

**SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ADMINISTRATIVA DE LA
COOPERATIVA INTEGRAL DE TRANSPORTE ANDINO TRANSANDINO**



**INSTITUCIÓN UNIVERSITARIA
COLEGIO MAYOR DEL CAUCA**

SOFIA MUÑOZ MUÑOZ CIFUENTES

ANYELA NATALIA MARTINEZ M.

KEVYN DANIEL BASTIDAS CERON

INSTITUCIÓN UNIVERSITARIA COLEGIO MAYOR DEL CAUCA

FACULTAD DE INGENIERÍA

TECNOLOGÍA DE DESARROLLO DE SOFTWARE

POPAYÁN, CAUCA

2025

**SISTEMA DE INFORMACIÓN PARA LA GESTIÓN ADMINISTRATIVA DE LA
COOPERATIVA INTEGRAL DE TRANSPORTE ANDINO TRANSANDINO**

**proyecto de grado presentado como requisito para optar al título
de Tecnólogo en Desarrollo de Software**

SOFIA MUÑOZ CIFUENTES

ANYELA NATALIA MARTINEZ

KEVYN DANIEL BASTIDAS

ASESOR

ING. GUSTAVO EDUARDO GIL

INSTITUCIÓN UNIVERSITARIA COLEGIO MAYOR DEL CAUCA

TECNOLOGÍA DE DESARROLLO DE SOFTWARE

POPAYÁN, CAUCA

2025

AGRADECIMIENTOS

Agradecemos a Dios por la fortaleza, la constancia y la sabiduría que hicieron posible culminar este proceso académico.

Expresamos nuestro agradecimiento a los jurados evaluadores Manuel Arturo Legarda y Victor Hugo Pinto por el tiempo dedicado a la revisión de este trabajo de grado, así como por sus observaciones y recomendaciones, las cuales contribuyeron significativamente al fortalecimiento y rigor académico del proyecto.

De manera especial, agradecemos al docente Gustavo Eduardo Gil asesor del proyecto por su acompañamiento constante, orientación metodológica y aportes técnicos, fundamentales para el adecuado desarrollo y consolidación de este trabajo de investigación.

Reconocemos nuestro esfuerzo, compromiso y dedicación como equipo de trabajo, quienes asumimos con responsabilidad cada una de las etapas del proyecto, demostrando disciplina, colaboración y crecimiento profesional durante el proceso académico.

A nuestras familias, extendemos un sincero agradecimiento por el apoyo incondicional, la comprensión y la motivación permanente, siendo un pilar fundamental para alcanzar este logro académico.

Finalmente, expresamos nuestro agradecimiento a la Cooperativa Integral de Transporte Andino TransAndino, en cabeza de su gerente, por permitir el desarrollo del proyecto, facilitar el acceso a la información y brindar el apoyo necesario para comprender sus procesos administrativos, lo cual hizo posible la formulación de una solución tecnológica acorde con las necesidades reales de la organización.

RESUMEN

El presente proyecto de grado aborda el desarrollo e implementación de un sistema de información web para la Cooperativa Integral de Transporte Andino TransAndino, ubicada en la ciudad de Popayán. La organización enfrentaba dificultades operativas debido a la gestión manual de su información administrativa, apoyada en hojas de cálculo y archivos físicos, lo cual generaba inconsistencias, duplicidad de datos, pérdida de información y riesgos legales por el vencimiento no detectado de documentación obligatoria (como SOAT y revisión técnico-mecánica).

El objetivo principal fue desarrollar una solución tecnológica que centralizara y optimizara el registro, control y consulta de la información relacionada con vehículos, conductores, asociados y mantenimientos. Para ello, se empleó una metodología de desarrollo ágil gestionada mediante Kanban, utilizando tecnologías modernas de código abierto: React y Vite para el frontend, Node.js y Express para el backend, y MySQL como sistema gestor de base de datos relacional.

Los resultados obtenidos demuestran que el sistema permite una administración eficiente de los datos, facilitando el seguimiento de historiales de mantenimiento y la gestión de vencimiento de documentos. Las pruebas funcionales y de usabilidad realizadas con el usuario administrador y usuarios finales validaron la eficacia de la herramienta, evidenciando una mejora significativa en la organización de la información y una reducción en los tiempos de gestión operativa. En conclusión, el sistema aporta a la modernización tecnológica de la cooperativa, garantizando la integridad de los datos y apoyando la toma de decisiones estratégicas.

ABSTRACT

This degree project addresses the development and implementation of a web-based information system for the Cooperativa Integral de Transporte Andino TransAndino, located in the city of Popayán. The organization faced operational difficulties due to the manual management of its administrative information, relying on spreadsheets and physical files. This led to inconsistencies, data duplication, information loss, and legal risks associated with undetected expiration of mandatory documentation (such as mandatory insurance and mechanical inspections).

The main objective was to develop a technological solution to centralize and optimize the registration, control, and consultation of information related to vehicles, drivers, associates, and maintenance records. To achieve this, an agile development methodology managed via Kanban was employed, utilizing modern open-source technologies: React and Vite for the frontend, Node.js and Express for the backend, and MySQL as the relational database management system.

The results obtained demonstrate that the system allows for efficient data administration, facilitating the tracking of maintenance histories and the management of document expiration alerts. Functional and usability tests conducted with the administrator and end-users validated the tool's effectiveness, evidencing a significant improvement in information organization and a reduction in operational management times. In conclusion, the system contributes to the technological modernization of the cooperative, ensuring data integrity and supporting strategic decision-making.

TABLA DE CONTENIDO

Contenido

Introducción	10
Planteamiento del Problema	11
Justificación	13
Objetivos.....	15
Objetivo General	15
Objetivos Específicos	15
Alcance	16
Marco Referencial.....	18
Cooperativa de Transporte Público TransAndino	18
Marco Teórico.....	19
Marco Conceptual.....	21
Sistema de Información.....	21
Digitalización	22
Transformación Digital.....	22
Gestión Administrativa.....	22
Base de Datos Relacional	23
Metodología Kanban	23
Usuario Administrador	23
Seguridad de la Información.....	24
Documentación Vehicular.....	24
Mantenimiento Preventivo	24
Antecedentes	25
Tecnologías utilizadas en el desarrollo del sistema	28
Marco metodológico de desarrollo del software	31
Metodología kanban	31
Metodología ágil para el desarrollo del sistema	34
Capítulo 1. Levantamiento de requisitos	36
Proceso de levantamiento de requisitos	37
Análisis de requisitos.....	40
Funcionalidades del sistema identificados.....	44
Requerimientos Funcionales	46
Requerimientos No Funcionales.....	48

Restricciones.....	49
Casos de Uso.....	50
Capítulo 2 - Implementación de la Aplicación Web	53
Arquitectura del Sistema	53
Organización de la arquitectura.....	54
Organización del Frontend	55
Backend – Node.js / Express.....	56
Diagrama Entidad Relación	58
Mockups e Interfaces	62
Desarrollo del Sistema	64
Implementación del Frontend	66
Implementación del Backend.....	72
Flujo de Funcionamiento del Backend.....	76
Capítulo 3. Validación del sistema TransAndino	79
Pruebas de Validación del Backend	79
Integración frontend–backend	84
Plan de Pruebas.....	88
Pruebas Funcionales.....	89
Pruebas de Usabilidad con Usuario Administrador.....	93
Pruebas de Usabilidad con Usuarios Finales.....	95
Análisis de Resultados	97
Análisis de los Resultados de las Pruebas de Usabilidad	97
Análisis de funcionamiento del sistema	100
Conclusiones	102
Recomendaciones	103
Bibliografía.....	105

ÍNDICE DE FIGURAS

Figura 1 - Árbol de Problemas Fuente: Propio	13
Figura 2 - Logo de la Cooperativa.....	19
Figura 3 – Tablero Kanban y flujo de trabajo de la metodología Kanban.....	32
Figura 4 - Ciclo de desarrollo de la metodología ágil.....	35
Figura 5 - Ciclo de desarrollo de la metodología ágil. Fuente - Popio.....	39
Figura 6 - Diagrama de flujo de la arquitectura del sistema. Fuente - Propio	54
Figura 7 - Diagrama de paquetes front-end. Fuente - Propio	55

