sido sugerido el uso de los siguientes colores:



ARGB (255, 134, 38, 33)



ARGB (255, 245, 245, 180)

Además, una animación fue agregada al momento de abrir la aplicación, dado que muestra simpleza en ese aspecto, por lo que, se opta por usar el logo de la aplicación, que ya ha sido agregada a los *assets*, dicho logo ha sido diseñado utilizando el software *Adobe*.

Con el objetivo de desarrollar una aplicación móvil de la más alta calidad, es imprescindible corregir las faltas de ortografía presentes en la misma. Esta atención al detalle es fundamental para garantizar un producto final que cumpla con los estándares de calidad esperados. Adicionalmente, es importante utilizar iconos apropiados que representen de manera clara las funciones que ofrece la aplicación, lo que contribuirá a mejorar la experiencia del usuario.

Por último, se realiza una documentación técnica sobre el código, indicando las funcionalidades del código usando comentarios detallando configuraciones e infraestructura del mismo.

Haciendo los cambios necesarios se completa la tercera fase de la metodología, cumpliendo con los estándares solicitados de la empresa, por lo que ahora se puede considerar como producto final a la aplicación, lo que conlleva a la siguiente fase.

2.4 Fase 4. Desarrollo de pruebas

Esta fase de la metodología se presenta en el Capítulo 4 del presente documento, en donde, se muestran los resultados de encuestas realizadas a usuarios designados para probar el funcionamiento de la aplicación *Freightem*.

2.5 Fase 5. Operación y Mantenimiento

La fase de operación y mantenimiento consta de la instalación del sistema y la operatividad de los usuarios, para este punto del desarrollo ya han usado la aplicación móvil en fases previas, por lo que, no supone un problema la capacitación. Si eventualmente un mantenimiento es requerido se vuelve a las fases del prototipado con los nuevos requerimientos planteados.

Para probar la operatividad de la aplicación *Freightem*, se consideró prestar el servicio de la aplicación móvil a la empresa de transporte privada que apoyó para el desarrollo de la aplicación móvil. Además, con la prueba de usabilidad se dio a conocer que la aplicación tiene un rápido aprendizaje por parte de los usuarios nuevos.

Una vez instalada la aplicación móvil directamente por parte del desarrollador, se somete la aplicación a uso continuo de los usuarios finales, estos usuarios son designados por la empresa, dado que se contiene información de todas las unidades que puede resultar sensible. Cabe resaltar que el funcionamiento en un ambiente laboral real que tenga diferentes dispositivos para usar la aplicación presenta una limitación en las funcionalidades que comprenden de la información brindada por otro departamento, o bien desde otro dispositivo móvil, por lo que en esta fase se limitó al uso las funciones de la aplicación *Freightem* que no implica la comunicación con otros usuarios. A continuación, se representan tres casos en los que ha sido usada la aplicación dentro del día a día de la empresa privada.

2.5.1 Caso uno

En marzo de 2024 se hizo la compra de 8 tractocamiones en la ciudad de Puebla, en donde se debieron agregar dichas unidades a la base de datos con la numeración 86 a 93. En la Figura 2.4 se observan los tractocamiones adquiridos, mientras que en la Figura 2.5 se muestra cómo agregaron una de las unidades adquiridas, en este caso es el económico 86.

Además, se hizo también la compra de su respectiva plataforma estándar para poder operar, esta plataforma obtuvo el número 135, la agregación de esta plataforma usando *Freightem* se observa en la Figura 2.6 y, por último, se contrató a un nuevo operador para la unidad 86 en cuestión, el registro en la base de datos de este operador se muestra en la Figura 2.7.

Figura 2.4

Compra de tractocamiones



Figura 2.5

Registro de las nuevas unidades mediante Freightem

A screenshot of a mobile application form titled "Agregar Tracto". The form contains the following fields and values:

Número*	86
Serie	
Motor	
Año	2023
Placas*	
Nota	
Modelo	T600
Línea	Fratsa
Marca	Kenworth
Tipo	Full
Clase	Tabiquero
Operador	SIN OPERADOR
Dolly Número	49

At the bottom of the form, there are five icons with labels: "Operador", "Tracto", "Plana", "Dolly", and "Extras". The "Tracto" icon is highlighted in blue.

Este proceso fue realizado múltiples veces por cada una de las unidades adquiridas.

Siendo de importancia mencionar que por confidencialidad de la empresa con el desarrollo de la aplicación móvil se solicitó mantener privados datos relevantes, por lo que, ciertos datos han sido censurados para respetar el acuerdo de privacidad.

Figura 2.6

Registro de nueva plataforma en Freightem

← Agregar Plana

Número*
135

Serie

Año
2023

Placas*
[Blurred]

Ejes
3

Pies
10

Marca
Corpus Christi

Tracto

Corpus Christi

Guardar Plana

Operador Tracto Plana Daily Extras

Figura 2.7

Registro de operador en Freightem

The screenshot shows a mobile application interface for adding an operator. The title bar is dark red with a back arrow and the text 'Agregar Operador'. The form consists of several input fields with labels and horizontal lines below them. The 'Nombre del Operador*' field contains the text 'CRUZ REYES'. The 'Licencia del Operador:' field is empty. The 'Teléfono del Operador*' field contains the number '72'. The 'Contacto del Operador:' field is empty. The 'Parentesco del Operador:' field contains the text 'Hermana'. Below the form is a dark red button with the text 'Guardar Operador'. At the bottom of the screen is a navigation bar with five icons: a person (Operator), a truck (Tracto), a plane (Plano), a dolly (Dolly), and three dots (Extras).

2.5.2 Caso dos

En abril de 2024, la unidad 25 se dirigía a Bacalar, Quintana Roo, en dicha ruta sufrió un incidente por otro vehículo durante la noche, por lo que una persona designada fue enviada para solucionar temas con la aseguradora y el ajustador, para ello, fue necesario contar con datos como número de póliza, número de serie, nombre del conductor, tipo de vehículo, marca y modelo. En la Figura 2.8 se muestra la consulta realizada a los datos del tractocamión para resolver con la aseguradora y el ajustador.

Figura 2.8

Consulta de datos para rescate de la unidad 25



2.5.3 Caso tres

En abril de 2024, la unidad 44 sufrió un incidente por otro tractocamión, por lo que se designó a una persona para ir a solucionar temas con la aseguradora y el ajustador, y al igual que en el caso anterior. se haría una consulta a los datos necesarios, en la Figura 2.9 se muestra el accidente mencionado y en la Figura 2.10 se muestra la consulta realizada a los datos del tractocamión.

Figura 2.9

Accidente de la unidad 44



Figura 2.10

Consulta de datos de la unidad 44

← Ver Económico: 44 ✎

Datos generales:

Serie	Motor
1	
Año	Placas
2016	
Modelo	Línea
T800	
Marca	Tipo
Kenworth	Full
Clase	Dolly
Tabiquero	
Aseguradora	Número de póliza
Vencimiento	
Datos extras:	
Descripción	
Nota	

✎

Una vez completada y demostrada la fase de operatividad se puede concluir que ha sido implementada de forma correcta la aplicación móvil siendo usada en los casos donde era necesario contar con la información de los tractocamiones de forma remota.

Por otro lado, no ha sido posible llegar a la parte del mantenimiento de la aplicación móvil debido a que durante el tiempo de desarrollo e implementación no fue requerido un mantenimiento a la aplicación *Freightem*. Con lo que se finaliza la aplicación de la metodología de prototipos evolutivos teniendo un producto funcional y en operatividad.

Capítulo 3. Desarrollo

En este capítulo, se describirán los prototipos desarrollados según la metodología de prototipos evolutivos, así como la retroalimentación obtenida a través de iteraciones del prototipado.

3.1 Prototipo 1

Este primer prototipo marca el inicio del proceso de prototipado. En esta etapa, se incorporan diversas herramientas necesarias para la funcionalidad de la aplicación móvil, como el diseño y la creación de la base de datos.

De acuerdo con la metodología, el primer prototipo debe centrarse en tener una interfaz de usuario básica con las funcionalidades solicitadas. El prototipo inicial, se basa en el boceto mostrado en la Fase 2 de la metodología, en este se presenta una ventana de inicio, como se muestra en la Figura 3.1, con un par de botones. Al hacer clic en el botón de búsqueda, se despliega una ventana donde se pueden buscar las unidades únicamente por su número económico, junto con sus elementos asociados. Esta ventana de búsqueda se ilustra en la Figura 3.2. Cabe mencionar que la búsqueda se hace únicamente por el número económico del tractocamión.

Figura 3.1

Inicio del primer prototipo

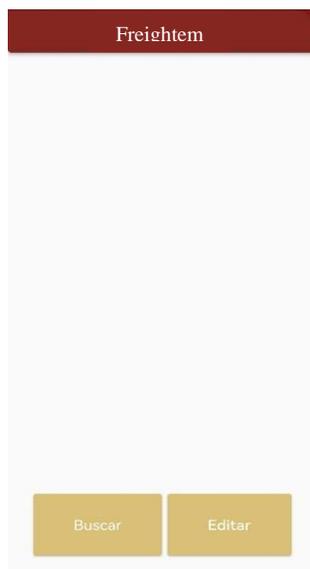


Figura 3.2

Ventana de búsqueda

The image shows a mobile application search interface. At the top, there is a dark red header with a back arrow and the text 'Freightem'. Below the header, there are several search criteria, each with a dropdown menu and an eye icon to its right. The criteria are: 'Carro' with a dropdown set to '1'; 'Plana 40"' with a dropdown set to 'Plana 1'; 'Plana 38"' with a dropdown set to 'Plana 1'; and 'Dolly' with a dropdown set to 'Dolly 1'. Below these, there is a section for 'Operador' with a dropdown set to 'Operador 1'. At the bottom right of the search area, there is a yellow button with the text 'Buscar'.

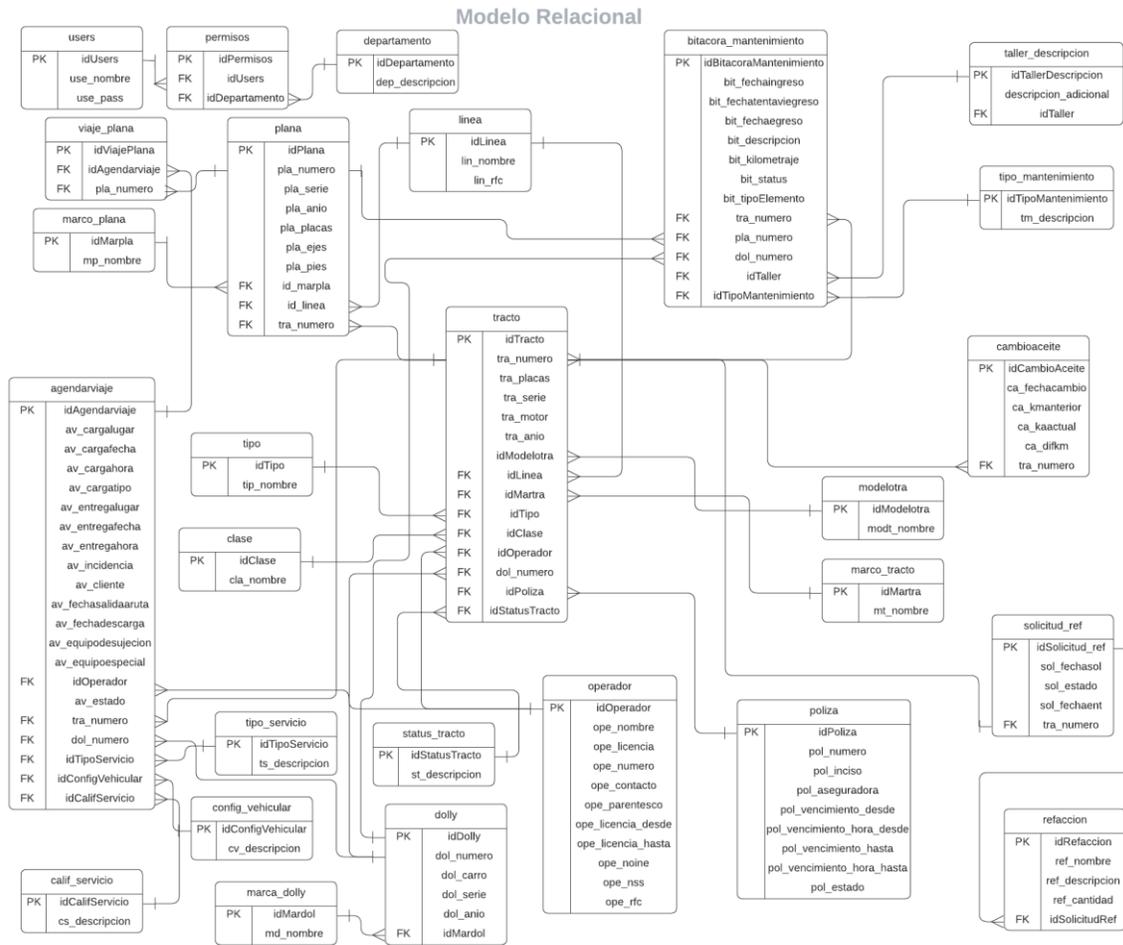
3.1.1 Modelo relacional

Como se ha mencionado previamente, al diseñar una base de datos por primera vez, es crucial contar con herramientas que faciliten la visualización de la estructura de la base desde diferentes perspectivas, verificando así la integridad de los datos antes de uso. En este caso, se utiliza el modelo relacional para este propósito, antes de proceder con la creación de la base de datos. La elaboración de este modelo es de suma importancia, ya que muestra la relación y cardinalidad de los datos, lo que contribuye a mantener la integridad de la base.

El modelo relacional correspondiente se muestra en la Figura 3.3.

Figura 3.3

Modelo relacional de la aplicación móvil Freightem



3.1.2 Diseño de Base de Datos

Una vez obtenido el modelo, se procede al diseño de la base de datos utilizando el lenguaje SQLite en el software *DB browser for SQLite*. Para lograr que la base de datos sea funcional y pueda recibir valores, es necesario procesar la información obtenida. Este proceso implica varias etapas. En primer lugar, se digitaliza la información proveniente de hojas de papel. Posteriormente, se establece un formato estandarizado para unificar la información.

Continuando con el proceso, se procede a eliminar la información que ya no es necesaria. Esta información obsoleta incluye unidades que han sido vendidas o robadas y que aún están en los registros. Después de consultar con la empresa, se identificaron 19 registros de esta

naturaleza, incluyendo ocho tractocamiones robados y 11 vendidos. Por lo tanto, esta información fue eliminada.

A continuación, se presenta un fragmento del esquema de base de datos relacional que se trabaja en este trabajo:

linea(id_linea, lin_nombre, lin_rfc).

marca_plana(id_plana, mp_nombre).

marca_tracto(id_martra, mt_nombre).

marca_dolly(id_mardol, md_nombre).

tipo(id_tipo, tip_nombre).

clase(id_clase, cla_nombre).

modelotra(id_modelotra, modt_nombre).

operador(id_operador, ope_nombre, ope_licencia, ope_telefono, ope_contacto,
ope_parentesco).

dolly(id_dolly, dol_numero, dol_serie, dol_anio, id_mardol).

tracto(id_tracto, tra_numero, tra_serie, tra_motor, tra_anio, tra_placas,
tra_nota, id_modelotra, id_linea, id_martra, id_tipo, id_clase, id_operador,
dol_numero).

plana(id_plana, pla_numero, pla_serie, pla_anio, pla_placas, pla_ejes, pla_pies,
id_marpla, id_linea, tra_numero).

cambioaceite(id_cambioaceite, ca_kmanterior, ca_kmactual, ca_difkm, tra_numero).

La información proporcionada por la empresa se presentó en dos formatos diferentes. Los detalles sobre tractos, plataformas (en adelante también llamadas planas) y dollys se entregaron en hojas de papel, mientras que la información sobre los operadores se proporcionó en un archivo de Excel.

Posteriormente, se procede a unificar la información útil, aplicando una transformación en cuanto al formato. Finalmente, se carga la información para llevar a cabo las pruebas correspondientes. Estas pruebas consisten en realizar consultas solicitadas por la empresa, con el objetivo de verificar la integridad de los datos con sus respectivas relaciones entre las diferentes tablas, por ejemplo, hacer consulta para identificar en cuál unidad está asociado una plana. Con esto, se demuestra la ventaja de implementar tecnología en las empresas de transporte.

Es importante mencionar que las *Primary Keys* de cada tabla son el atributo común 'id' dentro de las relaciones de las tablas. Sin embargo, en el contexto de uso de los datos, resulta más útil identificar los tractos, dollys y planas con base en el atributo 'número'. Esto se ilustra en la Figura 3.4, donde se muestra que es más ágil identificar estos elementos en función de dicho atributo.

Figura 3.4

Muestra del identificador de las unidades físicas



Siguiendo la premisa de que el número es el identificador principal en el día a día, se debe realizar una adaptación lógica para garantizar su utilidad, respetando las reglas de sintaxis de SQL. Por lo tanto, este atributo se indexa para facilitar la creación de relaciones basadas en

el número del tracto. Este mismo proceso se repite para el dolly y las planas, ya que se sigue el mismo concepto para identificarlos.

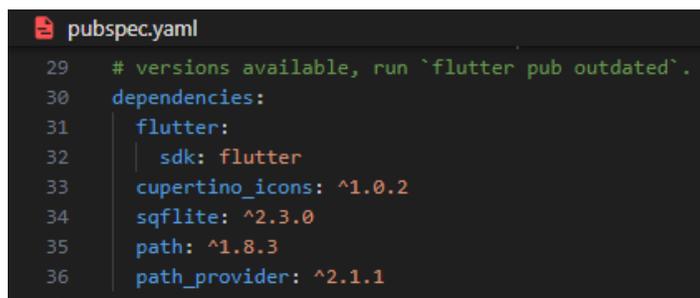
3.1.3 Conexión con la Base de datos

La aplicación necesita obtener la información que se pretende mostrar, por lo que simultáneamente a la creación del primer prototipo se diseña y se crea la base de datos. Siguiendo la documentación ofrecida por Flutter, se opta por utilizar SQLite, que es una versión de SQL adaptada para dispositivos móviles. Esta base de datos se agrega como un recurso dentro de la aplicación, por medio de una carpeta denominada *assets*, lo que significa que es una base de datos que se aloja dentro del dispositivo móvil al momento de instalarse, por ende, es una base de datos local.

Una vez creada la base de datos y diseñada una interfaz de usuario útil, se procede a implementarlas utilizando una dependencia ofrecida por *Flutter* llamada *sqlflite*. Esta dependencia debe ser agregada dentro de los archivos de desarrollo, específicamente en el archivo *pubspec.yaml*. En la Figura 3.5 se muestra cómo se realiza esta agregación de la dependencia. Es importante destacar que se deben agregar las versiones indicadas dentro de la documentación de *Flutter* para obtener una mejor experiencia de desarrollo, esta documentación se encuentra en el sitio oficial de *Flutter*.

Figura 3.5

Implementación de la dependencia *Sqlflite*



```
pubspec.yaml
29 # versions available, run `flutter pub outdated`.
30 dependencies:
31   flutter:
32     sdk: flutter
33   cupertino_icons: ^1.0.2
34   sqlflite: ^2.3.0
35   path: ^1.8.3
36   path_provider: ^2.1.1
```

Con la implementación de la dependencia, es posible conectar la base de datos de manera local con la aplicación. En este caso, la base de datos se agrega como parte del paquete de instalación de la aplicación. Para interactuar de manera lógica con la base de datos, se optó

por crear un archivo llamado *DatabaseHelper*, en el cual se almacenan todas las consultas a la base de datos como funciones, así mismo, el código para la conexión a la base.

3.1.4 Retroalimentación del prototipo

Una vez presentado el prototipo uno, se presta al uso de los usuarios finales, en donde, tras días de ser probado se concluye en la retroalimentación con nuevos requerimientos para el siguiente prototipo:

- Otro tipo de navegación, no ser necesario dar clic en un botón para poder buscar una unidad.
- Funcionamiento de las ventanas de edición, principalmente la de los operadores que no han sido creadas.
- Definir un icono para la aplicación.

3.2 Prototipo 2

En este momento, es crucial abordar las solicitudes de la empresa, especialmente la necesidad de mejorar la navegación dentro de la interfaz de usuario (UI). Se ha identificado como principal objetivo la optimización del proceso de búsqueda de información, dado que la rapidez en las consultas es uno de los principales motivos de desarrollar una aplicación móvil. La actual navegación que consta en presionar botones para realizar una búsqueda contradice este objetivo fundamental de agilidad y accesibilidad.

En respuesta a esta observación, se ha llevado a cabo un rediseño de la interfaz de usuario de la ventana inicial de la aplicación móvil *Freightem* para lograr una pantalla inicial más fácil de usar y de realizar las búsquedas de las unidades. Este nuevo diseño se enfoca en eliminar la dependencia de botones para iniciar las búsquedas, lo que permitirá a los usuarios acceder más rápido a la información deseada. De forma lógica, se hace una consulta a la base de datos para obtener todas las unidades por medio de su número económico y se muestran en una lista siendo ordenados por el mismo atributo, en donde, se agrega en la consulta por medio de un *LEFT JOIN* el nombre del operador para que esa sea la información que se muestre en primer instancia cuando se abra la aplicación móvil.

Además, se agrega un buscador para las unidades, el cual realiza la búsqueda por medio del número económico de la unidad para facilitar el método en que se realizan las búsquedas, esta implementación se realiza pensando en la experiencia de usuario al momentos de tener demasiadas unidades en existencia.

Este rediseño se muestra en detalle en la Figura 3.6. Debido a temas de confidencialidad de la empresa se solicitó censurar la información, por lo que, toda información delicada de la empresa ha sido censurada.