Figura 8 - Diagrama de paquetes Backend Fuente - Propio.....	57
Figura 9 - Diagrama entidad relación Fuente - Propio.....	59
Figura 10 - Diagrama Clases en alto Nivel Fuente - propio	60
Figura 11 - Diagrama Casos de Uso- Gestión de Vehiculos. Fuente – Propio	61
Figura 12 - Diagrama de secuencia. Fuente - Propio.....	62
Figura 13 - Mockup preliminar. Fuente - Propio.	63
Figura 14 - Mockup Secciones. Fuente - Propio.....	64
Figura 15 - Página Principal. Fuente - Propio.....	66
Figura 16 - Tablero de trabajo Figma. Fuente - Propio.....	67
Figura 17 - paquetes Forn-end. Fuente: Propio.....	68
Figura 18 - Átomos – Atomic Design Frontend.....	69
Figura 19 - Formularios registro de vehículo. Fuente - Propio.....	69
Figura 20 – Molécula Atomic design Frontend. Fuente – Propio.	70
Figura 21- Formularios registro de vehículo. Fuente: Propio.	71
Figura 22 -Dashboard. Fuente -Propio.	71
Figura 23 - Estructura del Backend. Fuente - Propio.....	73
Figura 24 – Diagrama físico en MYSQL de la base de datos. Fuente: Propio.	75
Figura 25 – Flujo del Backend. Fuente - Propio.	77
Figura 26 – Estructura backend Fuente - Propio.	78
Figura 27 – Pruebas backend _ Listar Personas. Fuente - Propio.....	80
Figura 28 – Pruebas backend _ Buscar Persona Por Cédula. Fuente - Propio	81
Figura 29 – Pruebas backend _ Crear Persona. Fuente- Propio	82
Figura 30– Pruebas backend _ Crear Persona Error Fuente - Propio	83
Figura 31– Pruebas backend Actualizar Personas Fuente: <i>Propio</i>	83
Figura 32– Pruebas backend Eliminar Personas Fuente: <i>Propio</i>	84
Figura 33 – Pruebas integración Frontend y Backend - Actualizar ConductorPersonas Fuente -Propio.	86
Figura 34– Diagrama de integración – Comunicación del sistema backend Fuente - Propio.	87
Figura 35- Funciones. Fuente - Propio.....	91
Figura 36 - Funcionalidad. Fuente -Propio.	92
Figura 37- <i>Proceso del test de Usabilidad con Usuario Administrador.</i> Fuente - <i>Propio</i>	95
Figura 38- Gráfica de Eficacia en Test de Usuario Administrador. Fuente – Propio.	97
Figura 39- Gráfica de resultados de Eficiencia en Test de Usuario Administrador. Fuente – Propio.	98
Figura 40 - Tabla de tiempos con usuarios finales . Fuente - Propio.	98
Figura 41 – Gráfico de Eficiencia Test con Usuarios Finales. Fuente -Propio.	99

ÍNDICE DE TABLAS

Tabla 1 - Plataformas tecnológicas pra la gestión administrativa y operativa en el sector transporte Fuente: Propio	22
Tabla 2 - Personal Involucrado Fuente: Propio	38
Tabla 3 - Acrónimos y Abreviaturas Fuente: Propio	38
Tabla 4 - Funcionalidades del Sistema de Información Fuente: Propio	39
Tabla 5 - Requerimientos Funcionales Fuente: Propio	41
Tabla 6 - Requerimientos No Funcionales <i>Fuente: Propio</i>	43
Tabla 7 - Resumen de Casos de Uso – Sistema de Información Cooperativa TransAndino Fuente:Propio.	46

Introducción

En la actualidad, las organizaciones enfrentan un entorno cada vez más dinámico y competitivo, en el cual la gestión adecuada de la información se ha convertido en un elemento fundamental para garantizar la eficiencia operativa y la sostenibilidad institucional. El uso de sistemas de información permite optimizar procesos administrativos, centralizar datos y facilitar la toma de decisiones, especialmente en organizaciones que dependen del manejo constante de información operativa y documental (Laudon & Laudon, 2020).

En el sector del transporte público, la correcta administración de la información relacionada con vehículos, conductores, asociados y documentación legal resulta indispensable para asegurar el cumplimiento normativo, la continuidad del servicio y la seguridad operativa. Sin embargo, muchas cooperativas aún gestionan estos procesos mediante métodos manuales o herramientas básicas como hojas de cálculo, lo que incrementa el riesgo de errores, la duplicidad de datos y la pérdida de información relevante, además de dificultar el acceso oportuno a registros históricos (Stair & Reynolds, 2017).

La Cooperativa Integral de Transporte Andino TransAndino no es ajena a esta problemática. Actualmente, gran parte de su gestión administrativa se realiza de forma manual, lo cual limita el control eficiente de la información y afecta el rendimiento del personal administrativo. Esta situación reduce la capacidad de la cooperativa para responder de manera ágil a las exigencias del sector y para contar con información confiable que respalde los procesos de control y seguimiento interno, aspectos clave en la administración organizacional moderna (Sommerville, 2016).

Ante este contexto, el desarrollo de un sistema de información se presenta como una solución tecnológica que permite centralizar, organizar y administrar de manera estructurada la información administrativa de la cooperativa. La implementación de una aplicación web facilita el registro, la consulta y la actualización de datos, contribuyendo a la reducción de errores operativos y al fortalecimiento de la gestión administrativa mediante el uso eficiente de la tecnología (Pressman & Maxim, 2020).

De esta manera, el presente proyecto tiene como finalidad diseñar e implementar un sistema de información para la gestión administrativa de la Cooperativa Integral de Transporte Andino TransAndino, orientado a optimizar el manejo de la información y contribuir a la modernización tecnológica de la organización. Esta iniciativa se enmarca dentro de los procesos de transformación digital, los cuales buscan mejorar la eficiencia institucional y adaptar las organizaciones a las exigencias actuales del entorno tecnológico y operativo (Vial, 2019; Westerman et al., 2014).

Planteamiento del Problema

La Cooperativa Integral de Transporte Andino TransAndino, ubicada en la ciudad de Popayán, gestiona gran parte de su información administrativa mediante procesos manuales, apoyados en hojas de cálculo, archivos físicos y documentos dispersos. Esta forma de trabajo dificulta el control organizado de la información relacionada con vehículos, conductores, asociados, documentación legal y mantenimientos, generando desorden y dependencia de revisiones manuales constantes.

La ausencia de un sistema de información centralizado ocasiona problemas como duplicidad de registros, inconsistencias en los datos, pérdida de información relevante y demoras en la consulta de documentos. Estas situaciones afectan la eficiencia del personal administrativo y limitan la disponibilidad de información confiable para apoyar la toma de decisiones, aspecto fundamental en la gestión organizacional moderna (Laudon & Laudon, 2020).

Adicionalmente, el manejo manual de la documentación legal obligatoria, como el Seguro Obligatorio de Accidentes de Tránsito (SOAT), la revisión técnico-mecánica y las licencias de conducción, incrementa el riesgo de vencimientos no detectados oportunamente. Esto puede derivar en sanciones, multas o suspensión del servicio, afectando la operatividad y el cumplimiento normativo de la cooperativa (Ministerio de Transporte, 2023).

Estas dificultades también limitan la trazabilidad de la información y el acceso a registros históricos, necesarios para el seguimiento de mantenimientos, el control interno y la realización de auditorías administrativas. La falta de una estructura de datos organizada incrementa la probabilidad de errores y reduce la confiabilidad de los procesos administrativos (Elmasri & Navathe, 2015).

La relación entre las causas, el problema central y sus consecuencias se ilustra en la Figura 1, donde se presenta el árbol de problemas elaborado a partir del diagnóstico realizado en la cooperativa. Este esquema evidencia cómo la gestión manual de la información genera ineficiencias administrativas, riesgos operativos y dificultades en la toma de decisiones.

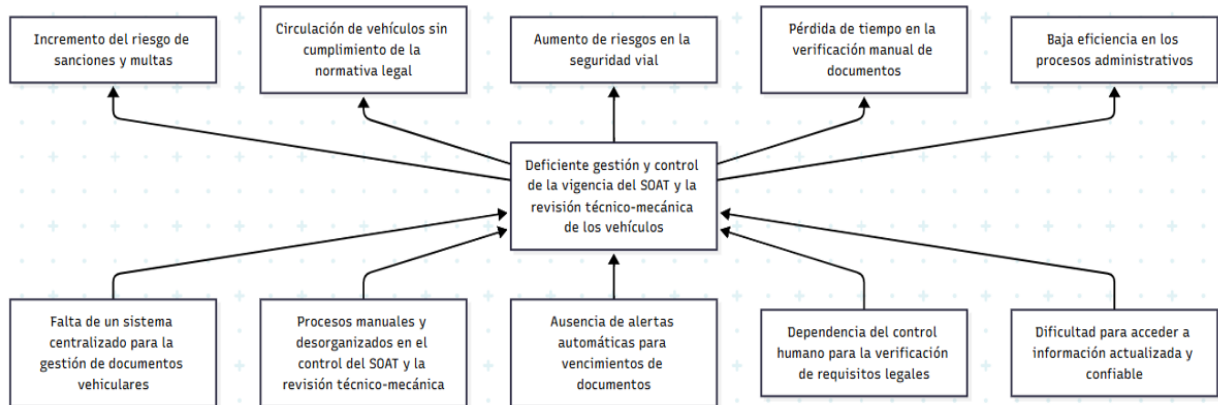


Figura 1 - *Árbol de Problemas* Fuente: *Propio*

En consecuencia, se identifica la necesidad de desarrollar un sistema de información que permita centralizar y organizar la gestión administrativa de la Cooperativa Integral de Transporte Andino TransAndino, con el fin de optimizar los procesos internos, mejorar el control documental y apoyar la toma de decisiones de manera oportuna y confiable.

Justificación

La implementación de un sistema de información para la Cooperativa Integral de Transporte Andino TransAndino se justifica por la necesidad de mejorar la gestión administrativa y superar las limitaciones generadas por el manejo manual de la información. Actualmente, el uso de hojas de cálculo y registros físicos dificulta el

control adecuado de los datos, aumenta la probabilidad de errores y retrasa los procesos administrativos relacionados con vehículos, conductores, asociados y documentación legal.

Desde el punto de vista organizacional, el sistema permitirá centralizar la información en una sola plataforma, facilitando el registro, la consulta y la actualización de los datos administrativos. Esto contribuirá a optimizar el tiempo del personal, mejorar el control documental y apoyar la toma de decisiones mediante información clara y confiable, aspectos fundamentales para una gestión administrativa eficiente (Laudon & Laudon, 2020).

En el ámbito tecnológico, el desarrollo del sistema se realiza mediante una aplicación web utilizando tecnologías modernas y ampliamente utilizadas en el desarrollo de software. Para el frontend se emplea React, apoyado por Vite, lo cual permite construir interfaces dinámicas, intuitivas y de fácil mantenimiento. Estas herramientas facilitan una mejor experiencia de usuario, aspecto importante considerando que los usuarios finales no cuentan con formación técnica avanzada.

Para el backend se utiliza Node.js junto con Express, permitiendo la creación de una API que gestiona la lógica del sistema y la comunicación con la base de datos. Esta arquitectura favorece un desarrollo organizado, escalable y acorde con las necesidades administrativas de la cooperativa. Como sistema gestor de base de datos se emplea MySQL, debido a su estabilidad y capacidad para manejar información estructurada, garantizando la integridad y coherencia de los datos (Pressman & Maxim, 2020).

Adicionalmente, el sistema facilita el control de la documentación legal obligatoria, como el SOAT, la revisión técnico-mecánica y las licencias de conducción, permitiendo registrar fechas de vencimiento y realizar un seguimiento oportuno. Esto reduce el riesgo de sanciones y contribuye al cumplimiento de la normatividad vigente en el sector transporte (Ministerio de Transporte, 2023).

En resumen, el sistema de información fue desarrollado con herramientas tecnológicas modernas y adecuadas a las necesidades reales de la organización. La elección de React, Vite, Node.js, Express y MySQL responde a criterios de eficiencia, facilidad de uso y escalabilidad, permitiendo una solución integral que optimiza la gestión administrativa, reduce riesgos operativos y prepara a la cooperativa para futuros procesos de modernización y crecimiento tecnológico.

Objetivos

Objetivo General

Desarrollar un sistema de información que facilite y optimice la gestión administrativa de la Cooperativa Integral de Transporte Andino TransAndino mediante el registro, actualización, consulta y control estructurado de la información de vehículos, asociados, conductores y mantenimientos.

Objetivos Específicos

1. Diseñar la estructura funcional y el modelo de datos que permitan una gestión clara y organizada de la información administrativa.

2. Implementar una aplicación web que centralice los procesos de registro, consulta, edición y control de datos relevantes para la cooperativa.
3. Validar el funcionamiento del sistema mediante pruebas funcionales y de usabilidad con usuarios finales para garantizar su eficacia y facilidad de uso.

Alcance

El presente proyecto contempla el diseño y desarrollo de un sistema de información orientado a mejorar la gestión administrativa de la Cooperativa Integral de Transporte Andino TransAndino. El sistema permitirá centralizar y organizar la información relacionada con los vehículos, conductores y asociados, facilitando su registro, consulta, actualización y control de manera estructurada.

Dentro del alcance funcional, el sistema incluirá módulos para la gestión de vehículos, asociados y conductores, permitiendo almacenar información básica y administrativa, así como la relación existente entre estos actores. Adicionalmente, se contará con un módulo para el registro y consulta de mantenimientos preventivos y correctivos, con el fin de llevar un control organizado del historial de cada vehículo.

Asimismo, se implementará un control de acceso mediante roles de usuario, donde el administrador contará con permisos completos para la gestión de la información, mientras que los usuarios con acceso limitado podrán consultar únicamente los datos que les correspondan. Esta diferenciación permitirá un uso más seguro y controlado del sistema.

El desarrollo del sistema se realizará como una aplicación web utilizando React y Vite para el frontend, Node.js y Express para el backend, y MySQL como sistema gestor de base de datos. El sistema estará orientado a su uso en un entorno local, a través de una red interna, sin depender de servicios externos para su funcionamiento básico.

El alcance del proyecto no incluye la integración con sistemas externos como plataformas de rastreo GPS, servicios de pago, aplicaciones móviles, ni módulos avanzados de analítica o reportes inteligentes. Estas funcionalidades se consideran posibles mejoras para fases futuras del sistema.

Finalmente, el proyecto contempla la realización de pruebas funcionales y de usabilidad con usuarios finales, con el propósito de verificar el correcto funcionamiento del sistema y su adecuación a las necesidades administrativas de la cooperativa.

Marco Referencial

El presente marco expone los fundamentos teóricos, conceptuales y antecedentes que sustentan el desarrollo del sistema de información propuesto para la Cooperativa Integral de Transporte Andino TransAndino. A través del análisis de conceptos relacionados con los sistemas de información, la gestión administrativa, la digitalización y el desarrollo de software, se establece un marco de referencia que respalda las decisiones técnicas y metodológicas adoptadas en el proyecto.

Asimismo, se presentan antecedentes relevantes que evidencian la importancia de la transformación digital en organizaciones del sector transporte, permitiendo contextualizar la necesidad de implementar soluciones tecnológicas orientadas a mejorar la organización, el control y la eficiencia de los procesos administrativos.

Cooperativa Integral de Andino TransAndino

La Cooperativa Integral de Transporte Andino TransAndino es una organización dedicada a la prestación del servicio de transporte público individual de pasajeros en la modalidad de taxi. La cooperativa cuenta con una sede principal ubicada en la ciudad de Popayán, departamento del Cauca, desde donde se coordinan los procesos administrativos y operativos. TransAndino gestiona y supervisa las rutas y zonas de operación autorizadas dentro del municipio, las cuales son cubiertas por una flota de taxis afiliados a la cooperativa. Su estructura empresarial se basa en la participación de asociados propietarios de los vehículos, conductores vinculados y personal administrativo, lo que permite garantizar un

servicio ordenado, seguro y conforme a la normativa vigente, contribuyendo a la movilidad urbana de la ciudad.



Figura 2 - Logo de la Cooperativa

Marco Teórico

El desarrollo de un sistema de información para la Cooperativa Integral de Transporte Andino TransAndino se fundamenta en conceptos teóricos relacionados con la gestión de la información, la administración organizacional y el uso de tecnologías digitales como apoyo a los procesos administrativos. El marco teórico permite contextualizar el proyecto y sustentar las decisiones técnicas adoptadas durante el diseño e implementación del sistema.

Los sistemas de información se definen como un conjunto de componentes que permiten recopilar, procesar, almacenar y distribuir información con el fin de apoyar las actividades operativas y administrativas de una organización. Su uso resulta esencial en entornos donde se maneja información de manera constante y estructurada, ya que contribuyen a mejorar la eficiencia, reducir errores y facilitar la toma de decisiones (Laudon & Laudon, 2020). En el contexto del transporte público, estos sistemas permiten organizar datos relacionados con vehículos, conductores,

asociados y documentación legal, garantizando mayor control y disponibilidad de la información.

La gestión administrativa hace referencia a los procesos de planificación, organización, dirección y control de los recursos de una organización. Cuando estos procesos se apoyan en herramientas tecnológicas, se mejora la trazabilidad de la información y se optimiza el uso del tiempo del personal administrativo. El manejo manual de datos, por el contrario, incrementa el riesgo de inconsistencias y pérdida de información, afectando el desempeño organizacional (Stair & Reynolds, 2017).

En este sentido, la digitalización se presenta como un proceso mediante el cual las organizaciones transforman actividades tradicionales en procesos apoyados por tecnologías de la información. La digitalización de la gestión administrativa permite centralizar la información, automatizar tareas repetitivas y conservar registros históricos de manera ordenada, lo cual resulta clave para el control interno y el cumplimiento normativo (Vial, 2019).

Por otra parte, el desarrollo de software orientado a sistemas de información requiere una correcta identificación de requerimientos y el uso de arquitecturas que permitan un funcionamiento estable y mantenible. Pressman y Maxim (2020) destacan que el diseño de aplicaciones debe responder a necesidades reales del usuario, priorizando la funcionalidad, la usabilidad y la confiabilidad del sistema. En este proyecto, estos principios orientan la construcción de una aplicación web que facilite el acceso y manejo de la información administrativa de la cooperativa.