Figura 3.6

Segundo prototipo del inicio



Como segunda modificación, se ha abordado el funcionamiento de las ventanas de edición, cuyo acceso se realizaba mediante botones, tal como se observa en la Figura 3.7. Estas ventanas permiten la modificación de todos los elementos asociados a cada unidad, con especial énfasis en la capacidad de editar la información del operador, solicitud resaltada por la empresa. Los elementos y características de cada unidad se presentan detalladamente en la Figura 3.8.

La última modificación del prototipo dos consistió en agregar un icono de la aplicación, reemplazando así el logo predeterminado del *framework Flutter*. Esta implementación del logo como icono se logró utilizando la dependencia *Flutter_launcher_icons* y agregando el archivo para usar como icono a la carpeta de *assets*. Para llevar a cabo esta modificación, se

debe actualizar el archivo *pubspec.yaml*, como se muestra en la Figura 3.9, además de definir en el mismo archivo la ruta del icono como indica la documentación de la dependencia *Flutter_launcher_icons*.

Figura 3.7

Acceso a las ventanas de edición en el prototipo dos



3.2.1 Retroalimentación del prototipo

Posterior a las modificaciones realizadas, el prototipo fue entregado a los responsables de la empresa para que hicieran las observaciones sobre el nuevo prototipo entregado. Durante esta revisión, se identificaron los siguientes nuevos requerimientos:

- Facilitar el acceso de las ventanas de editar, es decir, no tener una pantalla únicamente de botones.
- Agregar las opciones de agregar y de borrar registros de trailer, plana, dolly y operador, manteniendo un diseño atractivo como el nuevo de la edición.

Figura 3.8

Ventana de edición de los tractos



Figura 3.9

Agregación de la dependencia Flutter_launcher_icons

```
dev_dependencies:  
  flutter_test:  
    sdk: flutter  
  url_launcher: ^6.0.12  
  flutter_launcher_icons: ^0.13.1  
  
  flutter_lints: ^2.0.0  
flutter_launcher_icons:  
  android: "launcher_icon"  
  ios: true  
  image_path: "assets/icon/fratza_icon.png"
```

3.3 Prototipo 3

Se prosiguió con la resolución de las solicitudes de la empresa para continuar mejorando la aplicación. Como primer cambio, se solicitó evitar el uso de una pantalla llena de botones.

En respuesta, se optó por reutilizar las pantallas en las que se muestran los detalles, de modo que al pulsar el botón de edición se habilitaran las características para su modificación dentro de la misma ventana. Esta dinámica se implementó de manera uniforme en todas las pantallas de detalles, ya sea para tractocamiones, planas, dollys (en caso de estar relacionados) u operadores. Este cambio tuvo el beneficio directo de optimizar el uso de recursos al utilizar la misma ventana para dos fines, de esta forma, evitando el uso de demasiados botones únicamente para editar.

Por otra parte, el segundo cambio de este prototipo consiste en agregar las funciones de agregar y eliminar tractos, planas, dollys, operadores y marcas. Con el fin de evitar un exceso de botones, se optó por hacer uso de un *widget* llamado *Drawer*. Este *widget* permite habilitar un botón que despliega una barra lateral temporal, en la cual se agregan los vínculos a las ventanas de "agregar" y "eliminar".

Con el objetivo de simplificar el uso de la aplicación, se implementó un *widget* denominado *bottomNavigationBar*. Este *widget* permite agregar un menú en la parte inferior de la pantalla, donde se incluyen todas las opciones solicitadas para agregar. Tomando como referencia la acción de agregar un tracto, se procede lógicamente a crear un modelo lógico del tracto. Este modelo permite obtener los datos ingresados por el usuario y mapearlos para que sean ingresados a la base de datos mediante el lenguaje SQLite. Este proceso debe ser realizado para todas las opciones solicitadas, incluyendo tracto, plana, dolly, operador, modelos y marcas.

Las ventanas de agregar se muestran en la Figura 3.10, donde se aprecia el *widget bottomNavigationBar* mencionado. Este *widget* se implementa para evitar la necesidad de movilizarse entre múltiples botones, concentrando todas las opciones de agregar en una sola pantalla. Del mismo modo, todas las opciones de eliminar se concentran en su respectiva pantalla, que se muestra en la Figura 3.11.

3.3.1 Retroalimentación del prototipo

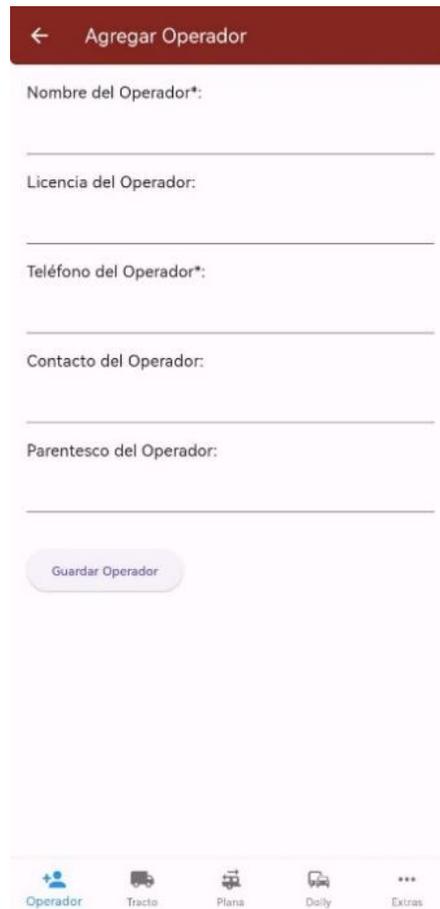
Al entregar el prototipo tres se concluyó en una única petición por el momento.

- Agregar un icono de llamada al operador de cada unidad el cual, sea capaz de marcar el número de cada operador para comunicarse directamente desde la

aplicación. No debe realizar la llamada, si no que solo marque el número dentro de la aplicación del dispositivo móvil, lo que permite la opción de elegir telefonía para realizar la llamada.

Figura 3.10

Prototipo de la ventana de agregar operador



Prototipo de la ventana de agregar operador. La interfaz muestra un encabezado con un botón de retroceso y el título "Agregar Operador". El formulario contiene cinco campos de texto con las siguientes etiquetas: "Nombre del Operador*", "Licencia del Operador:", "Teléfono del Operador*", "Contacto del Operador:", y "Parentesco del Operador:". Debajo de los campos hay un botón "Guardar Operador". En la parte inferior, una barra de navegación muestra cinco íconos con sus respectivos textos: "Operador", "Tracto", "Plana", "Dolly", y "Extras".

Figura 3.11

Prototipo de la ventana de eliminar operador

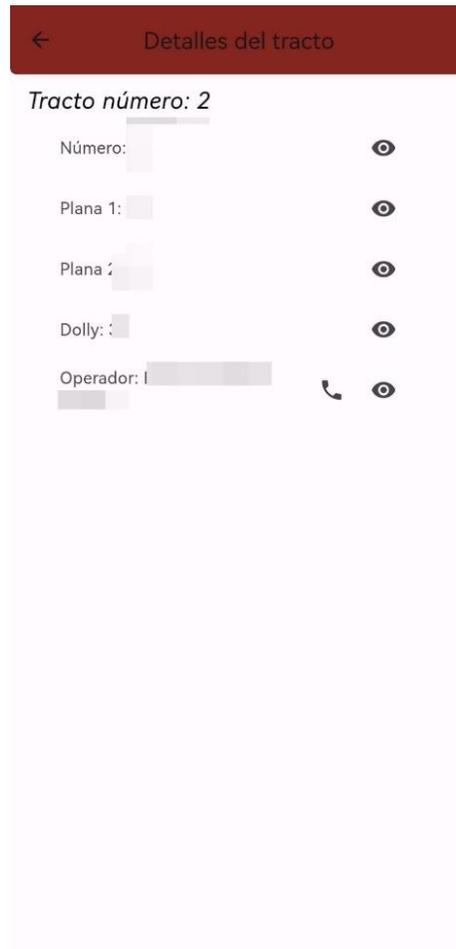


3.4 Prototipo 4

En este prototipo, se implementó el botón solicitado, aprovechando las capacidades del *framework* para interactuar con el sistema operativo. Se añadió un ícono de teléfono, el cual, al presionarlo, redirige a la aplicación nativa de llamadas con el número del operador ya ingresado en el teclado numérico. Esta implementación se muestra en la Figura 3.12. Se logra dinámicamente mediante consultas directas al atributo del teléfono del operador.

Figura 3.12

Prototipo cuatro con la implementación de la llamada al operador



3.4.1 Retroalimentación del prototipo

Una vez implementado se entregó el prototipo a la empresa y se solicitaron nuevos requerimientos, los cuales fueron los siguientes:

- Agregar una pantalla previa de inicio de sesión, con usuarios específicos, es decir, que ya hayan sido creados.
- Agregar la función de registrar cambios de aceite con kilometrajes como parámetros importantes, además, que estos mismos cambios se pueda ver como historial por tracto con su respectiva fecha, además, que muestre valores como kilometraje anterior y diferencia entre kilometrajes.

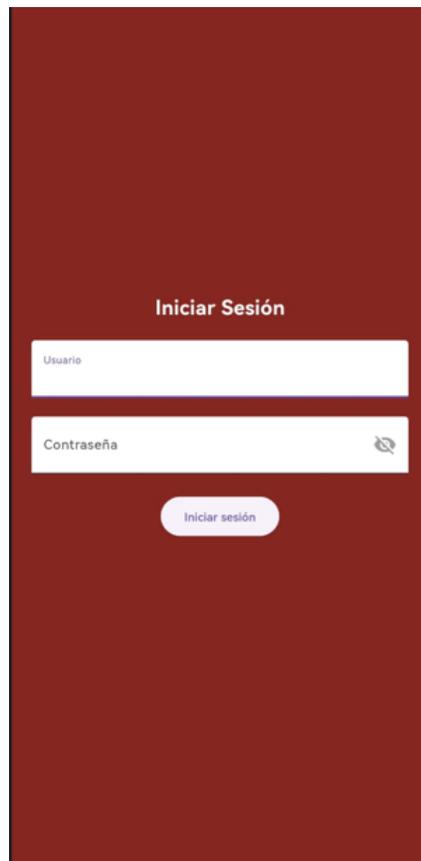
3.5 Prototipo 5

Como primer requerimiento en la última entrega, se solicitó la implementación de un inicio de sesión. Este cambio en la lógica de la aplicación consiste en seleccionar una nueva ventana para que sea la ventana de inicio de la aplicación móvil y realizar el inicio de sesión en la misma. Además, como segundo requerimiento, la función de cambio de aceite implica una modificación en la base de datos existente, donde se deben agregar las tablas necesarias, en este caso *user* y *cambio_aceite*.

En la nueva ventana de inicio de sesión se deben de validar credenciales en la base de datos, por lo que se hace una consulta y en caso de ser correctas las credenciales de usuario y contraseña se habilita el acceso a la aplicación, en caso contrario, se arroja un mensaje de que las credenciales no son válidas. La ventana de inicio de sesión se muestra en la Figura 3.13, mientras que la ventana de cambio de aceite se observa en la Figura 3.14.

Figura 3.13

Quinto prototipo en la ventana de iniciar sesión



El prototipo muestra una pantalla con un fondo rojo oscuro. En el centro, el título "Iniciar Sesión" está escrito en blanco. Debajo del título hay dos campos de entrada blancos con bordes redondeados. El primer campo está etiquetado como "Usuario" y el segundo como "Contraseña". El campo de contraseña tiene un ícono de ojo que indica la opción de mostrar u ocultar la contraseña. Debajo de los campos de entrada hay un botón redondeado con el texto "Iniciar sesión" en blanco.

Por parte de registrar cambios de aceite, se deben de hacer consultas a la base de datos para mostrar el kilometraje el tractocamión en cuestión, además, por medio de un campo de texto ingresar un nuevo kilometraje validando que sea mayor al existente. Al guardar el cambio de aceite se hace un *update* a la base de datos, de esta forma logrando registrar un cambio de aceite. Por último, un historial de los cambios de aceite que es una consulta de los cambios de aceite agrupando por el número del tractocamión.

Figura 3.14

Quinto prototipo en la ventana de cambio de aceite



The image shows a mobile application interface for changing oil. At the top, there is a dark red header with a back arrow and the text "Cambio de aceite". Below the header, the title "Cambio de aceite para la unidad 4" is displayed. There are two input fields: "Kilometraje anterior: 0" and "Kilometraje actual". A blue button labeled "Guardar Cambio de Aceite" is positioned below the input fields. At the bottom, there is a section titled "Historial de cambios de aceite:" which is currently empty.

3.5.1 Retroalimentación del prototipo

Una vez implementados los requerimientos solicitados, se entregó el prototipo a la empresa y se solicitaron nuevos requerimientos, los cuales fueron los siguientes:

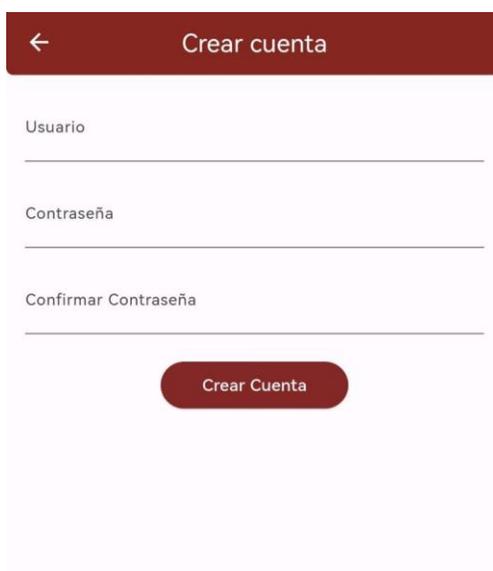
- Crear una ventana para añadir usuarios, dado que se proporcionaron usuarios limitados para validar las credenciales de acceso.
- Agregar una exportación de reporte de cambios de aceite en un archivo pdf.

Prototipo 6

Iniciando con los nuevos requerimientos solicitados en el prototipo cinco, se debe crear una ventana para agregar usuarios desde la ventana del inicio de sesión, por lo que la nueva ventana contiene los campos de texto para el nombre del usuario junto con su contraseña más la validación de la contraseña. De forma lógica, se ejecuta un *insert* a la base de datos con los datos ingresados. Esta ventana se observa en la Figura 3.15.

Figura 3.15

Sexto prototipo en la ventana de crear usuario

El prototipo muestra una interfaz de usuario para crear una cuenta. Tiene un encabezado rojo oscuro con un ícono de retroceso y el texto "Crear cuenta". Debajo hay tres campos de texto con las etiquetas "Usuario", "Contraseña" y "Confirmar Contraseña". En la parte inferior hay un botón redondeado rojo oscuro con el texto "Crear Cuenta".

← Crear cuenta

Usuario

Contraseña

Confirmar Contraseña

Crear Cuenta

El segundo requerimiento solicitado fue la posibilidad de realizar reportes de los cambios de aceite, por lo que, dentro de la ventana de cambio de aceite se agrega un botón con el icono de compartir, este botón es conocido como *FloatingActionButton*. Seguido a esto, se agregan las dependencias necesarias para poder manipular archivos pdf, la agregación de las dependencias se observa en la Figura 3.16 dentro del archivo *pubspec.yaml*.

En donde cada una de las dependencias agregadas son cruciales para poder realizar los reportes en pdf, de forma general, la dependencia *intl* se utiliza principalmente para la internacionalización, que incluye el manejo de formatos de fecha, número y moneda, entre otros, la dependencia *pdf* para la creación del archivo pdf, la dependencia *share_plus* para realizar el envío del archivo generado, la dependencia *printing* también para manejar los

archivos pdf y poder nombrarlos, y por último, la dependencia *google_fonts* para definir el tipo de fuente utilizado en el reporte.

Figura 3.16

Agregación de dependencias para generar reportes pdf

```
dependencies:  
  flutter:  
    sdk: flutter  
  intl: ^0.19.0  
  pdf: ^3.10.8  
  share_plus: ^9.0.0  
  printing: ^5.12.0  
  google_fonts: ^6.2.0
```

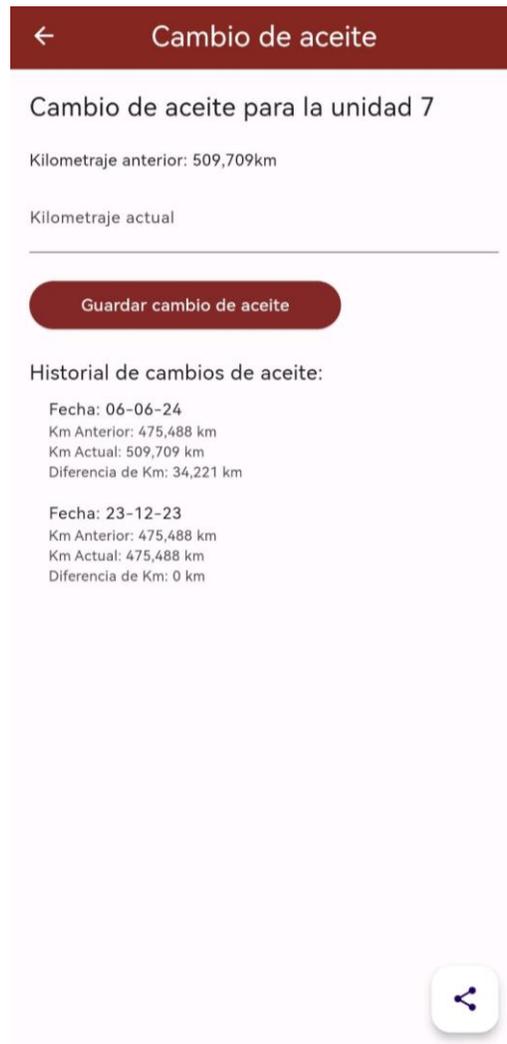
Para crear el reporte es necesario crear una función que haga una consulta a la base de datos específica para obtener los valores deseado a mostrar en el reporte, posteriormente seguir la documentación de la dependencia *pdf* para seguir de forma correcta el proceso de la generación de columnas y celdas.

El reporte es generado al dar clic en el *FloatingActionButton*, para después mostrar las opciones del dispositivo para compartir el archivo directamente, esto se logra gracias a la dependencia *share_plus*.

En la Figura 3.17, se observa el botón agregado a la ventana de registrar cambio de aceite, en seguida en la Figura 3.18, se observa la ventana emergente para compartir el reporte pdf y en la Figura 3.19 se observa el reporte generado en pdf, este reporte consta de una tabla que contiene los números de los tractocamiones con sus kilometrajes registrados, en caso de no haber tenido un cambio de aceite registrado el atributo de kilometraje anterior y el de kilometraje actual presenta el mismo valor. Con estas adiciones a la aplicación móvil se cumplen los requerimientos solicitados en la retroalimentación del prototipo anterior, por lo que se procede a la evaluación del presente prototipo.

Figura 3.17

Prototipo seis en ventana de cambio de aceite con generar reporte



3.6.1 Retroalimentación del prototipo

Una vez implementados los últimos requerimientos se solicitan nuevos requerimientos:

- Agregar una opción de solicitar refacciones, junto con un historial de las mismas.
- Agregar una ventana para reasignar las planas de los tractocamiones.

Figura 3.18

Prototipo seis en la ventana de compartir el reporte generado



Figura 3.19

Reporte generado de los cambios de aceite

Reporte de Cambios de Aceite

Tracto	Fecha	Km anterior	Km actual	Dif km
2	29-11-2023	299,338 km	299,338 km	0 km
3	09-03-2024	568,849 km	568,849 km	0 km
4	20-04-2024	432,318 km	432,318 km	0 km
6	29-12-2023	1,117,687 km	1,117,687 km	0 km
7	06-06-2024	475,488 km	509,709 km	34,221 km
9	08-12-2023	456,938 km	456,938 km	0 km
10	26-12-2023	1,134,102 km	1,134,102 km	0 km
11	13-01-2024	572,295.4 km	572,295.4 km	0 km
12	20-11-2023	803,199.8 km	803,199.8 km	0 km
13	06-10-2023	965,158.7 km	965,158.7 km	0 km
14	11-12-2023	770,570 km	770,570 km	0 km
15	01-03-2024	1,084,820 km	1,084,820 km	0 km
17	09-04-2024	912,897.3 km	912,897.3 km	0 km
18	12-12-2023	889,916 km	889,916 km	0 km

Prototipo 7

Abordando el nuevo requerimiento sobre las solicitudes de refacciones, se decide por agregar el acceso a la pantalla dentro del mismo menú de los cambios de aceite dentro de los detalles de los tractocamiones, este requerimiento demanda un cambio a la base de datos, el cambio consiste en agregar dos nuevas tablas, para permitir que una solicitud contenga varias refacciones. Las tablas llamadas *solicitudes* y *refaccion*, teniendo una relación de muchos a uno (N:1), respectivamente.

De forma lógica, se debe de crear un elemento *refaccion* por cada refacción agregada a una solicitud, una solicitud puede contener una o varias refacciones, cada refacción representa un *insert* a la base de datos, por lo que, se deben de agregar las claves primarias de las refacciones como claves foráneas a al *insert* de la solicitud. Para mostrar las solicitudes se agregó dentro de la misma ventana de la solicitud una sección del historial de refacciones de cada unidad, por consiguiente, se hace una consulta a la base de datos incluyendo un “*GROUP BY*” para mostrar de forma correcta las solicitudes con todas sus refacciones solicitadas, para mejorar la experiencia de usuario al momento de visualizar las solicitudes se usó un *widget* denominado *Card*, el cual le da un aspecto más agradable a cada registro evitando tener el texto plano en la pantalla. La forma en la que se muestran los detalles de cada solicitud es por medio de un dialogo, al momento de pulsar un registro, se muestran los detalles de cada refacción de la solicitud deseada.

Esta nueva ventana se observa en la Figura 3.20, donde se aprecia la sección de generar la solicitud de la refacción o refacciones, y en la parte inferior de la pantalla el historial de refacciones solicitadas del tractocamión en cuestión.