En cuanto al manejo de datos, las bases de datos relacionales constituyen una solución ampliamente utilizada para almacenar información estructurada. Este modelo permite organizar los datos en tablas relacionadas, garantizando la integridad y coherencia de la información, lo cual es fundamental en sistemas administrativos que gestionan múltiples entidades interrelacionadas (Elmasri & Navathe, 2015).

Finalmente, la adopción de metodologías ágiles en el desarrollo de software favorece una construcción progresiva del sistema y una mejor adaptación a los cambios. Kanban, como metodología de gestión visual, permite organizar el trabajo, controlar el avance de las tareas y facilitar la entrega continua de funcionalidades, resultando adecuado para proyectos de desarrollo de software de alcance académico y organizacional (Anderson, 2010).

Marco Conceptual

El marco conceptual presenta los principales conceptos que fundamentan el desarrollo del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino. Su propósito es unificar el significado de los términos técnicos, administrativos y tecnológicos utilizados en el proyecto, facilitando la comprensión del alcance y funcionamiento de la solución propuesta.

Sistema de Información

Un sistema de información es un conjunto organizado de componentes que permiten recopilar, procesar, almacenar y distribuir información para apoyar las actividades operativas y administrativas de una organización. Estos sistemas

contribuyen a mejorar la eficiencia, reducir errores y facilitar la toma de decisiones mediante el uso adecuado de los datos (Laudon & Laudon, 2020). En el contexto del proyecto, el sistema de información permite administrar de manera estructurada la información relacionada con vehículos, conductores y asociados.

Digitalización

La digitalización consiste en la transformación de procesos manuales o basados en documentos físicos en procesos apoyados por tecnologías digitales. Este proceso permite optimizar el manejo de la información, mejorar la trazabilidad de los datos y reducir los errores humanos asociados al registro manual (Vial, 2019). En la cooperativa, la digitalización facilita el paso de hojas de cálculo dispersas a una plataforma centralizada.

Transformación Digital

La transformación digital va más allá de la digitalización, ya que implica la adopción de tecnologías digitales para mejorar los procesos organizacionales y la forma en que se gestiona la información. Este proceso permite a las organizaciones adaptarse a nuevas exigencias del entorno, mejorar su eficiencia operativa y fortalecer su competitividad (Westerman et al., 2014).

Gestión Administrativa

La gestión administrativa comprende las actividades de planificación, organización, dirección y control de los recursos de una organización. El uso de sistemas de información en estos procesos permite mejorar el control interno,

optimizar el tiempo del personal y garantizar un manejo más eficiente de la información administrativa (Koontz & Wehrich, 2012).

Base de Datos Relacional

Una base de datos relacional es un modelo de almacenamiento que organiza la información en tablas relacionadas entre sí mediante claves primarias y foráneas. Este tipo de bases de datos garantiza la integridad, coherencia y consistencia de los datos, siendo ampliamente utilizado en sistemas administrativos que manejan múltiples entidades relacionadas (Elmasri & Navathe, 2015). En este proyecto se emplea MySQL como sistema gestor de base de datos relacional.

Metodología Kanban

Kanban es una metodología ágil de gestión del trabajo que utiliza tableros visuales para organizar y controlar el flujo de tareas. Su aplicación permite mejorar la planificación, visualizar el avance del proyecto y facilitar la entrega continua de funcionalidades, siendo adecuada para proyectos de desarrollo de software de alcance académico y organizacional (Anderson, 2010).

Usuario Administrador

El usuario administrador es el responsable de gestionar el sistema de información y controlar el acceso a las funcionalidades disponibles. Este rol permite registrar, actualizar y consultar información, así como administrar usuarios y permisos, garantizando un uso adecuado y seguro del sistema (Pressman & Maxim, 2020).

Seguridad de la Información

La seguridad de la información comprende el conjunto de medidas destinadas a proteger los datos contra accesos no autorizados, alteraciones o pérdidas. En los sistemas de información administrativos, la seguridad se apoya en mecanismos como autenticación, control de roles y buenas prácticas en el manejo de la información (ISO, 2018).

Documentación Vehicular

La documentación vehicular corresponde a los documentos legales requeridos para la operación de vehículos, tales como el Seguro Obligatorio de Accidentes de Tránsito (SOAT) y la revisión técnico-mecánica. El control administrativo de estos documentos permite garantizar el cumplimiento normativo y la continuidad del servicio de transporte (Ministerio de Transporte, 2023).

Mantenimiento Preventivo

El mantenimiento preventivo se refiere a las actividades programadas que se realizan a los vehículos con el objetivo de prevenir fallas mecánicas y prolongar su vida útil. El registro sistemático de estos mantenimientos facilita la planificación operativa y mejora el control administrativo de la flota vehicular (Elmasri & Navathe, 2015).

Antecedentes

Con el fin de contextualizar el desarrollo del presente proyecto, se presentan a continuación algunos antecedentes relacionados con el uso de sistemas de información en procesos administrativos y en el sector transporte. Estos antecedentes corresponden a referencias generales de otros sistemas y plataformas en la tabla 1 se pueden observar, y se utilizan únicamente como marco de referencia, sin que impliquen un análisis detallado ni una evaluación técnica de las soluciones presentadas.

La digitalización de los procesos administrativos se ha convertido en una necesidad para organizaciones que manejan grandes volúmenes de información y requieren control permanente de sus operaciones. Diversos estudios evidencian que el uso de sistemas de información permite mejorar la organización de los datos, reducir errores asociados al manejo manual y fortalecer la toma de decisiones en las organizaciones (Laudon & Laudon, 2020).

En el contexto latinoamericano, la transformación digital ha sido identificada como un factor clave para mejorar la eficiencia y competitividad de las organizaciones, especialmente en sectores con alta carga administrativa como el transporte. Dini, Gligo y Patiño (2021) señalan que muchas empresas de la región han iniciado procesos de digitalización para superar limitaciones derivadas del uso de herramientas básicas, como hojas de cálculo y registros físicos, las cuales dificultan la trazabilidad y el control de la información.

En Colombia, el sector transporte ha experimentado avances en la adopción de tecnologías orientadas a la gestión de flotas, el control documental y la optimización de procesos operativos. El Ministerio de Transporte ha promovido el uso de plataformas tecnológicas para fortalecer el control administrativo y el cumplimiento normativo, destacando la importancia de contar con información organizada y actualizada para garantizar la seguridad y continuidad del servicio (Ministerio de Transporte, 2020).

A nivel empresarial, existen soluciones tecnológicas orientadas a la gestión de flotas y al control administrativo, que integran funcionalidades como el registro de vehículos, seguimiento de mantenimientos y control documental. No obstante, muchas de estas plataformas están diseñadas para grandes empresas o requieren inversiones económicas elevadas, lo que limita su adopción por parte de cooperativas y organizaciones de menor tamaño. La tabla 1 muestra la información de algunas plataformas.

Tabla 1 - plataformas tecnológicas para la gestión administrativa y operativa en el sector transporte

Plataforma	Funciones principales	Comparación con otras soluciones	Ventajas	Desventajas	Referencia
Hino Connect	Monitoreo en tiempo real, seguimiento de ubicación, análisis del comportamiento de conducción, estado del	Se enfoca principalmente en vehículos de la marca Hino, con integración directa a sus sistemas internos.	Información en tiempo real, mejora la seguridad vial, facilita la toma de decisiones operativas.	Uso limitado a vehículos compatibles con la marca, dependencia de conectividad constante.	Hino Motors, Ltd. (2019)

Plataforma	Funciones principales	Comparación con otras soluciones	Ventajas	Desventajas	Referencia
	vehículo y alertas de fallas.				
	Telemetría avanzada, control de consumo de combustible, mantenimiento preventivo, análisis del desempeño del conductor.	Ofrece mayor capacidad analítica y escalabilidad frente a soluciones más básicas.	Reduce costos operativos, alta personalización, soporte para grandes flotas.	Implementación más compleja, costos elevados para pequeñas empresas.	Geotab Inc. (2023)
	Rastreo satelital, administración vehicular, reportes básicos de flota y control de ubicación.	Solución más sencilla en comparación con plataformas de telemetría avanzada.	Fácil implementación, adecuada para pequeñas y medianas empresas, interfaz amigable.	Menor nivel de análisis avanzado y automatización.	GPSafe (2023)
	Seguimiento GPS, gestión de mantenimientos, control documental, reportes operativos y alertas.	Integra funciones administrativas y operativas en una sola plataforma.	Plataforma integral, adaptable a distintos tipos de flotas.	Curva de aprendizaje moderada, costos de licencia.	Trimble Inc. (2023)
	Monitoreo en tiempo real, mantenimiento predictivo, control documental, análisis de seguridad y reportes avanzados.	Presenta un mayor nivel de automatización y analítica frente a soluciones tradicionales.	Alta automatización, mejora la seguridad y el control normativo.	Requiere mayor inversión económica e infraestructura tecnológica.	Samsara Inc. (2023)

En este contexto, TransAndino representa un caso común dentro del sector, ya que su gestión administrativa se apoya principalmente en procesos manuales y herramientas ofimáticas. Esta situación genera dificultades en el control de la información, el acceso a registros históricos y el seguimiento adecuado de

documentos y mantenimientos, evidenciando la necesidad de una solución tecnológica adaptada a sus capacidades y necesidades reales.

Los antecedentes revisados permiten concluir que el desarrollo de un sistema de información a la medida constituye una alternativa viable para mejorar la gestión administrativa en cooperativas de transporte. La implementación de este tipo de soluciones contribuye a la organización de la información, la reducción de errores y el fortalecimiento de los procesos internos, alineándose con las tendencias actuales de digitalización y modernización tecnológica.

Tecnologías utilizadas en el desarrollo del sistema

Las tecnologías de desarrollo web constituyen un elemento fundamental en la construcción de sistemas de información orientados a la gestión administrativa, ya que permiten crear aplicaciones accesibles, escalables y adaptadas a las necesidades de las organizaciones. En este proyecto se seleccionaron tecnologías ampliamente utilizadas en el desarrollo de software, considerando criterios como facilidad de implementación, estabilidad y compatibilidad con aplicaciones administrativas.

React es una biblioteca de JavaScript orientada a la construcción de interfaces de usuario basadas en componentes reutilizables. Su enfoque declarativo facilita la creación de interfaces dinámicas y organizadas, permitiendo una mejor experiencia de usuario y un mantenimiento más sencillo del código, lo cual resulta adecuado para sistemas de información administrativos (Facebook Open Source, 2023).

Vite es una herramienta de apoyo al desarrollo frontend que proporciona un entorno de trabajo optimizado, reduciendo los tiempos de compilación y mejorando la productividad durante el desarrollo. Su uso permite una detección temprana de errores y un flujo de trabajo más ágil en la construcción de aplicaciones web modernas (Vite Team, 2023).

En el desarrollo del backend se utilizó Node.js, un entorno de ejecución de JavaScript del lado del servidor que permite manejar múltiples solicitudes de manera eficiente mediante un modelo de entrada y salida no bloqueante. Esta característica lo convierte en una alternativa adecuada para aplicaciones web que requieren un procesamiento rápido y concurrente de la información (OpenJS Foundation, 2023).

Para la implementación de la lógica del servidor y la comunicación entre el frontend y la base de datos se empleó Express, un framework para Node.js que facilita la creación de servicios web mediante una estructura ligera y modular. Express permite gestionar rutas, solicitudes y respuestas HTTP de forma organizada, contribuyendo a la construcción de aplicaciones web administrativas confiables (Express.js Foundation, 2023).

El almacenamiento de la información se realizó mediante MySQL, un sistema gestor de bases de datos relacional que organiza los datos en tablas relacionadas entre sí, garantizando la integridad y consistencia de la información. Este tipo de bases de datos es ampliamente utilizado en sistemas administrativos debido a su estabilidad, eficiencia y facilidad de integración con aplicaciones web (Oracle Corporation, 2023).

En conjunto, la adopción de estas tecnologías permite el desarrollo de un sistema de información estructurado, funcional y acorde con las necesidades administrativas de la cooperativa, apoyando la centralización de la información y la mejora de los procesos internos.

Marco metodológico de desarrollo del software

El desarrollo del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino se llevó a cabo bajo un enfoque metodológico propio del desarrollo de software, orientado a la construcción progresiva de una solución funcional y acorde con las necesidades reales de la organización. Este enfoque permitió estructurar el proceso desde el levantamiento de requerimientos hasta el diseño de la arquitectura funcional y del modelo de datos.

Metodología kanban

Para la gestión del proceso de desarrollo se adoptó la metodología Kanban, la cual facilita la organización y visualización de las actividades mediante un flujo de trabajo continuo como muestra la figura 3. Kanban permitió dividir el desarrollo del sistema en tareas específicas, priorizar actividades y realizar seguimiento al avance de cada fase, favoreciendo una ejecución ordenada y flexible del proyecto.

El tablero Kanban se estructuró en diferentes columnas, cada una asociada a una etapa específica del proceso de desarrollo. Esta organización permitió visualizar el avance de las tareas, identificar prioridades y mantener un control constante sobre el progreso del sistema.

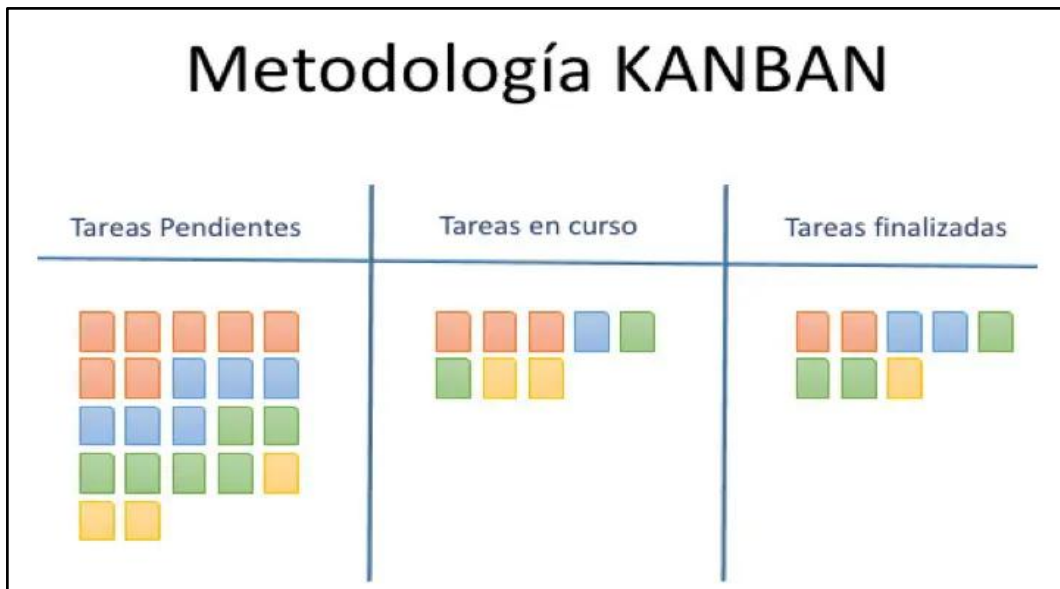


Figura 3 – Tablero Kanban y flujo de trabajo de la metodología Kanban.

Fuente: Calvo (s. f.).

En la columna “**Por hacer o pendientes**” se registraron todas las actividades planificadas que aún no habían sido iniciadas. Estas tareas corresponden principalmente a requerimientos funcionales identificados durante el levantamiento de información, ajustes solicitados por los usuarios y nuevas funcionalidades definidas a partir del análisis del contexto administrativo de la cooperativa. Esta columna permite mantener una visión clara del trabajo pendiente y apoyar la priorización de actividades.

La columna “**En progreso o en curso**” agrupó las tareas que se encontraban en desarrollo activo. En esta etapa se realizaban actividades relacionadas con el diseño, la implementación y los ajustes de las funcionalidades del sistema. El uso de esta columna facilitó la identificación de las tareas en curso y ayudó a evitar la sobrecarga de trabajo, permitiendo una distribución equilibrada de las actividades.

Posteriormente, las tareas avanzaban a la columna “**En pruebas**”, donde se verifica el correcto funcionamiento de las funcionalidades desarrolladas. En esta fase se realizaron pruebas funcionales y revisiones generales, con el fin de comprobar que cada componente cumpliera con los requerimientos establecidos antes de ser considerado como finalizado.

Finalmente, las tareas que cumplían satisfactoriamente con los criterios definidos se trasladaban a la columna “**Finalizado**”. Esta columna permite visualizar los avances logrados, validar el cumplimiento de los objetivos planteados y llevar un control claro de las actividades completadas durante el desarrollo del proyecto.

Cada actividad del proyecto fue representada mediante una tarjeta que avanzaba entre las columnas del tablero a medida que se desarrollaba. Este mecanismo permitió visualizar el estado real del proyecto, identificar posibles cuellos de botella y tomar decisiones oportunas para priorizar las actividades más relevantes.

En general, el uso de la metodología Kanban facilitó un desarrollo más organizado y flexible del sistema de información, permitiendo realizar ajustes continuos según las necesidades del proyecto y garantizando que cada avance fuera funcional, verificable y alineado con los requerimientos definidos

Asimismo, el marco metodológico adoptado permitió establecer una relación directa entre el análisis del contexto organizacional, la definición de requerimientos y el diseño de la arquitectura del sistema, garantizando coherencia entre los objetivos del proyecto y los componentes desarrollados.