Continuando con el segundo requerimiento, la ventana de reasignar planas seguirá un esquema lógico similar al de la ventana inicial de la aplicación, en la que se muestran los tractocamiones, pero para la nueva ventana se debe modificar la consulta a la base de datos, en esta consulta se debe de consultar el atributo del número de las planas y el número económico de su tractocamión asociado. Los registros obtenidos se muestran utilizando el *widget Card*, en donde, se define como atributo principal a mostrar el número de la plana y como atributo secundario el número económico de su tractocamión asociado.

Figura 3.20

Ventana de solicitar refacciones con historial de solicitudes



En esta nueva ventana se mostrarán las planas existentes, y al dar clic en un registro emergerá un dialogo que muestra el número del tractocamión asignado junto con la opción de modificar el tractocamión, por medio de una lista se muestran los tractocamiones existentes para seleccionar el nuevo tractocamión a asignar a la plana previamente seleccionada, al seleccionar el nuevo tractocamión se realiza el *update* en la base de datos, de esta forma modificando el tractocamión asociado de la plana seleccionada.

Esta ventana se observa en la Figura 3.21.

Figura 3.21

Ventana de reasignación de planas



3.7.1 Retroalimentación del prototipo

Una vez implementados los últimos requerimientos se presta la aplicación *Freightem* a los usuarios finales, y se solicitan nuevos requerimientos:

- Agregar una opción para compartir los datos entre usuarios.
- Crear una ventana en la que se muestren todas las solicitudes de refacciones generadas con filtros de estado de pendientes y entregadas.

Prototipo 8

Iniciando con el requerimiento de agregar una opción para compartir datos entre usuarios, se decide que la opción más viable para lograr compartir datos entre usuarios finales es exportar e importar la base de datos, esto es logrado gracias a la implementación de nuevas dependencias que en este caso es la dependencia *file_picker* y *permission_handler*, ya que

las demás dependencias que son necesarias ya han sido implementadas anteriormente. Este requerimiento demanda conocimientos más técnicos sobre el manejo y navegación de archivos internos del dispositivo que serán aclarados en el Anexo A del presente documento. La agregación de las dependencias mencionadas junto con las necesarias para exportar e importar la base de datos se observa en el Figura 3.22.

Figura 3.22

Agregación de dependencias para exportar e importar la base de datos

```
dependencies:
  flutter:
    sdk: flutter
  sqflite: ^2.3.0
  path: ^1.8.3
  path_provider: ^2.1.1
  file_picker: ^8.0.3
  permission_handler: ^11.3.1
  share_plus: ^9.0.0
```

La navegación para acceder a las ventanas que contienen la opción de compartir datos se agregó en el *drawer*, en un apartado de “ajustes”, en este punto el *drawer* contiene las opciones de “agregar”, “eliminar”, ”reassignar plana” y la nueva implementación de los “ajustes”, esta última contiene los botones para exportar e importar las bases de datos junto con unas recomendaciones. De forma lógica, se debe de obtener la ruta en la que ha sido creada la base de datos dentro del dispositivo, posteriormente, se le da un nombre y se comparte. Por otro lado, la importación de la base de datos demanda implementar una petición de permiso para acceder a los archivos internos del dispositivo, por lo que se debe de seguir la documentación brindada por *Flutter* para solicitar permisos en los diferentes sistemas operativos. Cuando se permite la navegación de los archivos internos del dispositivo y se selecciona el archivo con extensión *.db* este se aloja dentro de la ubicación definida para la base de datos, por medio de la dependencia *path*, logrando de esta forma compartir la base de datos entre los usuarios.

En la Figura 3.23 se observa la opción de exportar la base de datos con su leyenda que hace la recomendación de la ruta en la que se hace la exportación con un mensaje que sugiere el

manejo delicado de la base de datos, por otro lado, en la Figura 3.24 se observa la opción de importar la base de datos, en donde se sugiere una ruta a buscar en caso de haber sido compartido por medio de una aplicación de mensajería en específico, cabe mencionar que solo en la importación de la base de datos es donde se requiere solicitar permisos para acceder a los archivos del dispositivo.

Figura 3.23

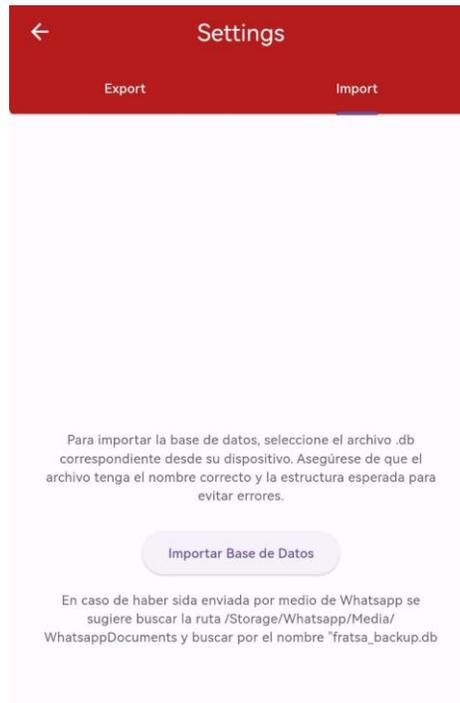
Prototipo ocho en la ventana de exportar la base de datos



Posteriormente, el segundo requerimiento solicitado ha sido agregar una ventana para poder ver todas las solicitudes realizadas con un filtro de estado. Esta opción será agregada en el *drawer* junto con el botón de “ajustes” del requerimiento anterior, para lograr el desarrollo de la nueva ventana se aplicó un diseño para identificar las solicitudes de refacciones que ya hayan sido entregadas y las pendientes por medio del uso de un identificador de color verde y rojo. Además, en el filtro se consideran las opciones de “todos” como predeterminada, “entregado” y “pendiente”.

Figura 3.24

Prototipo ocho en la ventana de importar la base de datos



De forma lógica, la aplicación de los filtros de estado demanda la agregación del atributo de estado a la tabla de las solicitudes, lo que conlleva a modificar la base de datos. Es de suma importancia mencionar que se agrega a la definición del atributo de estado lo siguiente: “*DEFAULT 'pendiente'*”, lo que indica que las solicitudes nuevas obtendrán el estado de pendiente automáticamente, con este atributo de estado se trabaja el uso del filtro, en donde al cambiar la opción del filtro se modifica la consulta a la base de datos de las solicitudes de refacciones, únicamente cambiando el parámetro “*WHERE estado = []*” en la respectiva consulta, logrando así, filtrar las solicitudes de refacciones por su estado.

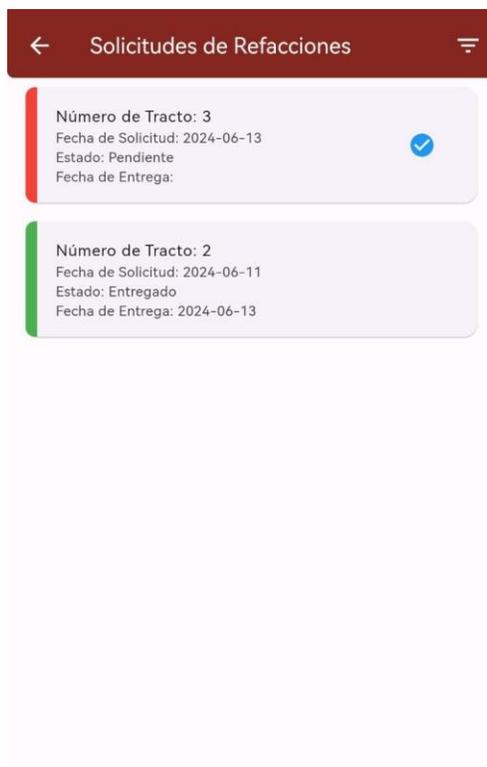
Se agregó también un icono dentro de los *card* de las solicitudes para marcar la solicitud como “entregada” lo que hace un *update* a su estado de la solicitud y también, se obtiene automáticamente la fecha para agregarse en el atributo de fecha de entrega completando así todos los atributos de una solicitud.

Como atributos principales se muestran la fecha, el número de solicitud y su estado que han sido considerados como los más relevantes. Esta nueva ventana se observa en la

Figura 3.25.

Figura 3.25

Prototipo ocho en la ventana de solicitudes de refacciones



3.8.1 Retroalimentación del prototipo

Una vez implementados los últimos requerimientos, se presta al uso de los usuarios el prototipo ocho, resultando en nuevos requerimientos introduciendo al departamento de ventas al uso de la aplicación móvil, este departamento es el encargado de negociar los viajes con los clientes

- Facilitar el acceso a la función de modificar el operador de cada unidad.
- Agregar nuevos datos importantes sobre los operadores, tales como: vigencia de la licencia, número de la INE, número de seguridad social y RFC.
- Crear una opción de agendar viajes con campos relevantes sobre el origen de carga, el destino para la entrega y datos extras.

Prototipo 9

Comenzando con el nuevo prototipo se abordan los requerimientos de la retroalimentación, iniciando por facilitar el acceso a la función de modificar los operadores de cada unidad, en donde se opta por agregar una ventana específicamente para modificar los operadores de cada unidad, recordar que la anterior forma de modificar el operador de cada unidad era modificando el tractocamión, lo cual resultó ineficiente y en palabras de un usuario “una función rebuscada”.

Para mantener como prioridad el fácil acceso se decide agregar esta función “Asignar operador” en el *drawer*, por lo que, se crea una interfaz simple que consta de consultar las unidades con su operador a la base de datos y mostrar las unidades por medio del *widget Card* en donde el atributo principal es el número económico de la unidad y el secundario es el nombre del operador que está relacionado a la unidad.

Para realizar la modificación se hace uso de un diálogo que se activa al dar clic a la *card*, esto se logra mediante el uso del *widget GestureDetector* para detectar el clic sobre la *card* en cuestión, mantenido el *id* de la unidad que se selecciona para poder hacer el *update* mediante ese atributo, el dialogo despliega todos los operadores existentes, posteriormente se hace el *update* a la base de datos con el *id* del operador seleccionado.

La navegación para acceder a esta ventana es por medio del *drawer*, la ventana para asignar el operador se observa en la Figura 3.26.

Cabe mencionar que en este punto del desarrollo, se utilizará una base de datos con registros del desarrollador, debido a que la empresa no proporcionó los datos necesarios para la implementación de las funciones con sus datos privados, por lo tanto, las funciones no pueden ser puestas a prueba por el desarrollador previamente de sus entregas a los usuarios. Esta base de datos consta de datos aleatorios que son usados únicamente con fines de pruebas.

Figura 3.26

Prototipo nuevo en la ventana para asignar operador

← Asignar Operador

🔍 Ingrese el número de tracto

Número: 101	Operador: MIGUEL ÁNGEL PÉREZ
Número: 102	Operador: HIGINIO PÉREZ
Número: 103	Operador: ERICK PÉREZ
Número: 104	Operador: JORGE MUNGUÍA
Número: 105	Operador: DANIEL PÉREZ
Número: 106	Operador: JUAN PÉREZ
Número: 107	Operador: PABLO MUNGUÍA
Número: 108	Operador: EMILIO SOSA

↻

Continuando con el segundo requerimiento solicitado, se requiere hacer una modificación en varios aspectos para poder agregar los nuevos datos sobre los operadores. El primer cambio consiste en modificar la base de datos, agregando los nuevos atributos que en su mayoría son de tipo *TEXT*, una vez teniendo la nueva base de datos, se procede a modificar el modelo de la tabla *operador* dentro del *IDE*, recalcando que se deben de agregar en los mapeos de la misma forma que fueron agregados en la base de datos para que exista integridad en los datos.

En este punto, se debe de modificar la lógica en la que se muestran las interfaces, principalmente la de agregar operador y la de ver detalles del operador debido a que los operadores contienen nuevos atributos que no existen en esas ventanas que ya han sido desarrolladas.

En la Figura 3.27 se observa la ventana de agregar operador actualizada con los nuevos datos, de esta forma, cumpliendo con este requerimiento.

Figura 3.27

Prototipo nueva en la ventana de los detalles del operador



El último requerimiento solicitado es crear una función de agendar viaje para el departamento de ventas. Este requerimiento demanda modificar la base de datos para agregar las tablas necesarias, después de analizar la situación se determina que son necesarias varias tablas para lograr agendar viajes, una para el viaje, la segunda asociar las planas de cada viaje y diversas tablas para detallar más los viajes, por ejemplo, una tabla para alojar las configuraciones vehiculares que pueden ser solicitadas, el tipo de servicio solicitado y la calificación del servicio, de las tablas mencionadas se utilizará sus *ids* para la tabla principal de los viajes. Una vez teniendo la base de datos actualizada se continua con crear un modelo lógico de la tabla *agendar_viaje* en el *IDE*.

Continuando con la interfaz por la cual se agendarán los viajes, es una interfaz simple con campos de texto en donde se ingresen los datos necesarios para los viajes tales como: origen,

destino, cliente, mercancía, tipo de servicio, configuración vehicular, diferentes fechas para la carga y descarga, por último, datos adicionales como el equipo de sujeción y equipo especial, como se ha mencionado, algunos de los detalles de los viajes se asociarán por medio de las claves primarias, en estos atributos se muestra por medio de un *DropDownButton*, este *widget* mostrará en forma de lista las opciones de la tabla en cuestión, por ejemplo, las configuraciones vehiculares posibles. además, se utiliza un calendario para seleccionar las fechas, este calendario es brindado por parte de la dependencia *timezone*, la agregación de esta dependencia se muestra en la Figura 3.28.

Figura 3.28

Agregación de la dependencia timezone

```
dependencies:  
  flutter:  
    sdk: flutter  
  timezone: ^0.9.4
```

La dependencia ayuda a obtener la fecha del dispositivo para aplicarla en los calendarios y brindar una mejor experiencia de usuario al seleccionar las fechas. Por otro lado, la ventana de agendar viajes se muestra en la Figura 3.29, cabe mencionar que esta información es llenada por el departamento de ventas, este departamento es el encargado de negociar con el cliente.

De forma lógica, se debe de obtener todos los campos definidos como obligatorios para poder hacer el *insert* de cada viaje, este *insert* consta de instanciar el objeto *agendar_viaje* para que al momento de construir dicho objeto se relacionen con las variables usadas para obtener los datos ingresados por el departamento de ventas.

De esta forma habiendo cumplido con los requerimientos solicitados en la retroalimentación anterior, lo que conlleva a entregar el presente prototipo a los usuarios finales para esperar su retroalimentación.

Figura 3.29

Ventana de agendar viaje

← Agendar viaje para la unidad 2

Datos Generales

Origen*

Destino*

Cliente*

Mercancia*

Tipo de servicio*
_____ ▼

Configuración vehicular*
_____ ▼

Datos de Carga

Fecha de Carga*

Hora de Carga

Fecha de salida a ruta

Datos de Entrega

Fecha de Entrega*

Hora de Entrega

Fecha de descarga*

Datos Adicionales

Equipo de sugestión

Equipo especial
_____ 

3.9.1 Retroalimentación del prototipo

Una vez implementados los últimos requerimientos y habiendo probado la aplicación *Freightem*, se demandan una secuencia de acciones relacionadas a los viajes, se pretende que sea trabajado por medio de los diferentes departamentos que pueden intervenir, es decir, el departamento de ventas, como ya se ha mencionado es el encargado de negociar los viajes con los clientes, entonces, este el encargado de agendar los viajes. Posteriormente, los viajes agendados pasan al departamento de tráfico, el cual, debe de asignar una unidad a los viajes dependiendo de los criterios solicitados, esto por medio de la disponibilidad de las unidades. Resultando en nuevos requerimientos que consiste en agregar varias ventanas para:

- Ver la disponibilidad de las unidades con identificadores de estado (tránsito, vacío, descanso y taller).
- Asignar unidad a los viajes.
- Ver los viajes pendientes de asignar.
- Ver los viajes activos.
- Ver los viajes completados.

Y por último:

- Agregar una opción para generar reportes de viajes.
- Implementar un manejo de notificaciones como recordatorio de las entregas de los viajes al inicio de la jornada laboral definida como las 8:00 horas, es decir, que cuando se agende un viaje, el día definido como día de entrega se programe una notificación en el día en cuestión para recordar el viaje.

Prototipo 10

Una vez recibida la retroalimentación del prototipo anterior, después de analizar todos los requerimientos solicitados, se determina por comenzar con la modificación a la base de datos, se debe de agregar a la tabla de *agendar_viaje* un atributo de estado, también, agregar un atributo de estado a los tractocamiones, que tendrá valores de “tránsito”, “vacío”, “descanso” y “taller”, con el fin de conocer su estado de disponibilidad.

Además, un atributo importante a considerar es definir atributo de estado del viaje con un valor predeterminado de “pendiente”, para que al crear un viaje, este obtenga el estado de pendiente para que inicie su ciclo de vida del viaje, esto se logra utilizando la siguiente sintaxis “*DEFAULT ‘pendiente’*” seguido a la definición del atributo de estado. De la misma forma, se define como atributo predeterminado el estado de los tractocamiones, con un valor de “vacío”, este valor se entiende como que la unidad está disponible para que se le asigne un viaje, de esta forma, todas las unidades se existentes definen como “vacío”, además, con los nuevos tractocamiones que sean registrados por medio de *Freightem*, obtendrán el estado de “vacío”.

Comenzando con los requerimientos del departamento de tráfico, para crear la interfaz de la disponibilidad de las unidades se recicla el código de la ventana de búsqueda de unidades, pero, se debe modificar la consulta a la base de datos, a esta se agrega el atributo del estado, identificando y diferenciando a cada uno de sus estados mediante un icono. Logrando con esto visualizar todas las unidades junto con su estatus de disponibilidad.

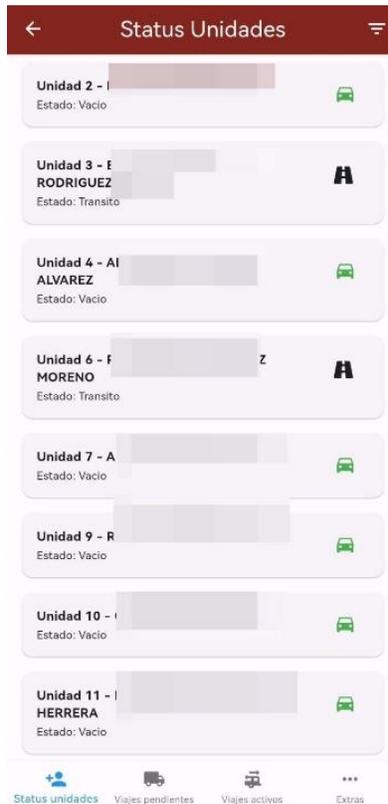
Para poder asignar un icono dependiendo del estado obtenido de la base de datos, se debe crear una función que sea capaz de igualar los estados obtenidos y asignar un icono para dicha cadena de texto obtenida, estos iconos para mejorar la experiencia de usuario son muy representativos al estado que quieren representar.

Dentro de la ventana de estatus de unidades se agregó un filtro en las búsquedas con base en los estados, para facilitar la visualización de las unidades deseadas, o en su defecto, ver el estado de disponibilidad de toda la flota. Esto se logra haciendo una consulta a la base de datos, pero cambiando el parámetro “*WHERE estado = []*” dependiendo de los estados identificándolos mediante su *id*, en este caso en específico hecho de forma manual dado que son únicamente 4 estados posibles lo que no complica el uso manual de las *ids*. Las opciones en este filtro de búsqueda son todos los estados posibles, y como vista predeterminada es la opción de “todos”.

La ventana de los estados de disponibilidad de las unidades se observa en la Figura 3.30.

Figura 3.30

Prototipo diez en la ventana de los estatus de las unidades



El siguiente requerimiento a abordar es asignar una unidad a los viajes, en esta ventana se hace una consulta a la base de datos para obtener los viajes agendados, pero definiendo el estado de “pendiente” en la consulta, con esto se obtienen los viajes que no poseen una unidad asignada, recordar que estos son brindados por el departamento de ventas.

Para asignar la unidad, se hace uso de un botón que despliega un dialogo, en este dialogo se muestran las unidades disponibles, es decir, que tienen el estado de “vacío”, junto con otro parámetro de filtración proporcionado por el departamento de ventas, este parámetro es la configuración vehicular, que existen dos, “sencillo” y “full”, esta configuración vehicular es un atributo de los tractocamiones por lo que se hace una consulta agregando ese condicional para mostrar las unidades en el dialogo.

Al asignar una unidad, esta pasa a obtener el estatus de “tránsito” y el viaje pasa a la ventana de viajes activos con el estado de “activo”, cabe mencionar que los elementos relacionados al tractocamión son relacionados directamente en cada viaje, es decir, al seleccionar un

tractocami3n se relaciona el operador, planas y dolly que tenga relacionado ese tractocami3n en el momento de asignar el viaje, esto 3ltimo se logra por medio del *insert* a una tabla que aloja las planas usadas en el viaje, una vez realizado ese *insert*, se obtiene el *id* de esa transacci3n y se agrega al *update* del viaje, en donde se ha actualizado con el tractocami3n asignado, el operador y dolly relacionados al tractocami3n seleccionado.

La ventana de los viajes pendientes se observa en la Figura 3.31, por otro lado, el dialogo para asignar una unidad al viaje se observa en la Figura 3.32.