Metodología ágil para el desarrollo del sistema

El desarrollo del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino se llevó a cabo bajo un enfoque de metodología ágil, orientado a la construcción progresiva de una solución funcional y alineada con las necesidades reales de la organización. Este enfoque permitió adaptarse a los cambios identificados durante el desarrollo y garantizar una mejora continua del sistema.

Dentro de las metodologías ágiles existentes, se adoptó Kanban como estrategia para la gestión y control del proceso de desarrollo de software. Kanban facilitó la organización de las actividades mediante un flujo de trabajo visual, permitiendo el seguimiento de tareas, la priorización de funcionalidades y el control del avance del proyecto.

La metodología se apoyó en el uso de un tablero Kanban, estructurado en diferentes columnas que representaban los estados del proceso de desarrollo como se abordó en el apartado anterior. Cada actividad del proyecto fue representada mediante una tarjeta que avanzaba entre las columnas a medida que se completaban las etapas correspondientes, lo que permitió visualizar el estado real del proyecto, identificar posibles cuellos de botella y realizar ajustes oportunos.

De manera complementaria, el desarrollo del sistema siguió un ciclo ágil iterativo e incremental, como se ilustra en la Figura 4, donde se representan las fases de planificación, diseño, desarrollo, pruebas, revisión y despliegue. Este ciclo permitió estructurar el trabajo en iteraciones cortas, asegurando que cada avance fuera funcional, evaluable y alineado con los requerimientos definidos.

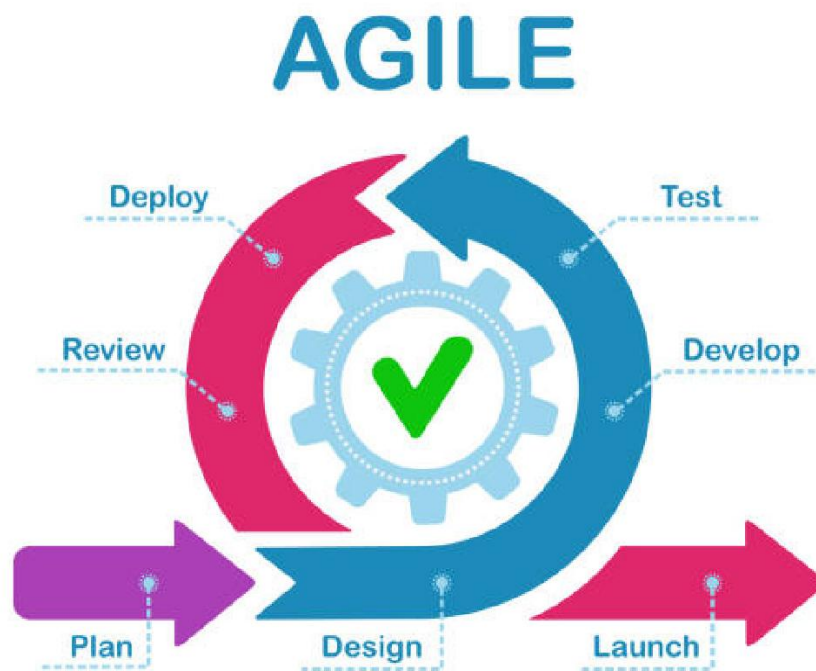


Figura 4 - Ciclo de desarrollo de la metodología ágil.

En la fase de Planificación (Plan) se identificaron y priorizaron los requerimientos funcionales y no funcionales del sistema, a partir del análisis del contexto administrativo de la cooperativa. Posteriormente, en la fase de Diseño (Design) se definieron la arquitectura funcional del sistema y el modelo de datos, estableciendo la estructura técnica necesaria para su implementación.

La fase de Desarrollo (Develop) correspondió a la construcción progresiva de los módulos del sistema, aplicando un enfoque incremental que permitió implementar funcionalidades completas en cada iteración. En la fase de Pruebas (Test) se verificó el correcto funcionamiento de las funcionalidades desarrolladas mediante pruebas funcionales, asegurando el cumplimiento de los requerimientos establecidos.

Durante la fase de Revisión (Review) se evaluaron los resultados obtenidos en cada iteración, identificando ajustes y mejoras a partir de la retroalimentación generada durante el desarrollo. Finalmente, la fase de Despliegue (Deploy / Launch) permitió poner a disposición las funcionalidades desarrolladas para su validación operativa, garantizando que los avances fueran funcionales y coherentes con los objetivos del proyecto.

En conjunto, la aplicación de una metodología ágil basada en Kanban, apoyada por un ciclo iterativo e incremental, permitió un desarrollo organizado, flexible y controlado del sistema de información. Este enfoque facilitó la entrega progresiva de funcionalidades, la incorporación de mejoras continuas y la alineación constante entre los requerimientos levantados y la solución tecnológica implementada.

Capítulo 1. Levantamiento de requisitos

El levantamiento de requerimientos del sistema de información se realizó mediante una reunión de trabajo con la gerencia de la Cooperativa Integral de Transporte Andino TransAndino, con el propósito de identificar las necesidades administrativas, definir el alcance del sistema y establecer las funcionalidades requeridas para apoyar los procesos internos de la organización. Este proceso permitió recopilar información directa de los actores responsables de la gestión administrativa, garantizando que la solución propuesta respondiera a las necesidades reales de la cooperativa.

Como resultado del levantamiento de requerimientos, se identificaron los requerimientos funcionales (RF), los requerimientos no funcionales (RNF) y los casos de uso, los cuales describen las acciones que el sistema debe realizar y las condiciones bajo las cuales debe operar. Adicionalmente, se generaron otros

artefactos de apoyo, tales como diagramas y descripciones funcionales, que sirvieron como base para el diseño de la arquitectura del sistema y el posterior desarrollo de la aplicación web.

Este proceso de levantamiento de requerimientos permitió establecer una base sólida para el diseño y construcción del sistema de información, asegurando la coherencia entre las expectativas de la cooperativa, los objetivos del proyecto y la solución tecnológica implementada.

Proceso de levantamiento de requisitos

El levantamiento de requerimientos del sistema de información se desarrolló a través de un proceso estructurado de indagación, análisis del estado actual AS-IS (se refiere al estado actual de un proceso, sistema o producto en el momento presente) y contextualización desde la digitalización del sector transporte, con el fin de identificar de manera clara las necesidades funcionales y no funcionales de la Cooperativa Integral de Transporte Andino TransAndino.

El proceso se llevó a cabo bajo un enfoque de entrevista semiestructurada la cual fue sincrónica a través de la plataforma Google Meet con la gerencia de la cooperativa, lo que permitió orientar la conversación a temas específicos sin limitar la posibilidad de profundizar en situaciones particulares expuestas por el gerente. El objetivo fue identificar las reglas de negocio, el flujo de trabajo actual y la gestión de la flota de vehículos. Este enfoque facilitó la identificación de necesidades explícitas y problemas operativos recurrentes relacionados con el manejo de la información.

Durante la entrevista se abordaron temas relacionados con el manejo actual de la información administrativa, las herramientas utilizadas para el registro de datos, las principales dificultades operativas y las necesidades identificadas frente a la implementación de un sistema de información. Algunas de las preguntas orientadoras fueron:

- *¿Cómo se gestionan actualmente los registros de vehículos, conductores y asociados?*
- *¿Qué problemas se presentan con mayor frecuencia en el manejo de la información*
- *¿Cómo se realiza el control de documentos obligatorios como el SOAT y la revisión técnico-mecánica?*
- *¿Qué funcionalidades considera necesarias para mejorar la gestión administrativa de la cooperativa?*

Durante la sesión, el gerente utilizó como artefacto de referencia el archivo de Excel que contiene información de la gestión actual como su principal medio transaccional y de consulta para la gestión administrativa de la flota vehicular de la cooperativa como muestra la figura 5.

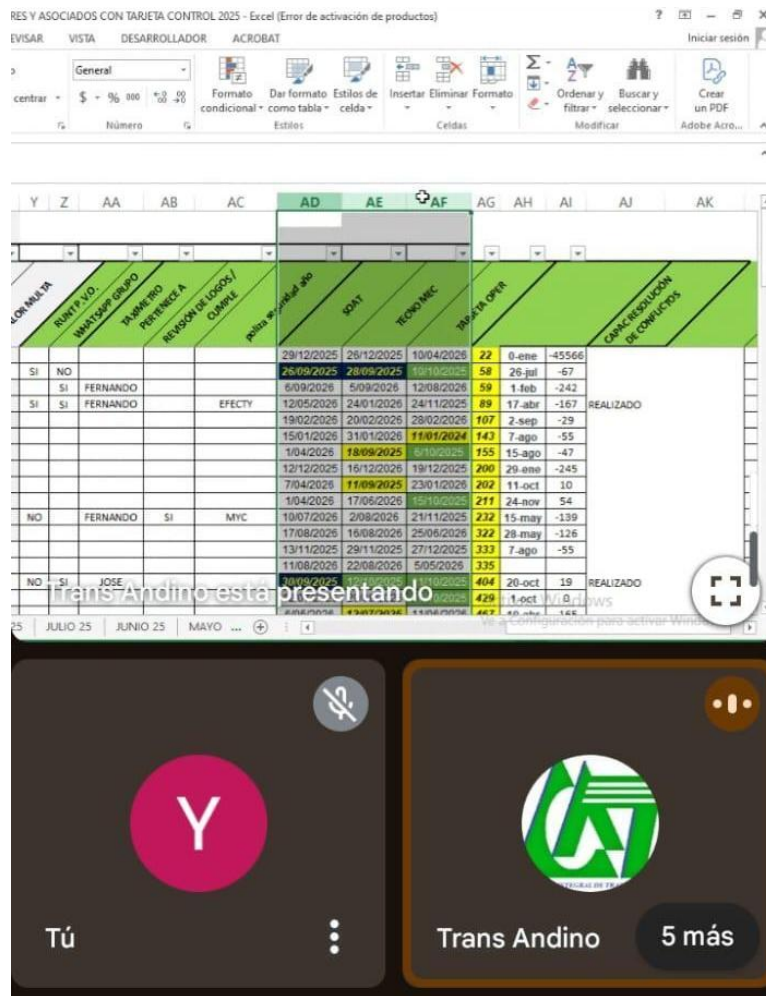


Figura 5 - Ciclo de desarrollo de la metodología ágil. Fuente - Popio

El archivo contiene múltiples campos relacionados con la gestión vehicular, entre los cuales se destacan: identificación del vehículo, estado de revisión, responsable del trámite, registros de multas, fechas de vencimiento de documentos obligatorios como SOAT y revisión técnico-mecánica, información asociada a pólizas, tarjetas de operación y observaciones administrativas. Asimismo, se evidencian columnas destinadas al control temporal de vencimientos, resaltadas mediante códigos de color, que permiten identificar de forma visual los documentos próximos a expirar o ya vencidos.

El uso de este archivo evidenció una alta dependencia del registro manual, la actualización constante por parte del personal administrativo y la ausencia de

validaciones automáticas, alertas sistemáticas y control de integridad de la información. Aunque el Excel permite llevar un control básico de la operación, su estructura limita la trazabilidad histórica, favorece la duplicidad de datos y dificulta la consulta eficiente de la información, especialmente cuando el volumen de registros aumenta.

El análisis de este artefacto permitió identificar de manera directa las necesidades del sistema de información propuesto, tales como la centralización de datos, la automatización del control de vencimientos, la gestión estructurada de documentos, la reducción de errores humanos y la mejora en la disponibilidad y confiabilidad de la información administrativa de la cooperativa.

Análisis de requisitos

El análisis de requerimientos constituye una etapa fundamental en el desarrollo del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino, ya que permite identificar y comprender de manera clara las necesidades reales de la organización. A través de este proceso se busca asegurar que la solución tecnológica responda efectivamente a los problemas existentes y contribuya a una gestión administrativa más organizada, eficiente y orientada a la transformación digital, alineándose con los objetivos institucionales de la cooperativa (Pressman & Maxim, 2020).

Este análisis no se limitó únicamente a la observación de los procesos actuales, sino que incorporó las experiencias y percepciones del personal administrativo de la cooperativa, quienes enfrentan de manera cotidiana dificultades en el registro, la

consulta y el control de la información. A partir de las reuniones realizadas, se identificaron los aspectos fundamentales que debía contemplar el sistema de información, permitiendo establecer una visión general del sistema, su propósito, alcance y principales funcionalidades, los cuales se describen a continuación

- **Propósito del sistema:** El sistema de información tiene como propósito optimizar la gestión administrativa de la cooperativa mediante la centralización y control de la información relacionada con la flota vehicular, conductores, asociados, documentación obligatoria y registros operativos. Actualmente, estos procesos se gestionan principalmente en un archivo Excel, lo que genera riesgos de duplicidad, inconsistencias y dificultades para el seguimiento de vencimientos. Por ello, el sistema busca mejorar la disponibilidad, confiabilidad y trazabilidad de los datos, incorporando funcionalidades de registro, consulta, actualización y alertas preventivas para apoyar la toma de decisiones y el cumplimiento normativo.
- **Contexto del sistema:** El sistema de información se desarrolla en el contexto administrativo de la Cooperativa Integral de Transporte Andino TransAndino, organización dedicada a la gestión de una flota vehicular de transporte público. La cooperativa administra información relacionada con vehículos, conductores, asociados, documentación legal y mantenimientos, la cual es actualmente gestionada de forma manual mediante hojas de cálculo y registros dispersos.

El sistema operará como una aplicación web de uso interno, orientada a apoyar las actividades del personal administrativo y del gerente de la cooperativa. Su implementación se realiza en un entorno local, considerando las limitaciones

tecnológicas, presupuestales y de capacitación del personal, y se integra a los procesos existentes sin alterar la estructura organizacional actual.

- **Alcance del sistema:** El sistema de información abarca la gestión administrativa de vehículos, conductores y asociados de la Cooperativa de Transporte Integral Andino TransAndino, incluyendo el control de documentación obligatoria y mantenimientos. El sistema permite el registro, consulta y actualización de la información, sin contemplar en su versión inicial integraciones con sistemas externos ni funcionalidades avanzadas de analítica.

- **Personal Involucrado:** El desarrollo del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino implica la participación de diferentes actores que, desde sus roles y conocimientos, contribuyen al éxito del proyecto. Cada persona involucrada cumple una función esencial, ya sea en la recolección de información, la toma de decisiones, el desarrollo del software o la validación de los resultados. Entender quiénes participan y cuál es su aporte permite que el proceso sea organizado, colaborativo y orientado a mejorar la calidad de los procesos administrativos de la cooperativa.

En la tabla se describen los perfiles y responsabilidades de cada uno de los participantes:

Tabla 2 - Personal Involucrado Fuente: Propio

Persona	Rol	Responsabilidades
Desarrollador / Equipo de Desarrollo de Software	Equipo técnico encargado del desarrollo del sistema	Analizar los requerimientos identificados durante la investigación; diseñar la arquitectura del sistema y la base de datos; desarrollar los módulos del software siguiendo metodologías ágiles; realizar pruebas técnicas y corrección de errores; garantizar que la plataforma sea estable e intuitiva; documentar el proceso técnico y funcional; mantener comunicación constante con el personal administrativo para adaptar el sistema a las necesidades reales de la cooperativa.
Usuarios finales (Asociados y Conductores)	Usuarios clave del sistema de información	Proveer la información requerida para el registro en el sistema; validar la veracidad de los datos almacenados; participar en pruebas de uso cuando sea necesario; beneficiarse indirectamente del sistema mediante la mejora en procesos administrativos como actualización de documentos, mantenimientos y trámites internos.
Asesor del proyecto	Tutor académico y supervisor metodológico	Orientar metodológicamente el desarrollo del proyecto; validar el marco teórico, el análisis y la estructura del documento; asegurar el cumplimiento de los criterios académicos establecidos; brindar retroalimentación continua para mejorar la calidad técnica, investigativa y formal del trabajo.

- Acrónimos y abreviaturas

Tabla 3 - Acrónimos y Abreviaturas Fuente: Propio

Acrónimo / Abreviatura	Descripción
IEEE	Institute of Electrical and Electronics Engineers
IEEE 29148	Estándar IEEE para la Ingeniería de Requisitos de Sistemas y Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
SOAT	Seguro Obligatorio de Accidentes de Tránsito
TI	Tecnologías de la Información
BD	Base de Datos
MySQL	Sistema gestor de bases de datos relacional
API	Application Programming Interface
UI	User Interface (Interfaz de Usuario)
UX	User Experience (Experiencia de Usuario)
CRUD	Create, Read, Update, Delete
LAN	Local Area Network (Red de Área Local)

Funcionalidades del sistema identificados

El sistema de información diseñado para la Cooperativa Integral de Transporte Andino Transandino cuenta con un conjunto de funcionalidades pensadas para resolver las necesidades reales del personal administrativo y mejorar significativamente el manejo de la información. Estas funcionalidades no solo automatizan tareas que hoy se hacen de manera manual, sino que también buscan facilitar el trabajo diario, reducir errores, agilizar consultas y proporcionar un entorno más organizado y seguro.

Cada funcionalidad fue definida a partir del análisis de los procesos actuales, conversaciones con el personal y la observación de las tareas que generan mayor carga operativa. Por ello, el sistema se construyó pensando en ser práctico, intuitivo y útil para quienes lo utilizarán.