Figura 3.31

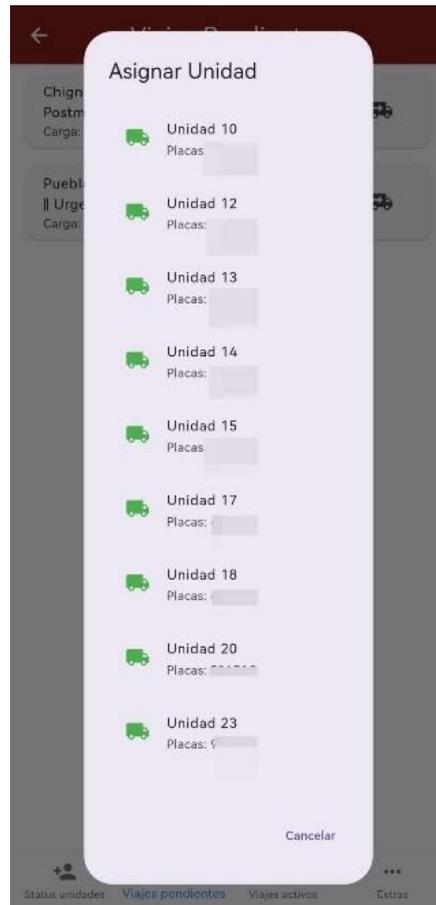
Prototipo diez en la ventana de viajes pendientes



El siguiente requerimiento a abordar es la ventana de viajes activos, en esta ventana se recicla el c3digo de la ventana de viajes pendientes, editando un par3metro de la consulta a la base de datos, el par3metro de estado del viaje debe de cambiar a “activo”, con lo que, se obtienen la informaci3n del viaje en su totalidad, aunque para una mejor visualizaci3n de los detalles se debe de agregar las cl3usulas *JOIN* para comprender la informaci3n obtenida en los casos de las claves for3neas. Nuevamente se hace uso del *widget card* para la visualizaci3n de los registros obtenidos en la base de datos.

Figura 3.32

Diálogo de asignar unidad a viaje



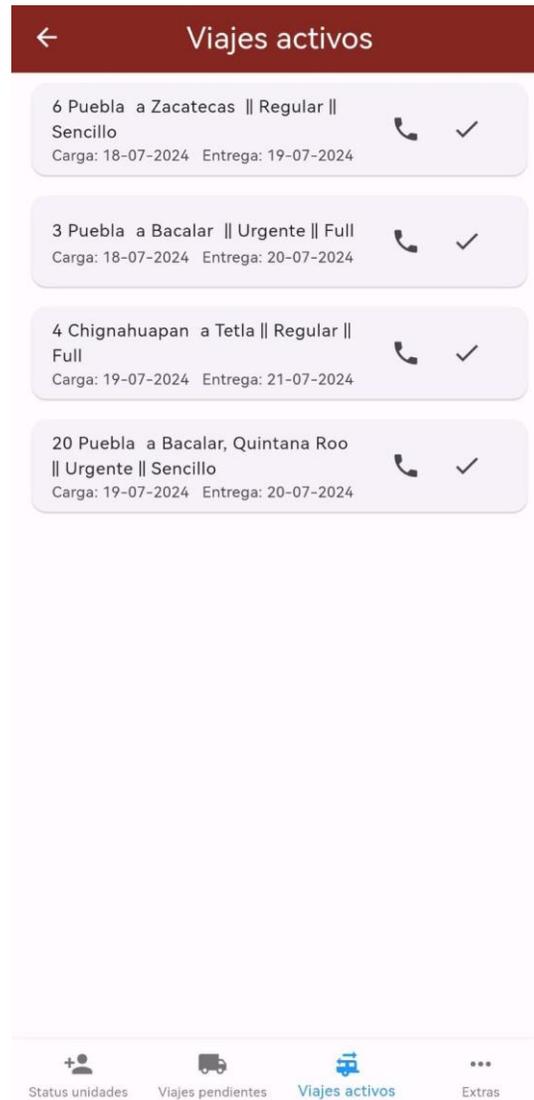
La interacción en esta ventana ha sido para fomentar el seguimiento del viaje y comunicación con el operador para el departamento de tráfico, por lo que, se decidió implementar un icono de llamada para poder llamar al operador del viaje en cuestión. La otra función que tiene el departamento de tráfico en esta ventana es el marcar el viaje como entregado, con esto, el viaje cambia su estado a “completado” y la unidad asignada para dicho viaje cambia su estado a “vacío”, el viaje que ha sido completado pasa a la ventana de viajes completados que es al cargo del departamento de ventas.

De forma lógica se aplican dos *update*, uno a la tabla del viaje y el segundo a la tabla del tractocamión para lograr el cambio en los estados en ambos casos.

La ventana de viajes activos se observa en la Figura 3.33.

Figura 3.33

Prototipo diez en la ventana de viajes activos



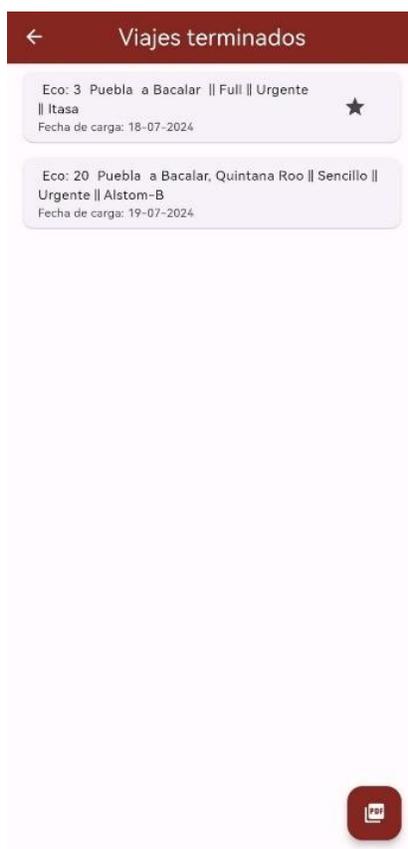
Continuando con los requerimientos solicitados, la ventana de viajes terminados para el departamento de ventas. Se recicla el código de la ventana de viajes pendientes, modificando el parámetro de consulta del estado del viaje por “completado”. En esta ventana se tiene la función de calificar el viaje, debido a la existencia de incidencias en el viaje o con el operador, se le asigna una calificación al viaje. Con esta última función se da por terminado el ciclo de vida de un viaje, lo que conlleva al último requerimiento solicitado que es generar reportes de viajes. De forma lógica, este cambio implica un *update* al registro del viaje en cuestión, en donde se modifica el atributo de calificación del viaje.

Para generar reportes de viaje es necesario agregar dependencia que ya han sido agregadas previamente para la función de generar reportes de cambio de aceite. Se determina que el parámetro de los reportes será “por unidad”. Se definen las columnas de “Cliente”, “Origen”, “Destino”, “Configuración Vehicular”, “Fecha de carga”, “Fecha de descarga”, “Operador” y “Calificación de servicio”, este reporte muestra el seguimiento de una unidad lo que representa múltiples propósitos de monitoreo para la empresa.

La ventana de viajes terminados que contiene también el botón de generar reportes de observa en la Figura 3.34.

Figura 3.34

Prototipo diez en la ventana de viajes terminados



Cumpliendo con los requerimientos solicitados y atendiendo las solicitudes detalladas por los departamentos de ventas y tráfico se completa el prototipo 10, lo que conlleva a entregar el mismo a los usuarios finales para verificar la funcionalidad correcta de la aplicación móvil.

3.10.1 Retroalimentación del prototipo

Una vez implementados los últimos requerimientos se demandan una secuencia de funciones relacionadas al departamento de mantenimiento. Como el trabajo ya ha sido segmentado por medio de departamentos, las funciones solicitadas para que el departamento de mantenimiento brinde información necesaria de las unidades se solicita agregar las siguientes ventanas:

- Crear bitácoras de mantenimiento, en donde se registre principalmente la fecha de ingreso y la fecha tentativa de egreso para aportar información de disponibilidad de unidades al departamento de tráfico.
- Ver las bitácoras con sus detalles de las mismas.
- Una ventana al departamento de tráfico donde se vea el estado de los operadores en cuestión de su licencia, es decir, que se encuentre vigente, definiendo como un parámetro a considerar el que este próxima a vencer su licencia, con un periodo de 15 días antes del vencimiento.

Y por último:

- Agregar una navegación óptima entre las ventanas de cada departamento.

Prototipo 11

Para comenzar con los requerimientos solicitados en la retroalimentación anterior se determina que es necesario modificar la base de datos, dado se debe de crear una bitácora de mantenimiento, además, algunas tablas extras para asociar sus relaciones foráneas, tales como: taller, tipo de mantenimiento y descripción del taller.

Una vez teniendo la base de datos actualizada, se procede a hacer el modelo lógico de la tabla de bitácora de mantenimiento, incluyendo sus funciones de mapeo. Posteriormente, se crea una ventana con los campos de texto para recabar todos los datos de la bitácora, esos datos incluye: el tipo de elemento (se puede realizar mantenimiento a un tractocamión, a una plana o un dolly), su número identificador, fecha de ingreso, fecha tentativa de egreso, descripción del mantenimiento, en caso de que sea el mantenimiento a un tractocamión se agrega el campo de kilometraje, el tipo de mantenimiento y el taller más sus detalles, dado que las

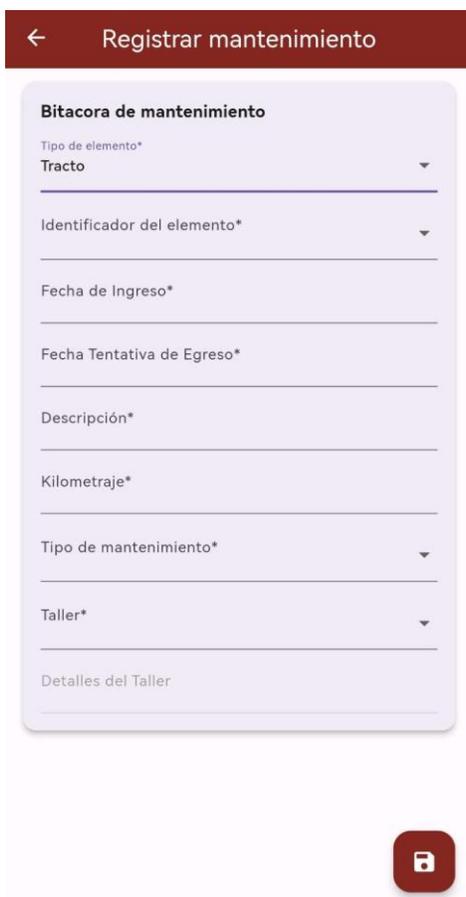
reparaciones se pueden dar en un taller interno, en un taller externo o bien en tránsito, es decir, que le reparación se hizo en un carretera a media ruta.

Una vez obtenidos todos los datos de la bitácora, se hace el *insert* a la base de datos, a la par se hace un *update* al tractocamión obteniendo el estado de “taller”, en caso de que el mantenimiento se haga a una plana o un dolly, el tractocamión asociado a dichos elementos también obtendrá el estado de “taller, dado que no puede trabajar sin tener sus elementos en correcto funcionamiento.

Concluyendo el primer requerimiento solicitado, en la Figura 3.35 se observa la ventana para registrar bitácoras de mantenimiento.

Figura 3.35

Prototipo once en la ventana para registrar bitácoras de mantenimiento



El prototipo muestra una interfaz de usuario para registrar un mantenimiento. El título de la pantalla es "Registrar mantenimiento". El formulario principal, titulado "Bitacora de mantenimiento", contiene los siguientes campos:

- Tipo de elemento* (Menú desplegable con "Tracto" seleccionado)
- Identificador del elemento* (Menú desplegable)
- Fecha de Ingreso* (Campo de texto)
- Fecha Tentativa de Egreso* (Campo de texto)
- Descripción* (Campo de texto)
- Kilometraje* (Campo de texto)
- Tipo de mantenimiento* (Menú desplegable)
- Taller* (Menú desplegable)
- Detalles del Taller (Campo de texto)

En la parte inferior derecha del formulario hay un botón rojo con un ícono de guardar.

El siguiente requerimiento a abordar es la ventana para ver las bitácoras registradas, para realizar esto se sigue un proceso similar al que se hizo con los viajes, se debe hacer una

consulta a la base de datos con todos los datos relacionados a la bitácora, con el fin de mostrar los detalles de forma completa, es decir, el uso de cláusulas *JOIN* para las claves foráneas.

En la ventana de las bitácoras se implementa la opción de terminar los mantenimientos activos existentes mediante el uso de un icono que active la función para realizar un *update*. Al terminar un mantenimiento se solicita el último atributo de la tabla de bitácora, la fecha real de egreso, al momento de terminar el mantenimiento se actualiza el estatus de la unidad a “Vacío”, lo que significa que ya puede ser asignado a algún viaje.

Por otro lado, para implementar la visualización de los detalles de la bitácora se hace mediante el uso de un dialogo, en donde se crea un objeto del modelo lógico de la tabla (modelo que ya ha sido creado), para poder mostrar todos los datos de la bitácora, para acceder a este dialogo se definió por medio de un clic a cada *card*.

La ventana de las bitácoras registradas se observa en la Figura 3.36, por otro lado, el dialogo con los detalles de la bitácora se observa en la Figura 3.37.

Figura 3.36

Prototipo once en la ventana de bitácoras registradas

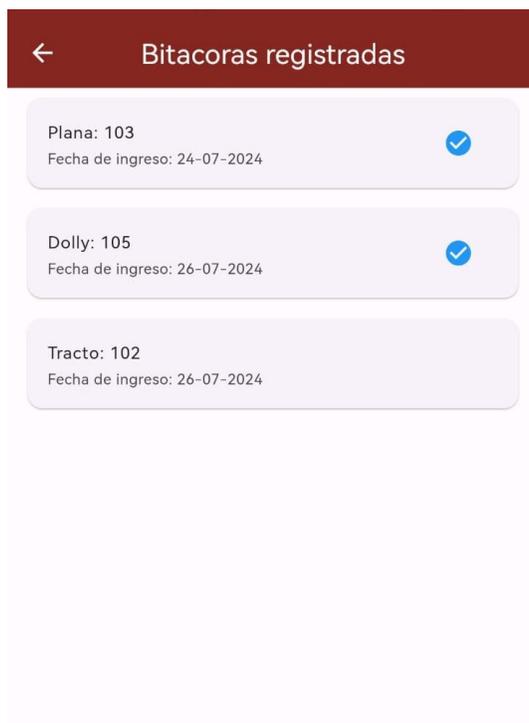
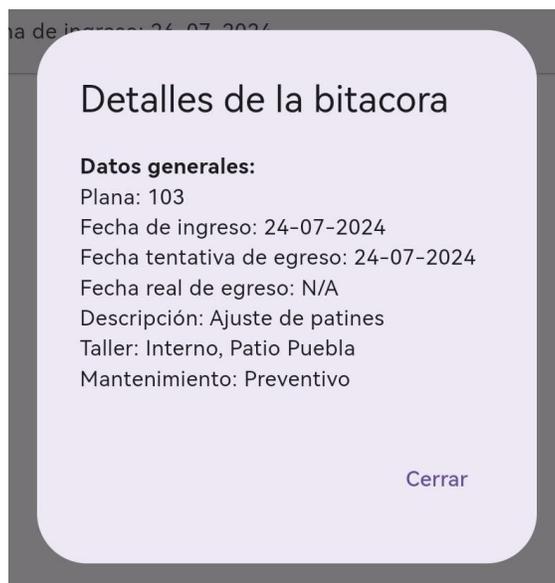


Figura 3.37

Dialogo de los detalles de la bitácora



El siguiente requerimiento a trabajar es agregar una navegación óptima entre las ventanas que tiene acceso cada departamento de trabajo, por lo que se decide implementar el widget *bottomNavigationBar* que ya ha sido implementado en la sección de las altas y bajas, por lo que, se debe de crear una clase de navegación definiendo las ventanas que tiene acceso cada departamento, siendo establecido de la siguiente forma para cada departamento:

Administración; Asignar operador y Reasignar plana.

Mantenimiento: Registrar bitácora, Mantenimientos activos y Solicitudes de refacciones.

Tráfico: Status de unidades, Viajes pendientes, Viajes activos y Status de operadores.

Ventas: Viajes terminados y Agendar viaje

El último requerimiento solicitado es agregar una ventana para ver el estado de los operadores, en donde, dicho estado se determina por el atributo del vencimiento de la licencia de conducir. Para la creación de la ventana se recicla el código que obtiene a todos los operadores con sus respectivos datos, en especial, el vencimiento de su licencia. Una vez obtenido el atributo del vencimiento se determinan parámetros siguiendo un modelo de semáforo, el verde significa que la licencia está en orden, es decir, está vigente y el operador puede trabajar sin problema. El color amarillo significa que la licencia está próxima a vencer,

definiendo un periodo de 15 días antes de su vencimiento para ser tomado en cuenta por los departamentos de trabajo y tomar decisiones en cuanto el papeleo de dicho operador. Y por último, el color rojo significa que la licencia del operador ya ha vencido, lo que representa un foco de atención de la empresa para evitar temas legales.

Los registros obtenidos de la consulta a la base de datos se muestran por medio de *Cards*, a estas se les agrega un *widget container* el cual cambiará su color dependiendo del estado de la licencia como ya ha sido mencionado.

Dicha ventana del estado de los operadores se observa en la Figura 3.38, donde han sido ingresados datos de prueba para ilustrar el funcionamiento de la ventana.

Figura 3.38

Prototipo once en la ventana de los status de los operadores



3.11.1 Retroalimentación del prototipo

Una vez implementados los requerimientos del departamento de mantenimiento y haber sido probada la aplicación móvil por los usuarios finales, se solicita agregar las siguientes ventanas:

- Asignar el estatus de “descanso”.
- Cambiar el estatus de “descanso” a “vacío”.
- Agregar datos sobre las pólizas, datos como el número de póliza, inciso, aseguradora y vigencia.
- Ver las pólizas existentes.
- Asociar pólizas a tractocamiones.

Y por último:

- Filtrar el inicio de sesión dependiendo de su departamento de trabajo autorizado, es decir, asignar departamento autorizados a cada usuario para limitar su navegación dentro de *Freightem*.
- Modificar la ventana de crear cuenta para que sea posible asignar los departamentos autorizados.

Prototipo 12

Atendiendo los nuevos requerimientos solicitados en el último prototipo, se decide comenzar por el inicio de sesión dependiendo de los departamentos autorizados. Para lograr este requerimiento se debe de modificar la base de datos para agregar un par de tablas nuevas, una que alojará los departamentos y la segunda tabla para relacionar los permisos de cada departamento con los usuarios, de forma inicial se relacionan de forma manual mediante el uso de las *ids* de los usuarios y los departamentos.

Una vez teniendo los usuarios con permisos agregados dentro de la base de datos por medio de sus claves foráneas en la tabla de los permisos, se debe de definir una función que consulte las *ids* de los usuarios con las de los departamentos, siendo 4 departamentos los existentes que son: administración, mantenimiento, tráfico y ventas, en caso de que no se asigne ningún departamento únicamente se tiene acceso a las ventanas iniciales de la aplicación móvil. La

función valida el acceso de los usuarios a los departamentos, esta validación se agrega en la ventana inicial de cada departamento, como ya se ha definido un *bottomNavigationBar* para la navegación entre las ventanas de cada departamento se ha facilitado este requerimiento. Para poder mantener el *id* con la que se hace el inicio de sesión inicial se necesita agregar la dependencia *shared_preferences*, el objetivo de mantener el *id* del inicio de sesión es realizar la validación por medio de la función mencionada. En caso de no ser valido el usuario con el que se desea acceder a la sección de un departamento no autorizado se agrega una ventana con un texto exponiendo que el usuario no tiene el acceso permitido a dicho departamento.

El siguiente requerimiento a abordar es la modificación de la ventana de crear cuenta para agregar los departamentos autorizados del usuario nuevo, esto se logra mostrando los departamentos existentes cuando se crea un nuevo usuario, dependiendo de los departamentos que se selecciones se hace un *insert*, uno a la tabla usuarios y el otro a la tabla que relaciona el nuevo usuario con los departamentos autorizados.

En la Figura 3.39 se muestra la ventana de crear usuarios actualizada agregando los departamentos autorizados.

Figura 3.39

Prototipo doce en la ventana de crear usuario actualizada

El prototipo de la ventana de crear usuario actualizada muestra un formulario con los siguientes elementos:

- Un encabezado con un botón de retroceso y el título "Crear cuenta".
- Un campo de texto etiquetado "Usuario".
- Un campo de texto etiquetado "Contraseña".
- Un campo de texto etiquetado "Confirmar Contraseña".
- Una lista de departamentos con casillas de selección:
 - Limitado
 - Administración
 - Mantenimiento
 - Tráfico
 - Ventas
- Un botón "Crear Cuenta" en la parte inferior.

Los siguientes requerimientos solicitados es agregar una ventana para asignar descansos y remover descansos, se determina implementar el *widget TapBar* para realizar estas acciones, de decide por reciclar el código de la ventana del estado de las unidades, únicamente se va a definir que al seleccionar una unidad se llame la función que asigne el estado de “descanso”, esta función hace un *update* dependiendo de la unidad que se haya seleccionado. En la Figura 3.40 se muestra la ventana para asignar descansos a las unidades que ha sido incluida en el departamento de administración.

Figura 3.40

Prototipo doce en la ventana de descansos en la vista de asignar descanso

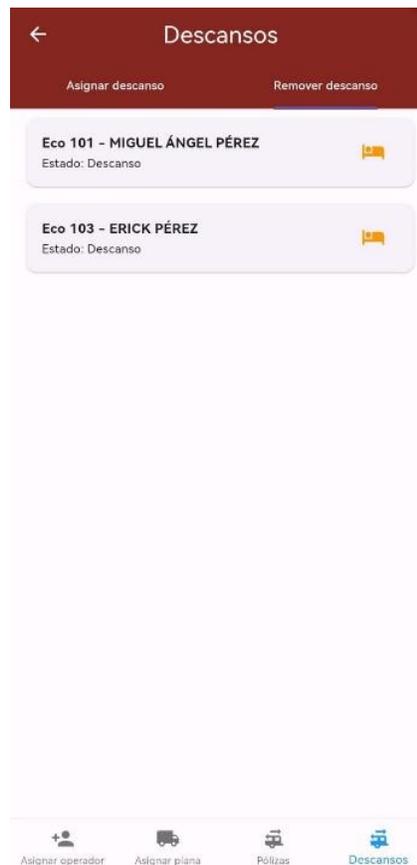


Por otra parte, para remover el estado de “descanso” de una unidad, es decir, cambiar de “descanso” a “vacío” se recicla el código de la ventana de asignar descanso, en donde se modifica la función para que en lugar de actualizar el estatus a “descanso” sea cambiado a “vacío”.

De forma lógica, se hace un *update* a la unidad seleccionada por medio de su *id* para que cambie su estado. Con esto logrando modificar los estatus de las unidades en cuestión del descanso. En la Figura 3.41 se observa la vista para remover el descanso de la unidad cambiando su estado a “vacío”. De esta forma ya se da uso a todos los estados posibles definidos para los tractocamiones.

Figura 3.41

Prototipo doce en la ventana de descansos en vista de remover descanso



El siguiente requerimiento es el agregar pólizas, para este requerimiento se debe modificar la base de datos para agregar la tabla de póliza con sus respectivos atributos y agregar la clave primaria de las pólizas a la tabla de los tractocamiones para poder asociarlos. Una vez modificada la base de datos, se debe crear un modelo lógico de la tabla pólizas, el siguiente paso es crear una ventana con los campos de texto necesarios para recabar los datos de la póliza, para la obtención de las fechas se hace uso de un calendario como ya ha sido implementado previamente. Una vez recabados todos los datos, se hace un *insert* a la base

de datos con todos los datos de la póliza. En la Figura 3.42 se observa la ventana para agregar las pólizas.

Figura 3.42

Prototipo doce en la ventana para agregar pólizas

El prototipo muestra una ventana con un encabezado rojo que contiene un ícono de retroceso y el título 'Registrar póliza'. El cuerpo de la ventana es un formulario con un fondo gris claro y el título 'Datos de la póliza'. El formulario incluye los siguientes campos de entrada:

- Número de póliza*
- Número de inciso*
- Aseguradora*
- Vigencia desde*
- Hora
- Vigencia hasta*
- Hora

En la parte inferior derecha de la ventana hay un botón redondo rojo con un ícono de guardar.

El siguiente requerimiento es agregar una ventana para observar las pólizas existentes, para esto, se debe de hacer una consulta a la base de datos para obtener todas las pólizas existentes junto con sus datos. Estos registros obtenidos se muestran por medio de *Cards* en la pantalla.

Se define la navegación a la ventana de previamente creada para registrar póliza a través de la ventana donde se muestran todas las pólizas por medio de un botón representativo de “agregar”. En la Figura 3.43 se observa la ventana que muestra todas las pólizas existentes.

Para mostrar los detalles de la póliza se usa un dialogo que muestra todos los datos de la póliza, así mismo, las unidades que tiene relacionada, en caso de no tener, se agrega la leyenda “No hay unidades asociadas a esta póliza”. En la Figura 3.44 se muestra dicho dialogo.

Figura 3.43

Prototipo doce en la ventana de pólizas



El siguiente requerimiento es asociar los tractocamiones a las pólizas, para este requerimiento se hace uso del dialogo que muestra los detalles de la póliza, en el dialogo se agregó un vinculo para asociar las unidades. Esta ventana emergente consulta a la base de datos los tractocamiones que en su atributo de relacion con la póliza sea igual a *NULL*, con esto se obtienen los tractocamiones que no tengan asociada una póliza. Seleccionando los tractocamiones que se asocian a la póliza se debe de hacer un *update* a la tabla del tractocami3n, agregando la clave foranea de la póliza que se haya agregado. En la Figura 3.45 se observa dicha ventana.

Figura 3.44

Dialogo de los detalles de la póliza

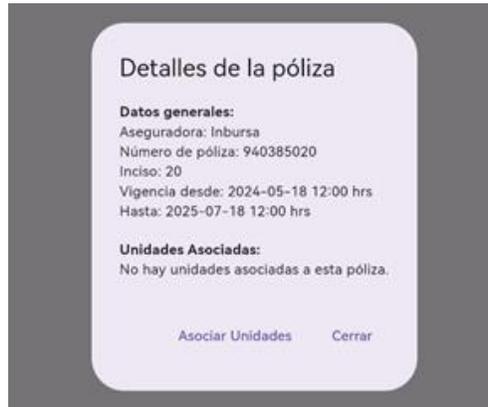
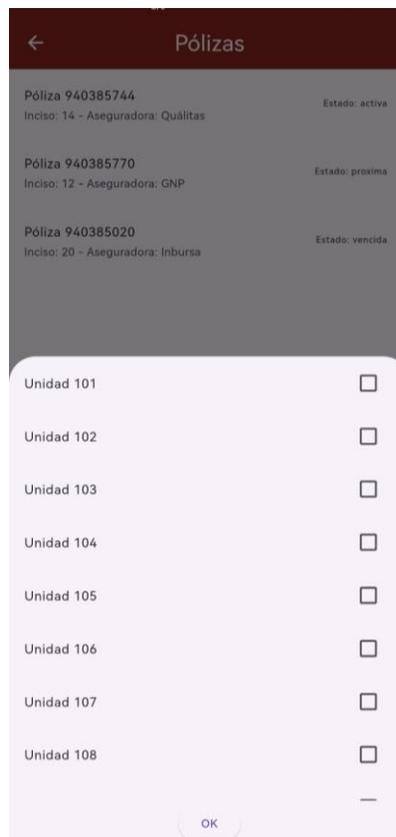


Figura 3.45

Ventana emergente para asociar tractocamiones a póliza



Estas funciones han sido agregadas a la sección de administración, por lo que el *bottomNavigationBar* ha sido actualizando considerando la nueva ventana. De esta forma, se logra cumplir con todos los requerimientos solicitados en la última retroalimentación.

3.12.1 Retroalimentación del prototipo

Una vez entregado el prototipo 12 a los usuarios finales y tras días de uso sin notificar algún inconveniente con el uso de la aplicación móvil *Freightem*, se concluyó que es el prototipo final, habiendo resuelto todos los requerimientos solicitados por los usuarios finales.

La aplicación móvil *Freightem* brinda las siguientes funciones:

- Inicio de sesión.
- Trabajo segmentado por departamentos.
- Mostrar información detallada de elementos tales como operador, tractocamión, planas y dollys.
- Búsqueda de tractocamiones.
- Llamar a operadores.
- Registrar cambios de aceite de motor.
- Exportar reportes de cambios de aceite en formato pdf.
- Solicitar refacciones para cada tractocamión.
- Asignar planas a los tractocamiones.
- Asignar operadores a los tractocamiones.
- Asignar dolly a los tractocamiones.
- Dar altas de operadores, tractocamiones, planas y dollys.
- Dar bajas de operadores, tractocamiones, planas y dollys.
- Exportar la base de datos.
- Importar una base de datos.
- Crear bitácoras de mantenimiento.
- Agendar viajes.
- Dar seguimiento a los viajes.
- Ver el estado de disponibilidad de los tractocamiones.
- Ver el estado de los operadores en cuestión de la vigencia de su licencia de conducir.
- Registrar pólizas.
- Asociar tractocamiones a su póliza correspondiente.
- Asignar un descanso a tractocamiones.

- Exportar reportes en formato pdf de los viajes realizados por unidad con una calificación del operador durante el viaje.

El prototipo final se continúa trabajando en la siguiente fase de la metodología continuando con su diseño final en la Fase 3 del Capítulo 2 del presente documento.

Capítulo 4. Validación

En el presente capítulo se presentará la cuarta fase de la metodología con la que se desarrolló la aplicación móvil *Freightem*. En esta fase de desarrollo de pruebas, se llevan a cabo evaluaciones para verificar los cambios de diseño realizados funciona de la forma que ha sido solicitada por los usuarios, asegurando que se han cumplido todos los requerimientos definidos y que el software funcione correctamente. Es crucial comprobar que los ajustes hechos durante la fase anterior, especialmente en la evolución del prototipado, no hayan afectado negativamente las funcionalidades establecidas en iteraciones anteriores.

Esta fase se centra en entregar la aplicación a los usuarios finales para que la utilicen en su trabajo diario, lo que permite verificar el cumplimiento de los requisitos en situaciones reales. Como se ha mencionado, una de las ventajas de la metodología de prototipos evolutivos es que los usuarios finales realizan pruebas constantes en cada prototipo presentado, lo que facilita significativamente esta etapa de validación.

Basándose en el principio de Crisci (2019), una característica de la calidad de software es la usabilidad. Se realizará una encuesta a los usuarios finales, tanto los usuarios que ya han usado la aplicación como a usuarios nuevos para tener la retroalimentación de estos dos tipos de usuarios y definir una conclusión enriquecida con la encuesta, dichos usuarios serán parte de la empresa privada que fue elegida para apoyar en el desarrollo del presente proyecto.

La encuesta evaluará los siguientes aspectos:

- Facilidad de aprendizaje.
- Facilidad de memoria.
- Eficiencia de la aplicación.

Las preguntas presentadas por Crisci (2019) han sido adaptadas al contexto del presente proyecto las cuales se mencionan a continuación:

Pregunta 1. ¿Me resultó fácil entender cómo realizar las tareas básicas en la aplicación en poco tiempo?

Pregunta 2. ¿Considero que la interfaz de la aplicación es intuitiva y facilita su uso en general?

- Pregunta 3. ¿Cuándo volví a usar la aplicación me acordé cómo hacerlo?
- Pregunta 4. ¿Supe qué hacer en cada momento? por ejemplo, al querer llamar al operador de una unidad x
- Pregunta 5. En el último mes, ¿cuántas veces he usado la aplicación?
- Pregunta 6. ¿La aplicación dejó de funcionar impidiendo su uso?

La población a la cual se realizó la encuesta fueron ocho personas de las cuales se incluye quienes hacían las retroalimentaciones de cada prototipo, como personas que no habían usado la aplicación, una vez realizadas las encuestas de usabilidad se obtienen los siguientes resultados:

Basándose en los resultados de la encuesta para la Pregunta 1 (“¿Me resultó fácil entender cómo realizar las tareas básicas en la aplicación en poco tiempo?”), se puede interpretar lo siguiente:

El 87.5% de los encuestados respondieron "Sí", lo que indica que la gran mayoría de los encuestados sintieron que aprendieron a usar la aplicación rápidamente. Esta es una señal positiva de usabilidad, sugiriendo que la aplicación es intuitiva y fácil de aprender para la mayoría de los usuarios.

El 12.5% de los encuestados respondieron "Tarde un poco". Aunque es una minoría, esta respuesta sugiere que algunos usuarios podrían haber experimentado cierto nivel de dificultad inicial al aprender a usar la aplicación. Sería importante investigar más a fondo las razones detrás de esta percepción y buscar formas de mejorar la experiencia de usuario para este grupo.

Ningún encuestado respondió "No", lo que indica que ningún usuario encontró que aprender a usar la aplicación fue difícil o imposible. Esto es una buena señal en términos de usabilidad, ya que ningún usuario tuvo una experiencia negativa significativa en este aspecto.

En resumen, los resultados sugieren que la mayoría de los usuarios encontraron que aprender a usar la aplicación fue rápido, lo que indica un buen nivel de usabilidad. Sin embargo, sería importante prestar atención a la minoría que indicó que les tomó un poco de tiempo aprender, para abordar cualquier posible área de mejora. Los resultados se presentan en la Gráfica 4.1.

Gráfica 4.1

Respuestas de la pregunta uno



Basándose en los resultados de la encuesta para la Pregunta 2 ("¿Considero que la interfaz de la aplicación es intuitiva y facilita su uso en general?"), se puede interpretar lo siguiente:

El 87.5% de los encuestados respondieron "Sí", lo que indica que la gran mayoría de los encuestados consideran que la aplicación es fácil de usar. Esta es una señal muy positiva en términos de usabilidad, sugiriendo que la aplicación es intuitiva y accesible para la mayoría de los usuarios.

El 12.5% de los encuestados respondieron "Poco". Aunque es una minoría, esta respuesta sugiere que algunos usuarios podrían tener ciertas reservas o encontrar ciertos aspectos de la aplicación un poco más complicados de lo esperado. Sería importante investigar más a fondo las áreas específicas que los usuarios encuentran más difíciles y buscar formas de abordar esas preocupaciones para mejorar la experiencia de usuario.

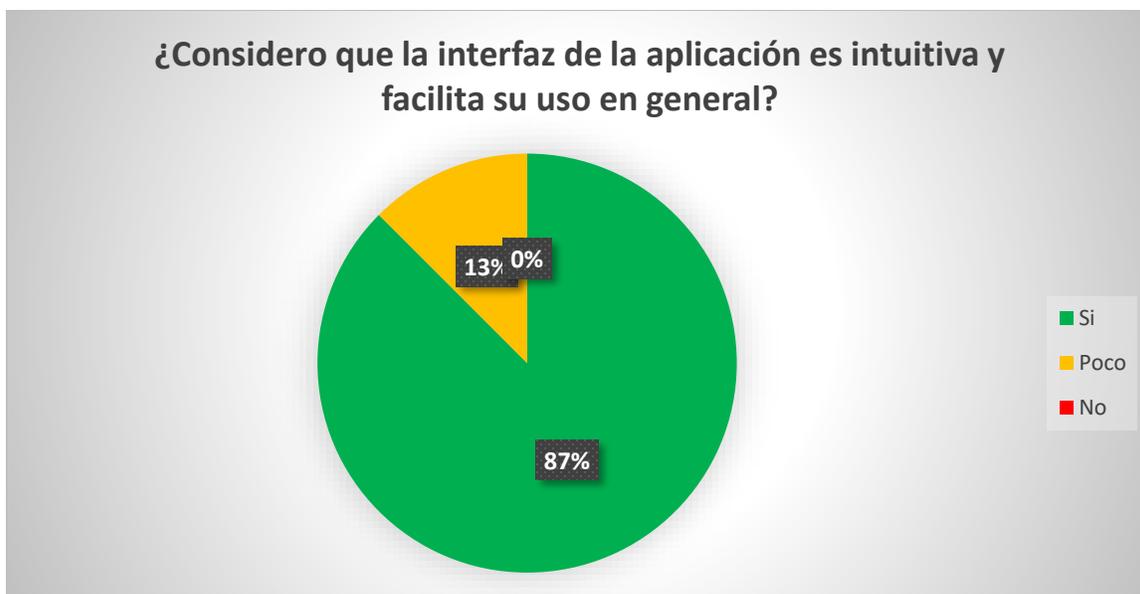
Ningún encuestado respondió "No", lo que indica que ningún usuario considera que la aplicación es difícil de usar en su totalidad. Esto es una señal positiva, ya que ningún usuario experimentó una experiencia negativa significativa en términos de usabilidad.

En resumen, los resultados sugieren que la mayoría de los usuarios perciben la aplicación como fácil de usar, lo que es una buena señal en términos de usabilidad. Sin embargo, sería

importante investigar más a fondo las preocupaciones de la minoría que considera la aplicación un poco difícil de usar, para abordar cualquier área de mejora y garantizar una experiencia de usuario óptima para todos los usuarios. Los resultados de esta pregunta se presentan en la Gráfica 4.2.

Gráfica 4.2

Respuestas de la pregunta dos



Basándose en los resultados de la encuesta para la Pregunta 3 ("Cuando volví a usar la aplicación, ¿me acordé cómo hacerlo?"), se puede interpretar lo siguiente:

El 62.5% de los encuestados respondieron "Sí", lo que indica que la mayoría de los encuestados se sintieron capaces de recordar cómo usar la aplicación cuando volvieron a utilizarla. Esto sugiere que la aplicación tiene una buena capacidad para mantener la familiaridad y la memoria de los usuarios, lo que contribuye a una experiencia de usuario positiva.

El 37.5% de los encuestados respondieron "A veces". Esta respuesta sugiere que algunos usuarios pueden haber experimentado dificultades ocasionales para recordar cómo usar la aplicación al volver a utilizarla. Sería importante investigar más a fondo las razones detrás de estas dificultades ocasionales y buscar formas de mejorar la consistencia y la facilidad de uso de la aplicación para todos los usuarios.

Ningún encuestado respondió "No", lo que indica que ningún usuario tuvo dificultades significativas para recordar cómo usar la aplicación al volver a utilizarla. Esto es una señal positiva en términos de usabilidad, ya que los usuarios no encontraron que la aplicación fuera difícil de recordar o usar después de un período de no uso.

En resumen, los resultados sugieren que la mayoría de los usuarios pudieron recordar cómo usar la aplicación al volver a utilizarla, lo que indica una buena capacidad de la aplicación para mantener la familiaridad y la memoria de los usuarios. Sin embargo, sería importante abordar las preocupaciones de la minoría que experimentó dificultades ocasionales para recordar cómo usar la aplicación, para garantizar una experiencia de usuario coherente y satisfactoria para todos los usuarios. Los resultados de dicha pregunta se presentan en la Gráfica 4.3.

Gráfica 4.3

Respuestas de la pregunta tres



Basándose en los resultados de la encuesta para la Pregunta 4 ("¿Supe qué hacer en todo momento?"), se puede interpretar lo siguiente:

El 75% de los encuestados respondieron "Sí", lo que indica que la mayoría de los encuestados se sintieron seguros y confiados en cuanto a qué hacer en todo momento mientras utilizaban la aplicación. Esto sugiere que la aplicación proporciona una guía clara y una experiencia de usuario fluida para la mayoría de los usuarios.

El 25% de los encuestados respondieron "A veces". Esta respuesta sugiere que algunos usuarios pueden haber experimentado momentos de incertidumbre o falta de claridad sobre qué hacer mientras utilizaban la aplicación. Sería importante investigar más a fondo las áreas específicas de la aplicación que causaron esta sensación de confusión ocasional y buscar formas de mejorar la claridad y la facilidad de uso para todos los usuarios.

Ningún encuestado respondió "No", lo que indica que ningún usuario se sintió completamente perdido o incapaz de saber qué hacer en algún momento mientras utilizaba la aplicación. Esto es una señal positiva en términos de usabilidad, ya que todos los usuarios pudieron encontrar su camino a través de la aplicación en algún grado.

En resumen, los resultados sugieren que la mayoría de los usuarios se sintieron seguros en cuanto a qué hacer en todo momento mientras utilizaban la aplicación, lo que indica una buena claridad y orientación proporcionada por la aplicación. Sin embargo, sería importante abordar las preocupaciones de la minoría que experimentó momentos de incertidumbre, para garantizar una experiencia de usuario fluida y confiable para todos los usuarios. Los resultados de dicha pregunta se presentan en la Gráfica 4.4.

Gráfica 4.4

Respuestas de la pregunta cuatro



Basándose en los resultados de la encuesta para la Pregunta 5 ("En el último mes, ¿cuántas veces he usado la aplicación?"), se puede interpretar lo siguiente:

El 50% de los encuestados respondieron que usaron la aplicación de seis a 10 veces en el último mes. Esto indica que la mitad de los encuestados son usuarios frecuentes de la aplicación, lo que sugiere un nivel significativo de compromiso con la misma.

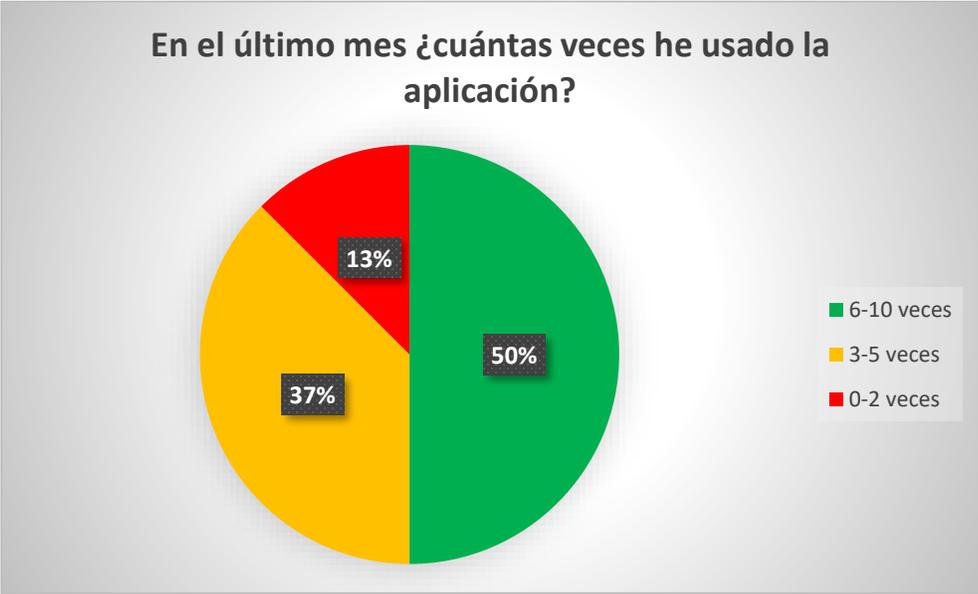
El 37.5% de los encuestados respondieron que usaron la aplicación de tres a cinco veces en el último mes. Esto indica que una parte considerable de los encuestados usa la aplicación moderadamente, lo que sugiere un nivel medio de compromiso o uso ocasional.

El 12.5% de los encuestados respondieron que usaron la aplicación de cero a dos veces en el último mes. Esto indica que una minoría de los encuestados usan la aplicación con poca frecuencia o casi nunca en comparación con el resto de los encuestados.

En resumen, los resultados sugieren que hay una variedad en la frecuencia de uso de la aplicación entre los encuestados, con algunos siendo usuarios frecuentes, otros siendo usuarios moderados, y una minoría usando la aplicación con poca frecuencia. Estos datos pueden ser útiles para comprender mejor el comportamiento de los usuarios y orientar futuras estrategias de desarrollo. Los resultados de dicha pregunta se presentan en la Gráfica 4.5.

Gráfica 4.5

Respuestas de la pregunta cinco



Basándose en los resultados de la encuesta para la Pregunta 6 ("¿La aplicación dejó de funcionar, impidiendo su uso?"), se puede interpretar lo siguiente:

El 100% de los encuestados respondieron "No". Esto indica que ninguno de los encuestados experimentó una interrupción completa en el funcionamiento de la aplicación que impidiera su uso durante el período especificado.

Basándose en los resultados de la encuesta para la Pregunta 6 ("¿La aplicación dejó de funcionar, impidiendo su uso?"), se puede interpretar lo siguiente:

El 100% de los encuestados respondieron "No". Esto indica que ninguno de los encuestados experimentó una interrupción completa en el funcionamiento de la aplicación que impidiera su uso durante el período especificado.

Basándose en los resultados de la encuesta para la Pregunta 6 ("¿La aplicación dejó de funcionar, impidiendo su uso?"), se puede interpretar lo siguiente:

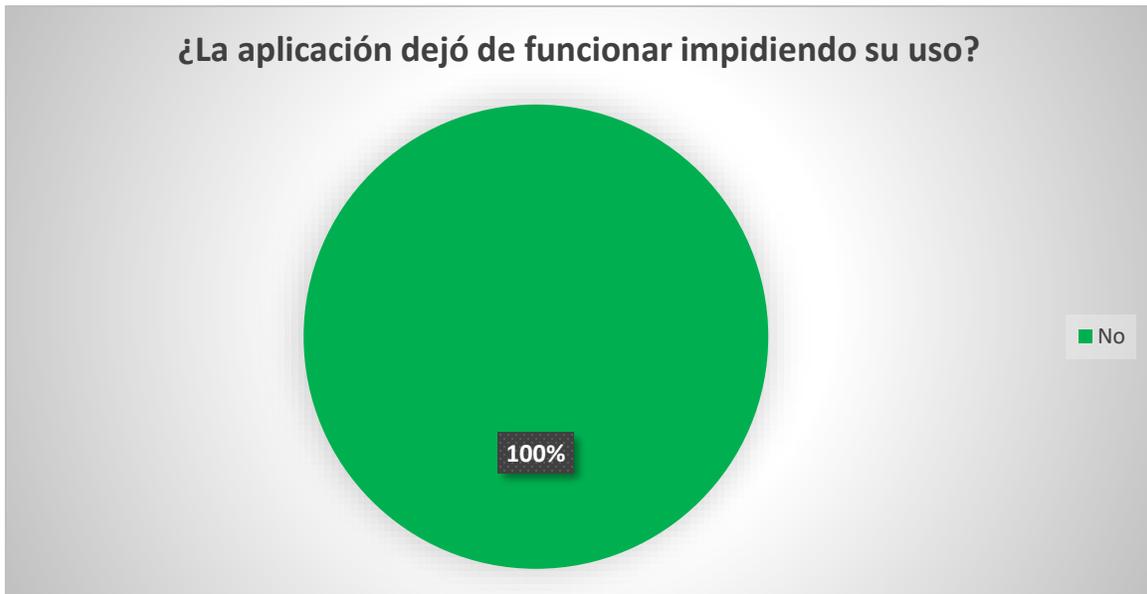
El 100% de los encuestados respondieron "No". Esto indica que ninguno de los encuestados experimentó una interrupción completa en el funcionamiento de la aplicación que impidiera su uso durante el período especificado.

En resumen, los resultados sugieren que la aplicación ha demostrado ser estable y confiable en términos de funcionamiento, ya que ningún usuario experimentó una interrupción total que impidiera su uso durante el período de tiempo especificado. Esto es una buena noticia para los desarrolladores y usuarios, ya que indica que la aplicación está bien mantenida y puede ser utilizada sin preocupaciones significativas sobre fallos graves. Los resultados de dicha pregunta se presentan en la Gráfica 4.6

.Una vez presentados todos los resultados de las encuestas se puede concluir que las pruebas de usabilidad arrojaron resultados positivos, mostrando que la curva de aprendizaje de la aplicación fue corta, con lo cual, se asegura el haber creado una interfaz de usuario intuitiva., sin embargo, presentan algunos focos rojos que significan que hay capacidad de mejora en cuanto a la usabilidad de la aplicación móvil *Freightem*. Por lo tanto, se puede continuar a la etapa final de la metodología.

Gráfica 4.6

Respuestas de la pregunta seis



Conclusiones

Durante el transcurso de esta tesis, se desarrolló una aplicación para una línea de autotransporte, logrando alcanzar todos los objetivos planteados inicialmente. Sin embargo, también se enfrentaron diversos desafíos que contribuyeron a un enfoque más especializado de acuerdo a las necesidades específicas de la empresa que apoyó al desarrollo. Al concluir esta tesis, es importante destacar las siguientes conclusiones:

Se logró la implementación de los requisitos funcionales planteados tanto al inicio como durante el desarrollo de la aplicación móvil. La aplicación ahora se ha convertido en una herramienta de uso remoto, lo que resulta invaluable en situaciones que demanda la logística de transporte, como la comunicación con operadores, el rescate de unidades y la identificación de las mismas.

La colaboración directa entre los usuarios finales y el desarrollador resultó ser un factor crucial para la realización de la aplicación. La participación constante contribuyó de forma significativa al diseño de la aplicación, que satisface en gran las necesidades del departamento de administración, mantenimiento, tráfico y ventas.

La seguridad de los datos es crucial en todo desarrollo de software, por lo que, el uso de una base de datos local resultó ser acertado. Además, el acceso al sistema mediante un usuario personalizado con contraseña aumenta la confidencialidad de los datos, cumpliendo con estándares de seguridad.

Llevar a cabo pruebas de forma constante garantiza un fácil aprendizaje por parte de los usuarios finales, además de asegurar el funcionamiento de la aplicación en diferentes versiones de sistemas operativos.

La documentación del código y la aplicación se ha elaborado de forma exhaustiva. Esto habilita la posibilidad de realizar un mantenimiento futuro de manera efectiva, además de permitir la incorporación de nuevas actualizaciones o funcionalidades.

En resumen, este proyecto de desarrollo de una aplicación móvil no solo ha resuelto la problemática identificada en las empresas medianas de transporte, sino que también ha sentado las bases para una gestión más eficiente y moderna. Las conclusiones obtenidas

resaltan el constante compromiso de entregar un producto funcional, así como el compromiso con la excelencia y la firme disposición ante el cambio y la evolución de las demandas en el entorno de la logística.

Esta aplicación no solo representa una solución tecnológica, sino también un avance en la forma de operar en las líneas de transporte hacia un enfoque más moderno y tecnológico para las empresas medianas de transporte.

Trabajos futuros

Una vez concluido el desarrollo de la aplicación móvil *Freightem* y observando las posibilidades, se determina que crear un entorno de trabajo completo sería una excelente mejora, en donde, al contar con un software para ordenador que sea capaz de interactuar con la aplicación móvil agilizaría y optimizaría los procesos. Siendo posible mediante la implementación de un servidor para garantizar las conexiones remotas a la base de datos.

Al mismo tiempo, se observa que la aplicación tiene el potencial de agregar más funciones que quizás no han sido pensadas por los usuarios finales durante el desarrollo de esta tesis.

Por último, en la aplicación *Freightem* solo han sido considerados los remolques de tipo plataforma estándar, es posible definir un catálogo de remolques que va más allá de las plataformas estándar para ofrecer una aplicación con un mercado más extenso.

Referencias

- Abelló, A., Rodríguez, M., & Rollón, E. (2021). *Diseño y administración de bases de datos*. <https://www.mdx.cat/handle/2099.3/36799>
- Acosta Espinoza, J. L., Lenin León Yacelga, A. R., & Sanafria Michilena, W. G. (2022). Las aplicaciones móviles y su impacto en la sociedad. *Revista Universidad y Sociedad*, 14(2), 237-243. http://scielo.sld.cu/scielo.php?pid=S2218-36202022000200237&script=sci_arttext&tlng=en
- Adobe. (2024). *Adobe*. <https://www.adobe.com>
- Ballou, R. H. (2003). *Business Logistics Management* (5th ed.). Prentice Hall.
- Barrera, J. A. (2020). *Metodologías para el Desarrollo de Proyectos* (Tesis de licenciatura). Unicatolica, Facultad de Administración, Contabilidad y Finanzas, Programa de Administración de Empresas.
- Bhardwaj, K., Saunders, M., Juneja, N., & Gavrilovska, A. (2019). Serving Mobile Apps. *Proceedings of the Fourteenth EuroSys Conference 2019*, (pp. 1–15). <https://doi.org/10.1145/3302424.3303989>
- Cardona Ramirez, M. (2024). *Plan de empresa para la creación de Catch, software para logística y transporte* (Tesis de licenciatura). Santiago de Cali. <https://red.uao.edu.co/server/api/core/bitstreams/d06e100b-9402-4b1d-b37a-0b56f4e6504e/content>
- Collaguazo, M. L., Venegas, M. M., Guerrero, A. A., & Beltrán, M. S. (2022). Desarrollo híbrido con flutter. *Ciencia Latina Revista Científica Multidisciplinar*, 6(4), 4594-4609. <https://www.ciencialatina.org/index.php/cienciala/article/view/2959>
- Crisci, J. F. (2019). *Evaluación de satisfacción y usabilidad de una aplicación móvil* (Tesis de licenciatura). Buenos Aires.
- Dianes, M. (2019). Fleet Complete Blog. <https://blog.fleetcomplete.com/fleet-complete-releases-new-advanced-fleet-management-and-mobile-workforce-management-functionality/>

- Fleet Complete. (2024). *Fleet Complete*. <https://www.fleetcomplete.com/es/>
- Flutter. (2024). *Flutter*. <https://www.flutter.dev>
- Forcada, J. (2020). *Frameworks para desarrollo de aplicaciones móviles híbridas: análisis comparativo y aplicación a servicios de emergencia* (Tesis Doctoral). Universidad Politécnica de Madrid. <https://oa.upm.es/64414/>
- Gamma Corporation . (2024). *Corpus Trailers*. <https://corpustrailers.com>
- Geotab. (2024). *Geotab*. <https://www.geotab.com/es-latam/>
- Jiménez Sánchez, J. E., & Jiménez Castillo, J. (2016). Logística del autotransporte de carga: estrategias de gestión. <https://trid.trb.org/View/1470282>
- Lambert, D. M., Cooper, M. C., & Pagh, J. D. (1998). Supply chain management: implementation issues and research opportunities. *The international journal of logistics management*, 9(2), 1-20. <https://www.emerald.com/insight/content/doi/10.1108/09574099810805807/full/html>
- Letelier, P. (n.d.). Proceso de Desarrollo de Software. *Universidad Politécnica de Valencia*.
- López Tulcanaza, V. A. (2010). *Fragmentación de tablas en bases de datos* (Tesis de licenciatura). Ecuador. <https://repositorio.utn.edu.ec/handle/123456789/522>
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software* (Tesis de Licenciatura en Sistemas y Computación). Universidad Católica Argentina. <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Marqués, M. (2011). *Bases de datos*. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions.
- Martínez, C. J., Moreno, H., & Hernández, M. B. (2020). The evolution from Traditional to Intelligent Web Security: Systematic Literature Review. (pp. 1-9). IEEE. <https://ieeexplore.ieee.org/document/9297240>

- McIlroy, S., Ali, N., & Hassan, A. E. (2016). Fresh apps: An empirical study of frequently-updated mobile apps in the Google play store. *Empirical Software Engineering*, 21(3), 1346-1370. <https://link.springer.com/article/10.1007/s10664-015-9388-2>
- Mejía García, D., & Acosta Pérez, B. (2019). Avances tecnológicos modernos y sus implicaciones en el pensamiento social. *AULA Revista de Humanidades y Ciencias Sociales*, 65(5), 29-37. <http://dx.doi.org/2636.2236/AULA.2019.012>
- Menéndez, J. G. (2022). *Utilización de las bases de datos relacionales en el sistema de gestión y almacenamiento de datos*. Ediciones Paraninfo. https://books.google.com.mx/books?hl=es&lr=&id=hPV2EAAAQBAJ&oi=fnd&pg=PR7&dq=bases+de+datos+relacional&ots=ZuQ148DgA9&sig=zsoZXA68L-TmJ7yyJXO_wQgq9ZM&redir_esc=y#v=onepage&q=bases%20de%20datos%20relacional&f=false
- Microsoft. (2024). *Visual Studio Code*. <https://code.visualstudio.com>
- Nahuel Delía, L. (2017). *Desarrollo de aplicaciones móviles multiplataforma* (Trabajo final presentado para obtener el grado de especialista en ingeniería de software). <https://sedici.unlp.edu.ar/handle/10915/60497>
- Osuna, K. (2020). *Implementación de Flutter para el desarrollo de aplicaciones móviles nativas en iOS y Android* (Tesis de licenciatura). Mazatlán, Sinaloa, México. <http://repositorio.upsin.edu.mx/Fragmentos/tesinas/A031LIZARRAGAOSUNAKEVINANTONIO8101.pdf>
- PACCAR Inc. (2024). *Kenworth*. <https://kenworth.com.mx>
- Palumbo, N. &. (2023). Un Un Entorno de Desarrollo Integrado para Dispositivos Móviles. *Memorias de las JAIIO*. <https://ojs.sadio.org.ar/index.php/JAIIO/article/view/655>
- Patino, A. (2020). *Compara Software*. <https://blog.comparasoftware.com/metodologia-waterfall/>
- Pazmiño, A. M., Poma, M. E., Cárdenas, C. V., & Ordoñez, L. H. (2020). *Administración de empresas. Elementos básicos*. Infinite Study. <https://books.google.com.mx/books?hl=es&lr=&id=zLgoEAAAQBAJ&oi=fnd&pg>

=PR11&dq=ADMINISTRACIÓN+DE+EMPRESASELEMENTOS+BÁSICOS+AI
exandra+Maribel+Arguello+Pazmiño+2020&ots=81qqCuw8Ud&sig=BBq4Jj2yH
MKb_syvXQJeh4-
8s1E&redir_esc=y#v=onepage&q=ADMINISTRACIÓN%20DE%20

- Pressman, R. S. (2010). *Ingeniería del Software Un enfoque práctico*. Mc graw hill.
- Ríos, J. R., Tapia, J. A., Souto, N. P., & León, H. P. (2021). Comparativa de metodologías de desarrollo de aplicaciones móviles. *3c Tecnología: glosas de innovación aplicadas a la pyme*, 10(2), 73-93. <https://dialnet.unirioja.es/servlet/articulo?codigo=8044474>
- Roig, M. V., & Castillo, C. (2022). Evolución de la logística: pasado, presente y futuro. *Oikonomics: Revista de economía*, 17, 1-8. <https://oikonomics.uoc.edu/divulgacio/oikonomics/es/numero17/dossier/mviu.html>
- SQLite. (2024). *SQLite*. <https://www.sqlite.org>
- SQLite. (n.d.). *DB Browser for SQLite*. <https://sqlitebrowser.org/>
- Wisdom Factory. (2024). *Sygie*. <https://www.sygie.com>

Anexo A. Manual de usuario

En este anexo, se presenta el manual de usuario de la aplicación móvil *Freightem*. Este anexo es fundamental para que el cliente comprenda cómo interactuar de manera efectiva con la aplicación, aprovechando al máximo sus funcionalidades. *Freightem* ha sido diseñada pensando en el usuario, para ofrecer una experiencia intuitiva y eficiente desde el primer momento.

Para empezar, se guiará través del proceso de inicio de sesión, el cual es la puerta de entrada a todas las funcionalidades y beneficios que tiene la aplicación para ofrecer. Al iniciar la aplicación, se encuentra con una pantalla de inicio de sesión donde se deben validar las credenciales del usuario, en caso de no tener credenciales autorizadas se deberá crear un usuario que se explica más adelante en este anexo. Esta pantalla consta de dos campos que deben completarse: uno para el nombre de usuario y otro para la contraseña. La pantalla mencionada se muestra en la Figura A.1. Esta es la pantalla inicial de nuestra aplicación, el primer paso hacia una gestión más sencilla y organizada, en dicha pantalla se observan los siguientes elementos:

1. Campo de texto para ingresar el nombre del usuario.
2. Campo de texto para ingresar la contraseña.
3. Botón para iniciar sesión.
4. Opción para crear un nuevo usuario.

Si se accede a la creación de un nuevo usuario se despliega la pantalla desde la que puede agregar un nuevo usuario con sus credenciales correspondientes, además, se puede asignar los departamentos que tendrá autorizado para acceder, existen 5 opciones para seleccionar, estas opciones son las siguientes: “Limitado”, “Administración”, “Mantenimiento”, “Tráfico” y “Ventas”. La opción de “Limitado” no es un departamento, pero funciona para tener acceso limitado a la aplicación, siendo un acceso únicamente a la pantalla inicial de la aplicación. En la pantalla mostrada en la Figura A.2 se observan los elementos:

1. Campo de texto para agregar el usuario nuevo.
2. Campo de texto para agregar la contraseña.
3. Campo de texto para verificar la contraseña.

4. Lista de opciones de acceso a la aplicación móvil.

Figura A.1

Inicio de sesión



Una vez se ha iniciado sesión con credenciales validas, se muestra la pantalla principal de la aplicación móvil, esta ventana se observa en la Figura A.3. En esta ventana es posible hacer la búsqueda de las unidades por medio de su número económico y navegar a todas las funciones que ofrece la aplicación móvil. En esta ventana se observan los elementos:

1. Menú desplegable que contiene las opciones de “Agregar”, “Eliminar”, “Administración”, “Mantenimiento”, “Tráfico”, “Ventas”, “Información” y un icono para los “Ajustes”.
2. Campo en el que se puede ingresar el número de la unidad que de desea buscar, en donde el rango de números depende de los tractocamiones existentes.
3. Una lista de todos los tractocamiones existentes junto con su operador, en donde, al dar clic se despliega la ventana de los detalles del tracto.

4. Un icono flotante que solo aparecerá en caso de tener vencimientos próximos de licencias de operadores o de pólizas.
5. Icono para cerrar sesión.
6. Botón para recargar la ventana principal.

Figura A.2

Pantalla para crear un nuevo usuario

1

2

3

4

Al dar clic en una unidad se presenta la ventana de los elementos asociados al tractocamión seleccionado, en donde se pueden ver de forma general las planas y dolly (en caso de tener) asociados a la unidad esta ventana se observa en la Figura A.4 que contiene los elementos:

1. Identificador del tracto que se está visualizando.

- Identificador de las planas que tiene relacionada(s) el tractocamión, en caso de ser un tracto sencillo no existe relación de una segunda plana ni de un dolly, por lo que, se muestra una leyenda indicando la ausencia de elementos.
- El nombre del operador que tiene asociado el tractocamión.
- Botón para llamar al operador del tractocamión.
- Botones para visualizar las características específicas de cada elemento.
- Menú desplegable que contiene la opción de registrar “Cambio de aceite” y “Registrar refacciones” para la unidad que se está visualizando.

Figura A.3

Pantalla principal de la aplicación Freightem

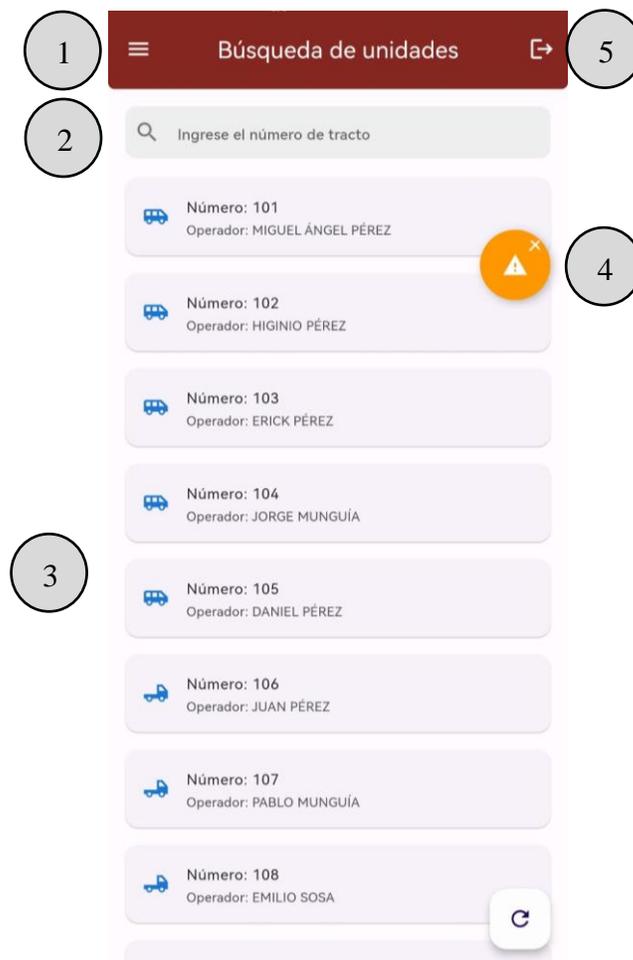
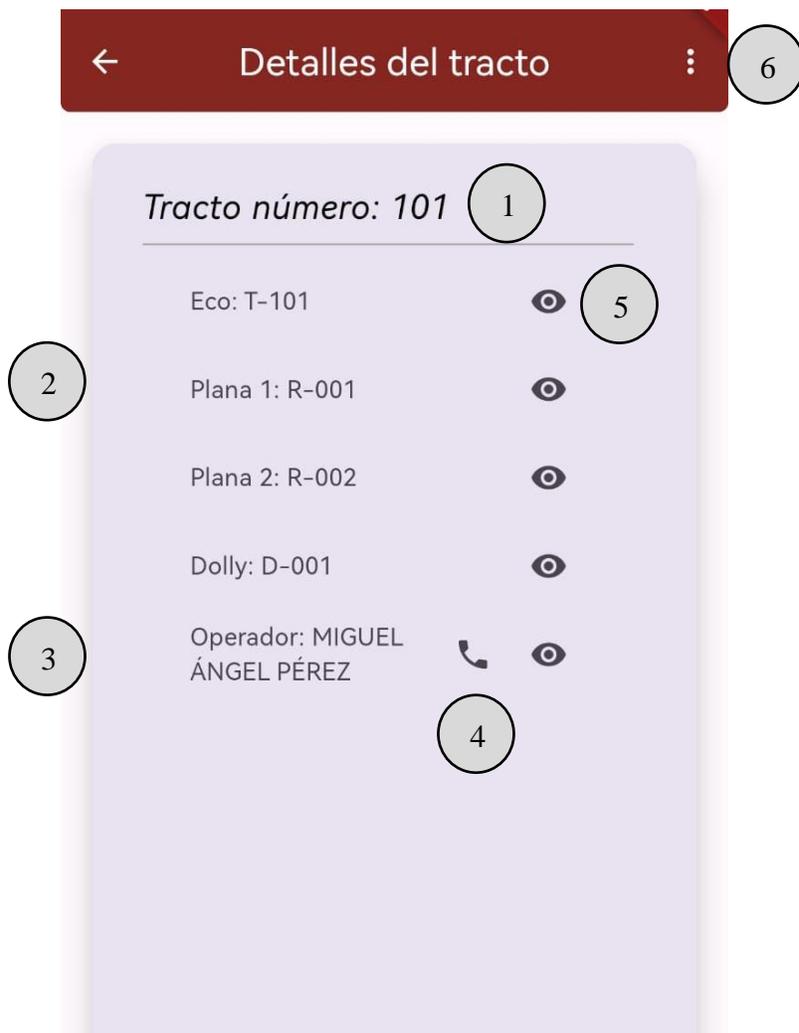


Figura A.4

Pantalla de elementos asociados al tractocamión



Al ingresar a las características específicas de algún elemento, en el caso que se presenta en la Figura A.5 son de las características del operador, se observan las características específicas, esta ventana contiene los elementos:

1. Descripción del elemento que se está visualizando.
2. Detalles de dicho elemento.
3. Botón que habilita la pantalla de edición de dicho elemento.

Figura A.5

Pantalla de las características específicas del operador

Operador	
MIGUEL ÁNGEL PÉREZ	
Licencia KSDKMXA	Teléfono 7971287471
Contacto 2213492653	Parentesco Esposa
Licencia desde 2024-12-01	Licencia hasta 2025-10-01
Número de INE DDSCNASDI59	NSS 465132165
RFC PEMM011102IX3	

Una vez habilitando el modo de edición en la pantalla anterior se observan los siguientes elementos en la Figura A.6. En el modo edición se activan los campos de texto para modificar las características del elemento, sin embargo, existen algunos elementos que no es posible modificar y se mantienen bloqueados incluso en el modo de edición, dado que la edición de esos datos es accesible desde otras funciones de la aplicación.

1. Nombre o número del elemento que se está editando.
2. Los campos que son editables en forma de texto o de opciones.
3. Botón para guardar los cambios realizados.

Figura A.6

Pantalla de edición de características específicas del operador

The screenshot shows a mobile application interface for editing operator characteristics. At the top, a dark red header bar contains a back arrow, the text "Ver Operador", a circular callout "1" over the text, and a save icon with a circular callout "3". Below the header is a light purple card titled "Operador" with the name "MIGUEL ÁNGEL PÉREZ". The card contains several fields: "Licencia" (KSDKMXA) and "Teléfono" (7971287471); "Contacto" (2213492653) and "Parentesco" (Esposa); "Licencia desde" (2024-12-01) and "Licencia hasta" (2025-10-01); "Número de INE" (DDSCNASDI59) and "NSS" (465132165); and "RFC" (PEMM011102IX3). A circular callout "2" is positioned to the left of the card.

Volviendo a la pantalla de elementos asociados al tractocamión, se encuentra dentro del menú desplegable la opción de registrar cambio de aceite, en esta función, cuando es la primera vez que se abre esta pantalla en un tractocamión, esta opción registra un kilometraje de cero, por lo que, se debe de agregar el kilometraje que tenga el tracto en el momento del registro del cambio de aceite de motor.

En caso de que no sea la primera vez que se abre la pantalla de cambio de aceite de algún tractocamión, se puede observar el kilometraje anterior del último cambio de aceite realizado previamente, además, se debe agregar el nuevo kilometraje con el cual se registrará el cambio de aceite. En la parte inferior de la pantalla, se muestra el historial de cambios de aceite de cada unidad por medio de una lista. Por último, el botón para generar reportes acerca de los últimos cambios de aceite de todos los tractocamiones, este reporte se generará

automáticamente y desplegará las opciones que existen en el dispositivo para que se comparta el reporte pdf. En la Figura A.7, se observa esta pantalla de cambio de aceite a una unidad que se registrará por primera vez un cambio de aceite, esta pantalla contiene los elementos:

1. Título con el tractocamión que se va a registrar el cambio de aceite de motor.
2. Kilometraje anterior, es decir, el kilometraje del último cambio de aceite registrado (en caso de no tener será cero).
3. Campo para ingresar el kilometraje actual con el que registra el cambio de aceite de motor.
4. Botón de guardar el cambio de aceite.
5. Historial de los cambios de aceite del tractocamión que se está visualizando.
6. Botón para generar reporte pdf de todos los tractocamiones sobre el último cambio de aceite realizado.

Figura A.7

Pantalla de cambio de aceite



Volviendo a la pantalla de los elementos asociados, se encuentra dentro del menú desplegable la opción de registrar refacciones. En la pantalla de solicitar refacciones se puede hacer una solicitud de refacciones donde, pueden ser agregadas n cantidad de refacciones, además, se muestra un historial de las refacciones solicitadas por ese tractocamión únicamente.

Cada refacción agregada requiere los campos de “Nombre”, “Descripción” y “Cantidad”, en donde, es obligatorio definir el “Nombre” y la “Cantidad. Al agregar las refacciones necesarias en una solicitud, se debe de presionar el botón inferior para enviar la solicitud de las refacciones, por lo que, se agregará esta solicitud en el historial de solicitudes del tractocamión, esta solicitud se muestra con un color que indica su estado, en donde existen dos opciones de color, el color verde significa que la solicitud ha sido entregada y el color rojo significa que la solicitud no ha sido entregada, en este último caso, se agrega un botón en solicitud, este botón tiene la función de definir la solicitud de las refacciones como entregada.

En cada solicitud de refacciones se puede dar clic al registro para desplegar los detalles de todas las refacciones solicitadas en dicha solicitud, en estos detalles se agregan los detalles de la fecha de solicitud y la fecha de entrega, esta pantalla se observa en la Figura A.8, dicha pantalla contiene los elementos:

1. Indicador del tractocamión que está solicitando refacciones.
2. Botón para agregar refacciones a la solicitud.
3. Refacciones agregadas a la solicitud.
4. Icono para borrar refacción agregada.
5. Historial de las solicitudes realizadas con su indicador de estado.
6. Botón para marcar la solicitud como entregada.
7. Botón para enviar la solicitud de refacciones.

Volviendo a la pantalla inicial de la aplicación móvil, el menú desplegable de la izquierda contiene las siguientes opciones: “Agregar”, “Eliminar”, “Administración”, “Mantenimiento”, “Tráfico”, “Ventas”, “Información” y el icono para los “Ajustes”. A continuación, se presenta la opción de “Agregar” que solicita permisos del departamento de para su acceso.

Figura A.8

Pantalla para solicitar refacciones



A continuación, se explicarán todas las opciones del menú desplegable de la pantalla inicial, iniciando con la opción de “Agregar”, esta despliega de forma inicial la pantalla de “Agregar operador”, en donde, se muestra los campos de texto para introducir toda la información sobre el operador, en la Figura A.9 se visualizan los elementos:

1. Título de la pantalla que indica que se está agregando.
2. Campos de texto para agregar el nuevo elemento.
3. Botón para guardar el nuevo elemento.
4. Menú de navegación que contiene todos los elementos que se pueden agregar: operador, tracto, plana, dolly y extras (marcas y modelos).

Figura A.9

Pantalla para agregar operador

The screenshot shows a mobile application interface for adding an operator. At the top, a dark red header bar contains a back arrow and the text 'Agregar Operador', with a circled '1' next to it. Below the header is a light purple form with several input fields: 'Nombre del Operador*', 'Licencia del Operador*', 'Teléfono del Operador*', 'Contacto del Operador:', 'Parentesco del Contacto:', 'Licencia desde:' (with a date picker), 'Licencia hasta:' (with a date picker), 'Número de INE del Operador:', 'NSS del Operador:', and 'RFC del Operador:'. A red button labeled 'Guardar Operador' is at the bottom of the form, with a circled '3' next to it. At the very bottom of the screen is a navigation bar with icons for 'Operador', 'Tracto', 'Plana', 'Dolly', and 'Extras', with a circled '4' next to the 'Extras' icon. A circled '2' is also placed to the left of the 'Contacto del Operador:' field.

En la opción antes mencionada de extras, habilita la posibilidad de agregar elementos secundarios que son: modelos de tractos, marcas de tractocamión, marcas de plana y marcas de dolly, siguiendo con la misma dinámica de las pantallas de agregar ya mencionadas.

La opción de “Eliminar”, esta función despliega de forma inicial la opción de eliminar tractocamiones, cada registro se muestra de forma general, es decir, con información clave que hace referencia al elemento a eliminar, por ejemplo, en caso de un tractocamión se muestra la leyenda “tractocamión 100”. Las pantallas de “Eliminar” siguen el mismo concepto de interfaz, estas pantallas contienen los siguientes elementos que se muestran en la Figura A.10 que es el caso de eliminar tractocamión.

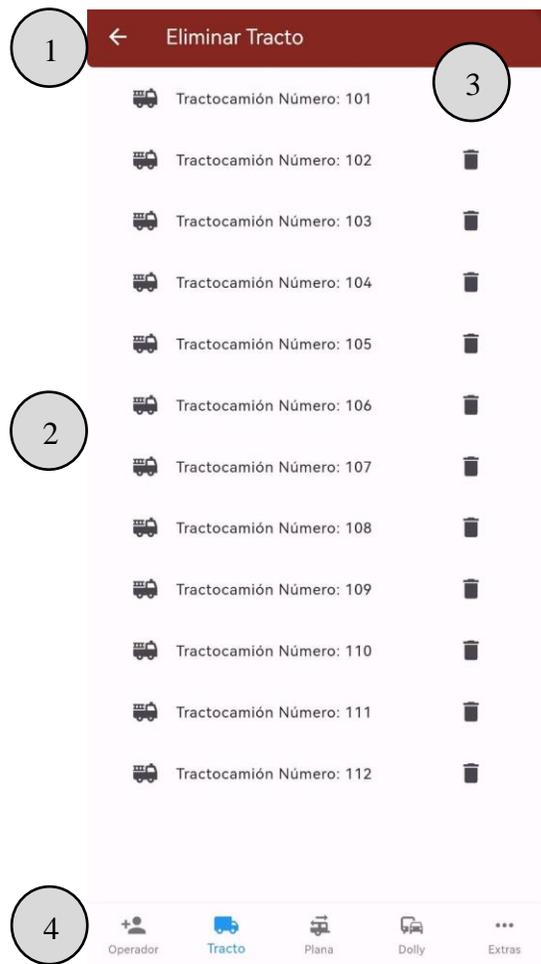
1. Título del tipo de elemento que se va a eliminar.
2. Lista de elementos registrados.

3. Botón de eliminar.
4. Menú de navegación que contiene las opciones de Operador, Tracto, Plana, Dolly y extras, este último, solo muestra los modelos de tractocamiones.

Se implementó una ventana emergente de confirmación con el fin de evitar incidentes en las pantallas de eliminar con un mensaje de confirmación del elemento que se va a eliminar.

Figura A.10

Pantalla para eliminar operador



La siguiente opción del menú desplegable es la sección del departamento de Administración, para acceder a este menú se debe de tener autorización previa, en caso de que el usuario con el que ha iniciado sesión en la aplicación *Freightem* no posea permiso se mostrará un mensaje que indica la falta de permiso para acceder a la sección en cuestión.

La pantalla inicial del departamento de Administración es la pantalla de “Asignar Operador”, en esta pantalla se pueden reasignar a los operadores que están asociados a los tractocamiones. Para reasignar un operador se debe de buscar el tractocamión deseado y dar clic en el registro, se mostrará una ventana emergente ilustrada en la Figura A.11, por medio de una lista se debe de buscar el nuevo operador que se desee asignar al tractocamión en cuestión, una vez seleccionado el nuevo operador se debe de presionar el botón de guardar cambios para cambiar exitosamente el operador del tractocamión. Cabe mencionar que el operador ya debe estar agregado dentro de la aplicación para poder asignarlo a una unidad. La pantalla de reasignar el operador se observa en la Figura A.12 que contiene los elementos:

1. Título de la pantalla.
2. Buscador de tractocamiones por medio de su número económico.
3. Tractocamiones existentes.
4. Menú de navegación de las funciones del departamento de administración.
5. Botón para actualizar los registros.

Figura A.11

Ventana emergente para modificar el operador asignado

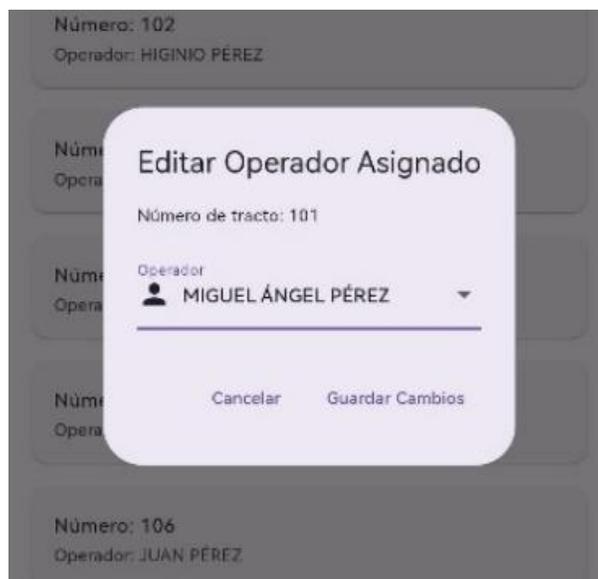
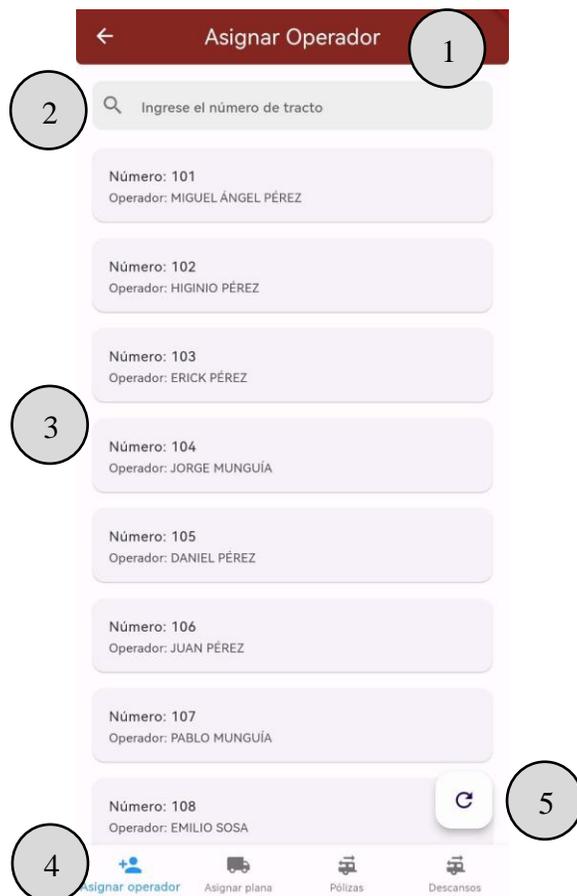


Figura A.12

Pantalla para reasignar operador

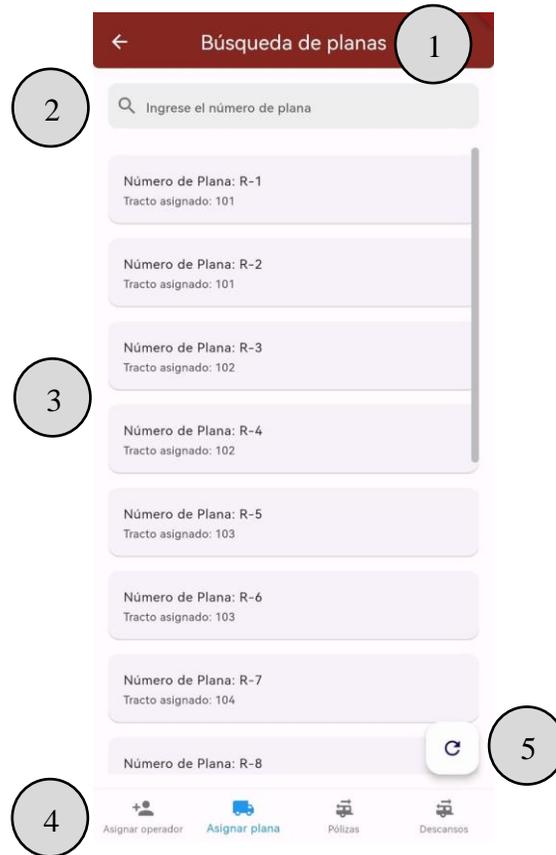


La siguiente opción en el menú de navegación de administración es la pantalla de asignar plana, en esta pantalla de forma inicial se despliega una lista de todas las planas existentes, en donde, se puede editar el tractocamión al que están relacionadas, la pantalla de asignar plana se observa en la Figura A.13, esta pantalla contiene los elementos:

1. Título de la pantalla.
2. Buscador de plana con entrada numérica en donde el rango de las opciones es el número de planas existentes.
3. Lista de planas existentes con su número identificador y el número económico del tractocamión relacionado.
4. Menú de navegación de las funciones del departamento de administración.
5. Botón de actualizar listas.

Figura A.13

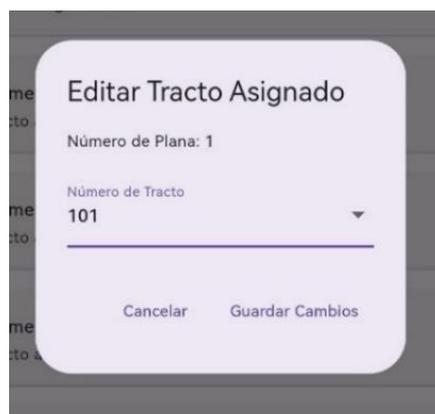
Pantalla para asignar planas



Al dar clic en alguna plana se desplegará la siguiente pantalla emergente mostrada en la Figura A.14, en donde es posible cambiar el tractocamión asignado seleccionando una unidad de la lista y confirmar el cambio.

Figura A.14

Ventana emergente para modificar las planas asignadas

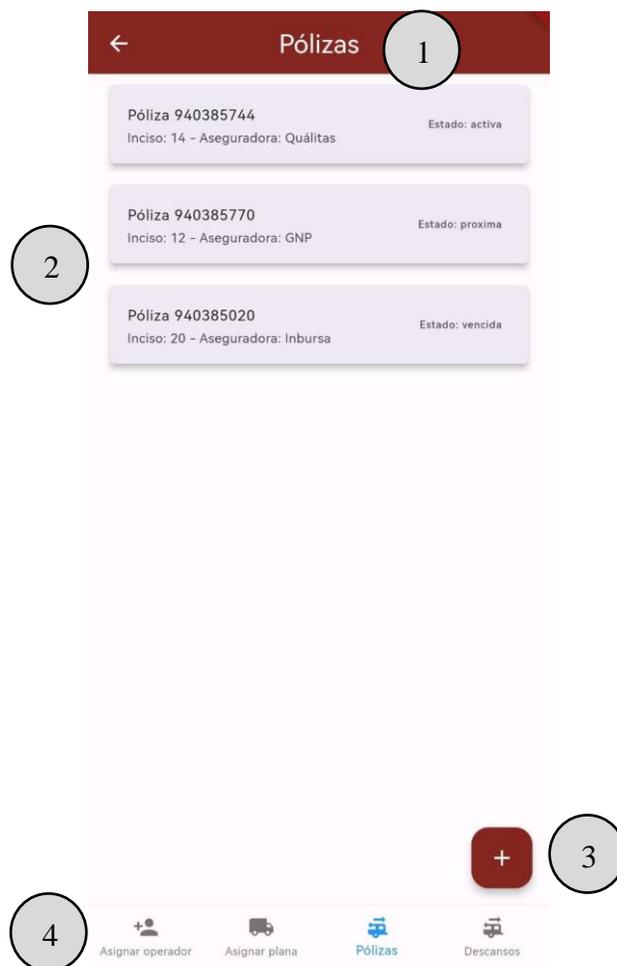


La siguiente opción del departamento de administración es la gestión de pólizas. En esta pantalla, se muestran todas las pólizas registradas, se puede dar clic a cada póliza para visualizar sus detalles, en la misma ventana emergente de los detalles es posible asociar nuevos tractocamiones a la póliza, esta pantalla de las pólizas se observa en la Figura A.15 que contiene los elementos:

1. Título de la pantalla.
2. Pólizas registradas.
3. Botón para registrar nuevas pólizas.
4. Menú de navegación de las funciones del departamento de administración.

Figura A.15

Pantalla de pólizas registradas



Al entrar a la pantalla de registrar póliza, se muestra varios campos de texto para completar con los datos de la póliza nueva, esta pantalla se visualiza en la Figura A.16 que contiene los elementos:

1. Título de la pantalla.
2. Campos a llenar sobre la póliza.
3. Botón para guardar.

Figura A.16

Pantalla para registrar póliza

The screenshot shows a mobile application interface for registering a policy. At the top, there is a dark red header bar with a white back arrow on the left, a white circle containing the number '1' in the center, and the text 'Registrar póliza' on the right. Below the header is a light purple rounded rectangle titled 'Datos de la póliza'. This rectangle contains several text input fields: 'Número de póliza*', 'Número de inciso*', 'Aseguradora*', 'Vigencia desde*', 'Hora', 'Vigencia hasta*', and 'Hora'. A white circle containing the number '2' is positioned to the left of the 'Aseguradora*' field. At the bottom right of the screen, there is a dark red square button with a white floppy disk icon, and a white circle containing the number '3' is positioned to its right.

La siguiente función del área de administración es asignar una unidad en descanso, en la pantalla se muestran un menú con las opciones de “Asignar descanso” y “Remover descanso”, en este caso, en la opción de asignar descanso se muestran todos los tractocamiones existentes con su respectivo estado, para asignar el descanso se debe de dar

clic en la unidad deseada y confirmar el descanso, esta pantalla se observa en la Figura A.17 que contiene los elementos:

1. Título de la pantalla.
2. Submenú de descanso.
3. Tractocamiones existentes.
4. Estado de los tractocamiones.
5. Menú de navegación de las funciones del departamento de administración.

Figura A.17

Pantalla para asignar descanso a los tractocamiones



La opción de remover descanso solo muestra los tractocamiones que están en descanso, en caso de no tener ningún tractocami3n en descanso se mostrará la leyenda “No hay unidades en descanso”. Para remover el descanso de un tractocami3n se debe dar clic en la unidad

deseada y confirmar la acción, de esta forma se actualiza el tractocamión con el estado de “Vació”.

La pantalla para remover descansos se observa en la Figura A.18 que contiene los elementos:

1. Título de la pantalla.
2. Submenú de descanso.
3. Tractocamiones existentes.
4. Estado de los tractocamiones.
5. Menú de navegación de las funciones del departamento de administración.

Figura A.18

Pantalla de tractocamiones en descanso



Volviendo al menú desplegable lateral de la pantalla inicial de la aplicación, la siguiente opción a abordar son las funciones del departamento de mantenimiento, para acceder a esta sección se debe de poseer autorización, en caso de que no se negará el acceso.

La primera función del departamento de mantenimiento es generar bitácoras de mantenimiento, en esta bitácora se deben llenar los campos como el tipo de elemento al que se le hace mantenimiento (tracto, plana o dolly), agregar su número identificador, en caso de que el mantenimiento sea a un tractocamión se habilitará el kilometraje, en la descripción del mantenimiento se debe mencionar de que consta el mantenimiento en cuestión por ejemplo, un engrasamiento de quinta rueda, cambio de llantas, revisión de niveles, etc. Por último, el taller (interno, externo o tránsito) donde se realizará el mantenimiento con un campo extra de detalles para especificar el lugar. La pantalla para registrar una bitácora de mantenimiento se observa en la Figura A.19 que contiene los elementos:

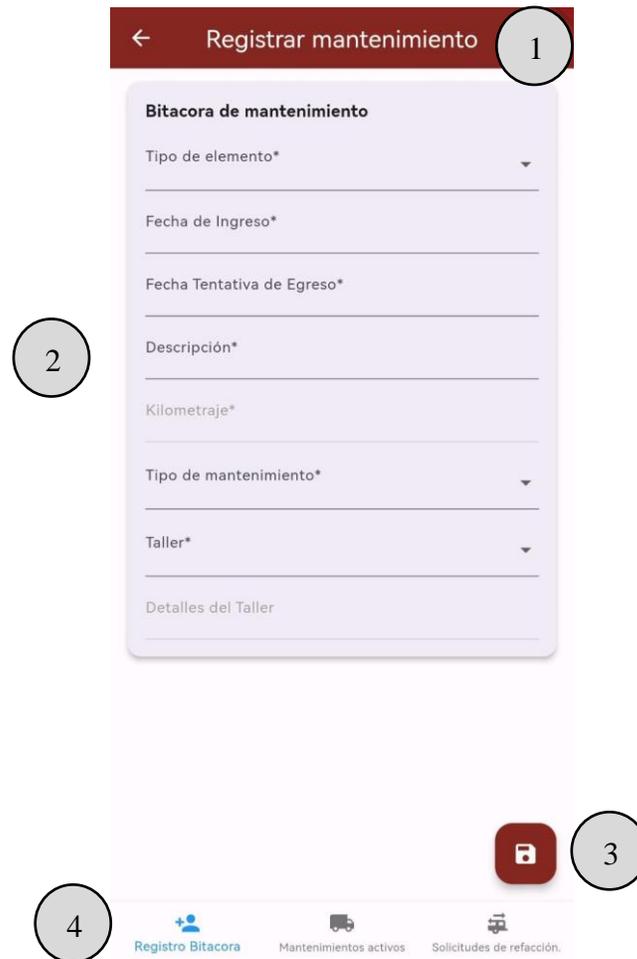
1. Título de la pantalla.
2. Campos de la bitácora.
3. Botón para guardar bitácora.
4. Menú de navegación de las funciones del departamento de mantenimiento.

La siguiente opción del departamento de mantenimiento es ver los mantenimientos realizados, en esta pantalla de muestran todas las bitácoras de mantenimiento, en caso de estar activo el mantenimiento, es decir, que no ha sido definido como completado, la bitácora tendrá un botón para completar el mantenimiento solicitando la fecha real de egreso. Para ver los detalles de la bitácora de mantenimiento se debe dar clic al mantenimiento que se desea visualizar. La pantalla de mantenimientos se observa en la Figura A.20 que contiene los elementos:

1. Título de la pantalla.
2. Mantenimientos registrados.
3. Botón para marcar como completado el mantenimiento.
4. Menú de navegación de las funciones del departamento de mantenimiento.

Figura A.19

Pantalla para registrar bitácoras de mantenimiento

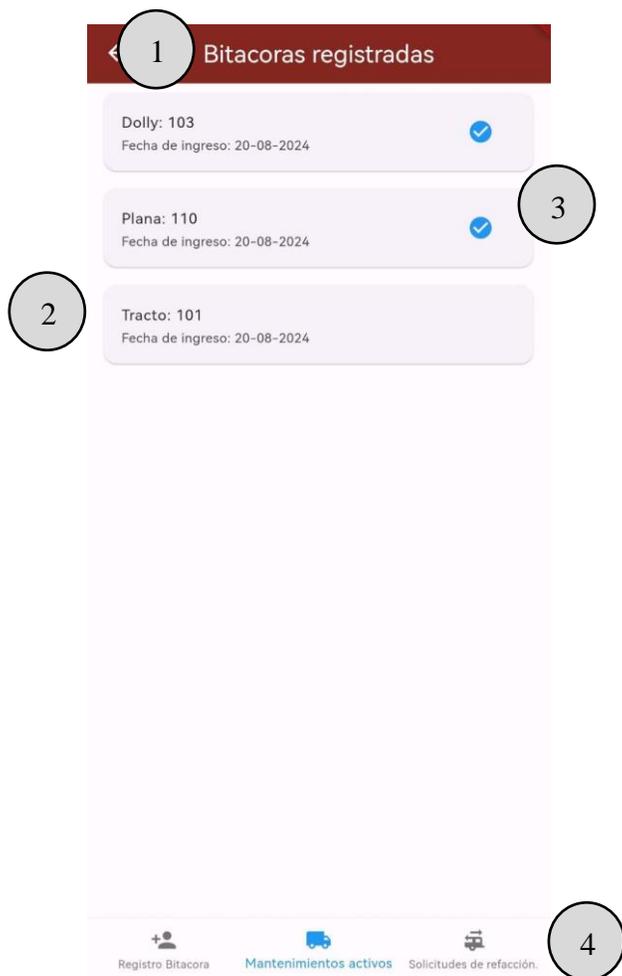


La siguiente opción del departamento de mantenimiento es ver las solicitudes de refacciones, en esta pantalla se muestran todas las solicitudes realizadas con un filtro de búsqueda por su estado, se puede cambiar entre sus diferentes opciones (todos, entregado o pendiente). Para observar los detalles de la solicitud con sus refacciones solicitadas se debe hacer clic en la solicitud deseada. Esta ventana se observa en la Figura A.21 que contiene los elementos:

1. Título de la pantalla.
2. Filtro de búsqueda por el estado de la solicitud.
3. Solicitudes de refacciones realizadas.
4. Botón para marcar como entregada la solicitud.
5. Menú de navegación de las funciones del departamento de mantenimiento.

Figura A.20

Pantalla de los mantenimientos registrados



Volviendo al menú desplegable de la pantalla inicial de la aplicación, la siguiente opción a abordar son las funciones del departamento de tráfico, para acceder a esta sección se debe de poseer permiso para acceder, en caso de que no se negará el acceso.

La primera función del departamento de tráfico es ver el estado de disponibilidad de los tractocamiones (unidades), en esta pantalla se visualizan todos los tractocamiones existentes, a su derecha se observa un icono que expresa su estado de disponibilidad, entre las opciones de este estado existen: vacío, tránsito, taller y descanso. Además, un filtro de búsqueda por medio del estado para visualizar las unidades con un estado de disponibilidad en común.

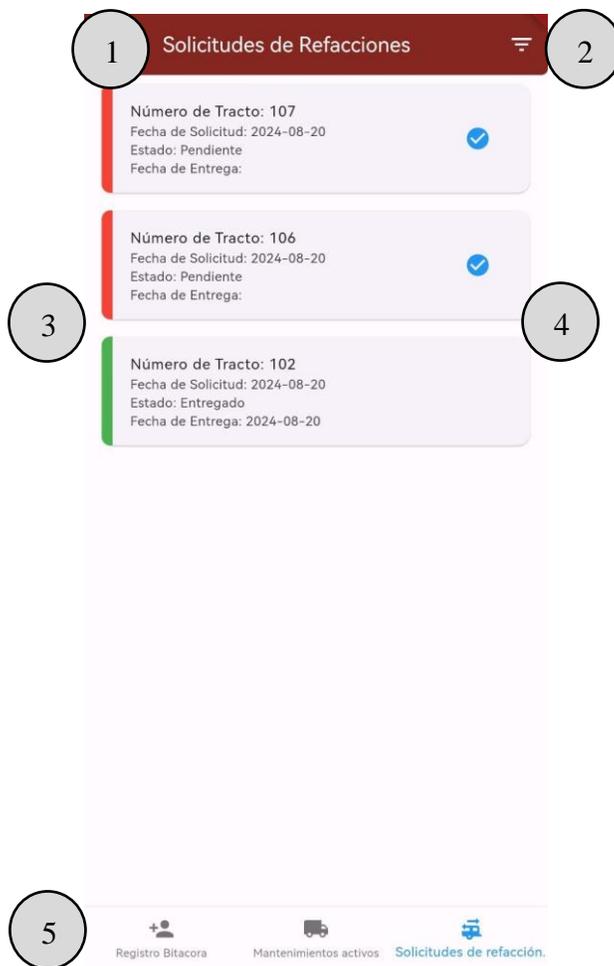
Esta pantalla se observa en la Figura A.22 que contiene los elementos:

1. Título de la pantalla.

2. Filtro de búsqueda por estado de disponibilidad.
3. Tractocamiones existentes.
4. Icono representativo que indica estado de disponibilidad.
5. Menú de navegación de las funciones del departamento de tráfico.

Figura A.21

Pantalla de las solicitudes de refacciones



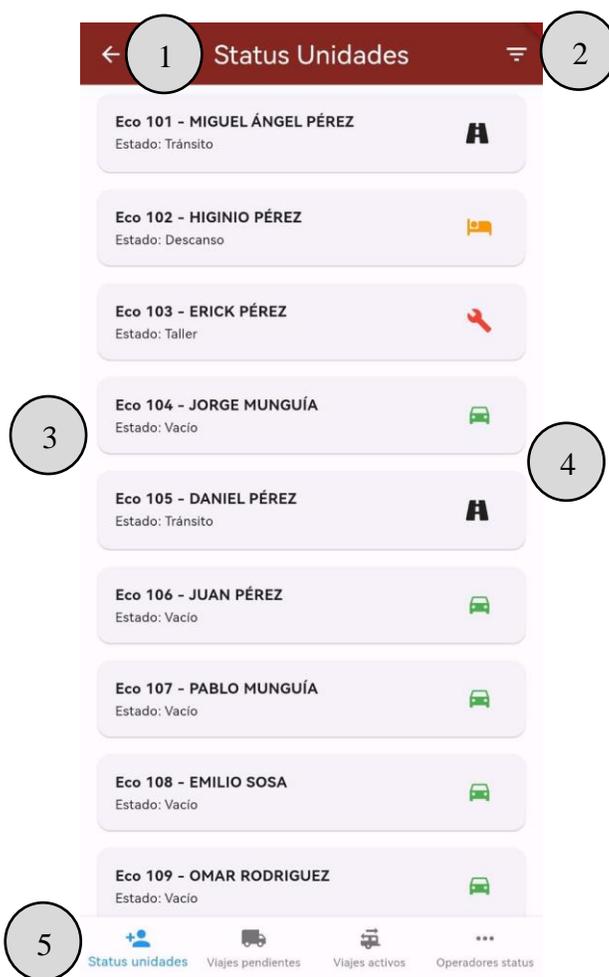
La siguiente función del departamento de tráfico es asignar un tractocamión a un viaje (viaje agendado por departamento de ventas). En esta pantalla se visualizan los viajes pendientes, es decir, viajes que aún no han sido comenzados, por lo que, se debe de asignar un tractocamión óptimo para cubrir el viaje. En cada registro se visualizan detalles del viaje en donde se debe prestar suma atención a la configuración vehicular (sencillo o full), a su lado, un icono para asignar el tractocamión a un viaje. Es posible observar los detalles completos del viaje al dar clic al registro del viaje.

La pantalla de viajes pendientes se observa en la Figura A.23 que contiene los elementos:

1. Título de la pantalla.
2. Viajes pendientes.
3. Icono para asignar tractocamión al viaje.
4. Menú de navegación de las funciones del departamento de tráfico.

Figura A.22

Pantalla del estado de disponibilidad de las unidades

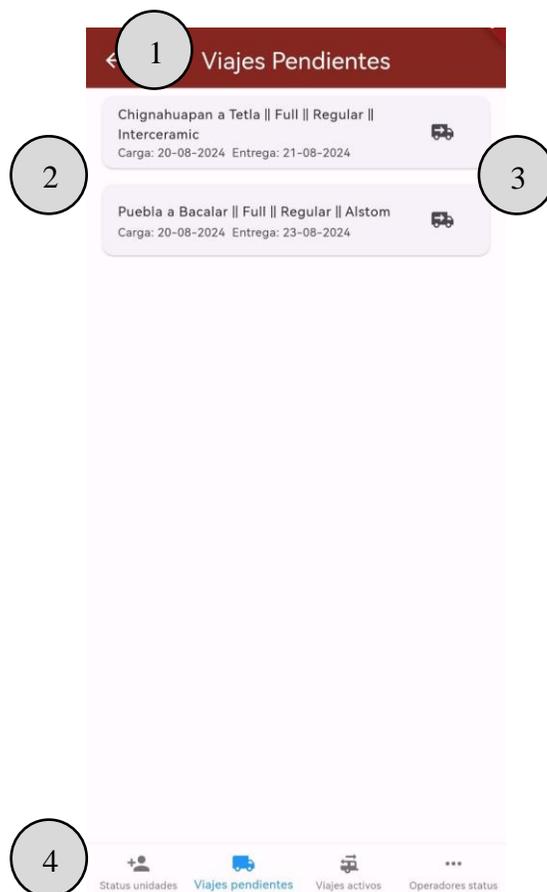


Cuando se hace clic en el icono para asignar una unidad al viaje se despliega una lista de tractocamiones que se encuentran disponibles para operar, y además, que cumplen con la configuración vehicular solicitada, es decir, si el viaje a asignar solicita una configuración vehicular “full”, se mostrarán en una ventana emergente todas las unidades disponibles con

esta configuración, para asignar un tractocamión solo se debe dar clic en la unidad y esta se asignará al viaje.

Figura A.23

Pantalla de viajes pendientes



La ventana emergente que muestra los tractocamiones también muestra al operador de la unidad con un indicador en cuestión de la vigencia de su licencia con un modelo de semáforo, es decir, si el indicador es verde significa que su licencia está en orden, si está en naranja significa que está próxima a vencer y si está en rojo significa que la licencia ha vencido. Esta ventana emergente se observa en la Figura A.24 que contiene los elementos:

1. Título de la ventana.
2. Unidades disponibles que cumplen con la configuración vehicular solicitada.
3. Indicador de estado de la licencia del operador.

Figura A.24

Ventana emergente para asignar tractocamión al viaje



La siguiente función del departamento de tráfico es ver los viajes activos. En la pantalla de viajes activos, se observan todos los viajes que están en curso, cada registro muestra datos generales del viaje, como el número económico de la unidad asignada al viaje, origen, destino y fechas de carga y entrega, además, cada registro presenta dos botones, el primero para llamar al operador del viaje y el segundo para marcar como entregado el viaje. Además, se pueden ver los detalles completos del viaje, que a diferencia de los detalles de viajes pendientes, este agrega los detalles del tractocamión y elementos asociados como la plana (s), dolly en caso de aplicar y el nombre del operador.

Cuando se marca un viaje como entregado, se solicita el dato si existió alguna incidencia durante el viaje, este dato puede dejarse vacío en caso de que no haya existido ningún percance que haya entorpecido la entrega.

La pantalla de viajes activos se observa en la Figura A.25 que contiene los elementos:

1. Título de la pantalla.
2. Viajes activos
3. Icono para llamar al operador del viaje.
4. Icono para marcar como entregado el viaje.
5. Menú de navegación de las funciones del departamento de tráfico.

Figura A.25

Pantalla de viajes pendientes



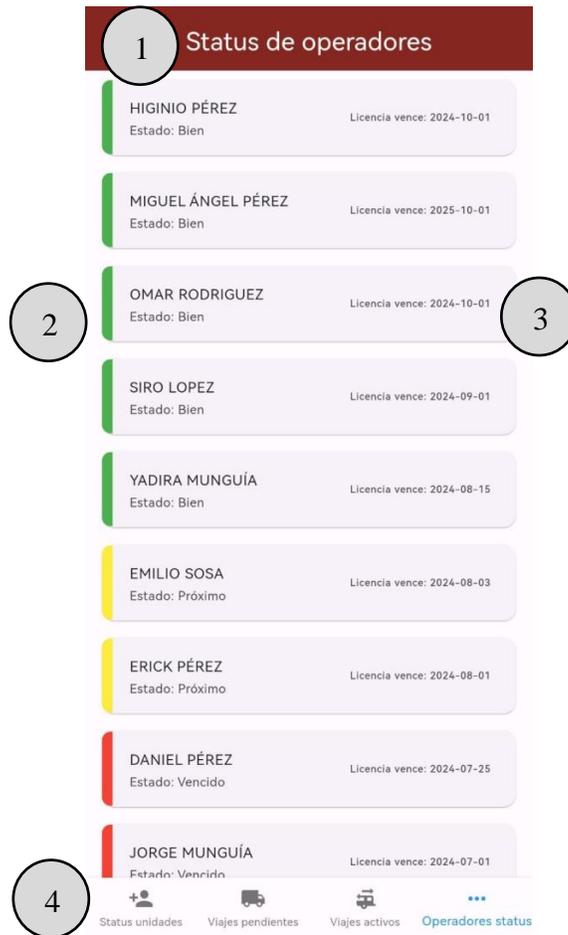
La siguiente y última función del departamento de tráfico es ver el estado de la licencia de los operadores dependiendo de la vigencia de su licencia. En esta pantalla, se muestra todos los operadores existentes con un indicador dependiendo de su color, si el indicador es verde significa que la licencia está en orden, si es naranja significa que está próxima a vencer, si es rojo significa que está vencida la licencia de conducir y si es gris significa que no se ha ingresado la vigencia de la licencia correctamente del respectivo operador.

Esta pantalla se observa en la Figura A.26 que contiene los elementos:

1. Título de la pantalla.
2. Operadores existentes con el indicador de estado de licencia.
3. Vencimiento de su licencia de conducir.
4. Menú de navegación de las funciones del departamento de tráfico.

Figura A.26

Pantalla del estado de la licencia de los operadores



Volviendo al menú desplegable de la pantalla inicial de la aplicación, la siguiente opción a abordar son las funciones del departamento de ventas, para acceder a esta sección se debe de poseer permiso, en caso de que no, se negará el acceso. La primera función del departamento de ventas es revisar los viajes finalizados. En esta pantalla, se pueden ver todos los viajes entregados mediante registros. Cada viaje recientemente terminado mostrará un ícono que permite calificar el servicio con base en el criterio del cliente. Si el viaje ya ha sido calificado, el ícono desaparecerá.

Para ver los detalles completos del viaje es necesario dar clic en el viaje que se desea visualizar para que se muestren estos detalles. Además, es posible generar reportes pdf por medio del botón inferior, este reporte solicitará seleccionar un tractocamión para generar el reporte de viajes realizados, posteriormente se generará automáticamente desplegando una ventana para compartir dicho reporte.

Esta pantalla se muestra en la Figura A.27 que contiene los elementos:

1. Título de la pantalla.
2. Viajes terminados.
3. Icono para calificar el viaje
4. Botón para generar reporte de viajes.
5. Menú de navegación de las funciones del departamento de ventas.

Figura A.27

Pantalla de viajes terminados



La segunda función del departamento de ventas es programar los viajes. En la pantalla de agendar viaje, se muestra un formulario con todos los campos necesarios para agendar un viaje, en donde, es de importancia tener cuidado con estos datos debido a que no pueden ser modificados una vez guardados.

La pantalla para agendar viajes se observa en la Figura A.28 que contiene los elementos:

1. Título de la pantalla.

2. Campos a rellenar acerca del viaje.
3. Botón para guardar viaje.
4. Menú de navegación de las funciones del departamento de ventas.

Figura A.28

Pantalla para agendar viajes

1

← Agendar viaje

Datos Generales

Origen*

Destino*

Cliente*

Mercancia*

Tipo de servicio*

Configuración vehicular*

Datos de Carga

Fecha de Carga*

Hora de Carga

Fecha de salida a ruta

Datos de Entrega

Fecha de Entrega*

Guardar

4

Viajes terminados

Agendar viaje

3

De esta forma, se da por completada la navegación entre las secciones de los departamentos. Por último, en la pantalla inicial de la aplicación, se muestra el icono de cerrar sesión, este icono cerrará la sesión iniciada, lo que redirigirá de nuevo a la pantalla de inicio de sesión.

De esta manera concluyendo con todas las funciones que la aplicación móvil *Freightem* tiene para ofrecer.