En la tabla 4, se presentan las principales funcionalidades del sistema:

Tabla 4 - Funcionalidades del Sistema de Información **Fuente:** Propio

ID	Nombre	Descripción	Flujo	Resultado (Positivo / Negativo)
F-01	Gestión de Vehículos	Permite registrar, consultar y actualizar de manera sistemática la información de los vehículos afiliados a la cooperativa. Incluye datos técnicos del vehículo, información del asociado y control de documentos obligatorios como SOAT, revisión técnico-mecánica y tarjeta de propiedad, con fechas de emisión y vencimiento.	El usuario accede al módulo de vehículos, registra o consulta la información requerida y el sistema almacena y valida los datos, generando alertas cuando un documento está próximo a vencer.	Positivo: Centralización de la información, reducción de errores, prevención de sanciones por documentos vencidos. Negativo: Requiere ingreso correcto y constante de la información para mantener datos actualizados.

ID	Nombre	Descripción	Flujo	Resultado (Positivo / Negativo)
F-02	Gestión de Conductores	Facilita la administración de la información personal de los conductores, permitiendo su registro, consulta y control dentro de una base de datos. Incluye la vinculación del conductor a un vehículo específico y a una jornada de trabajo determinada.	El usuario registra al conductor, asigna el vehículo y la jornada correspondiente; el sistema guarda y relaciona la información con el vehículo asignado.	Positivo: Mayor control operativo y claridad en la asignación de conductores. Negativo: Dependencia de la actualización oportuna cuando se presentan cambios de conductor.
F-03	Gestión de Asociados	Permite administrar la información de los propietarios o socios de los vehículos, registrando, editando y consultando sus datos, así como su relación con uno o varios vehículos afiliados.	El usuario registra o consulta al asociado y el sistema vincula automáticamente sus datos con los vehículos correspondientes.	Positivo: Mejora la gestión de relaciones internas y agiliza procesos administrativos. Negativo: Requiere validación constante para evitar inconsistencias en la relación asociado-vehículo.
F-04	Módulo de Mantenimientos	Permite registrar y consultar los mantenimientos preventivos y correctivos realizados a cada vehículo, incluyendo fecha, tipo de mantenimiento, descripción del trabajo y próxima fecha sugerida.	El usuario selecciona un vehículo, registra el mantenimiento y el sistema guarda el historial asociado.	Positivo: Facilita el seguimiento del estado mecánico y contribuye a prolongar la vida útil de los vehículos. Negativo: Depende del registro oportuno de cada intervención realizada.
F-05	Gestión de Usuarios y Accesos	Controla el acceso al sistema mediante credenciales y roles definidos, diferenciando usuarios administradores con permisos completos y asociados con acceso solo de visualización. Incluye un registro básico de actividad.	El usuario inicia sesión, el sistema valida credenciales y asigna permisos según el rol configurado.	Positivo: Mayor seguridad, control de la información y trazabilidad de acciones. Negativo: Requiere una correcta administración de usuarios para evitar accesos indebidos.

ID	Nombre	Descripción	Flujo	Resultado (Positivo / Negativo)
F-06	Interfaz Amigable e Intuitiva	Presenta una interfaz con menús organizados, formularios claros, botones identificables y mensajes de confirmación o error, facilitando la navegación y el uso del sistema.	El usuario interactúa con el sistema mediante pantallas claras que guían cada acción hasta su finalización.	Positivo: Reduce la curva de aprendizaje y mejora la experiencia del usuario. Negativo: Puede requerir ajustes visuales según retroalimentación de usuarios finales.

Requerimientos Funcionales

Los requerimientos funcionales (RF) describen las acciones y funcionalidades que el sistema de información debe permitir realizar para apoyar la gestión administrativa de la cooperativa. Estos requerimientos fueron identificados a partir del análisis de las funcionalidades existentes en los procesos actuales y de las necesidades detectadas durante el levantamiento de información. Para su definición se tuvo en cuenta la revisión de los documentos utilizados actualmente, la observación de las tareas administrativas diarias y los conversatorios realizados con el personal involucrado en la gestión de la información. Este proceso permitió establecer un conjunto de funcionalidades orientadas a mejorar el registro, la consulta, el control y la organización de la información administrativa. La tabla 5 muestra los RF establecidos.

Tabla 5 - Requerimientos Funcionales Fuente: Propio

ID	Nombre del Requerimiento	Descripción	Prioridad
RF-01	Registro de Vehículos	El sistema debe permitir registrar la información básica de cada vehículo afiliado a la cooperativa.	Alta
RF-02	Actualización de Información Vehicular	El sistema debe permitir actualizar la información registrada de un vehículo sin perder el historial asociado.	Alta

ID	Nombre del Requerimiento	Descripción	Prioridad
RF-03	Consulta de Información Vehicular	El sistema debe permitir consultar la información detallada de un vehículo.	Alta
RF-04	Registro de Documentos Vehiculares	El sistema debe permitir registrar documentos obligatorios del vehículo, incluyendo SOAT, revisión técnico-mecánica y tarjeta de operación.	Alta
RF-05	Control de Fechas de Vencimiento	El sistema debe permitir registrar fechas de emisión y vencimiento de los documentos vehiculares.	Alta
RF-06	Registro de Mantenimientos	El sistema debe permitir registrar mantenimientos preventivos y correctivos de los vehículos.	Media
RF-07	Consulta de Historial de Mantenimientos	El sistema debe permitir consultar el historial de mantenimientos por vehículo.	Media
RF-08	Registro de Conductores	El sistema debe permitir registrar información de los conductores vinculados a la cooperativa.	Alta
RF-09	Asignación de Conductores a Vehículos	El sistema debe permitir asignar conductores a vehículos y jornadas de trabajo.	Alta
RF-10	Registro de Asociados	El sistema debe permitir registrar la información de los asociados propietarios de vehículos.	Media
RF-11	Vinculación de Asociados con Vehículos	El sistema debe permitir vincular uno o varios vehículos a un asociado.	Media
RF-12	Búsqueda y Filtrado de Información	El sistema debe permitir buscar información mediante filtros por vehículo, fecha, documento o estado.	Alta
RF-13	Autenticación de Usuarios	El sistema debe permitir el inicio de sesión de los usuarios registrados.	Alta
RF-11	Gestión de Roles y Permisos	El sistema debe permitir asignar roles y restringir funcionalidades según el perfil del usuario.	Alta

Requerimientos No Funcionales

Los requerimientos no funcionales describen las características de calidad que debe cumplir el sistema para asegurar un funcionamiento adecuado, seguro y eficiente, tales como usabilidad, rendimiento y escalabilidad. Estos aspectos no definen lo que el sistema hace, sino cómo debe hacerlo para garantizar una experiencia óptima para los usuarios finales.

Según Pressman y Maxim (2020), los requerimientos no funcionales establecen criterios esenciales para evaluar la calidad del software y su aceptación. De igual manera, Sommerville (2016) indica que estos requisitos influyen directamente en la arquitectura del sistema y en su capacidad de adaptación a futuras necesidades, lo cual resulta fundamental para el contexto administrativo de la Cooperativa Integral de Transporte Andino TransAndino.

Tabla 6 - *Requerimientos No Funcionales Fuente: Propio*

ID	Nombre del requerimiento	Descripción	Prioridad	Criterio de aceptación
RNF-1	Usabilidad del sistema	La interfaz debe contar con menús identificables, mensajes de error, y botones con etiquetas visibles, permitiendo que un usuario complete procesos básicos como buscar, consultar, crear, actualizar o deshabilitar.	Media	Al menos el 85% de los usuarios administrativos completa las tareas sin ayuda y con menos de 2 errores por proceso
RNF-2	Rendimiento del sistema	El sistema debe procesar consultas de manera "rápida" que no haga esperar al usuario además de cargar la información en el menor tiempo posible	Media	Las pruebas de rendimiento muestran tiempos de consultas eficientes para los usuarios
RNF-3	Escalabilidad del sistema	La arquitectura del sistema debe permitir la integración de nuevos módulos sin modificar los existentes así como soportar el incremento	Media	Es posible agregar un nuevo módulo sin que afecte el funcionamiento

		progresivo de usuarios y datos		normal del sistema
--	--	--------------------------------	--	--------------------

Restricciones

Las restricciones del sistema describen las limitaciones técnicas, operativas y organizacionales que condicionan su diseño, desarrollo e implementación. Estas restricciones fueron identificadas a partir del análisis del contexto institucional de TransAndino, los recursos disponibles y las características del personal involucrado, y deben ser consideradas para garantizar la viabilidad y correcta adopción del sistema de información.

- **Recursos tecnológicos limitados:** El desarrollo del sistema se realizó considerando las limitaciones de infraestructura y herramientas disponibles en la cooperativa. Por esta razón, se optó por tecnologías de código abierto y de amplia adopción. El frontend fue desarrollado con React y Vite, el backend con Node.js y Express, y la base de datos con MySQL, permitiendo construir un sistema funcional, mantenible y escalable sin requerir inversiones adicionales en licencias o infraestructura especializada.
- **Dependencia de conexión a Internet:** El sistema opera bajo una arquitectura cliente–servidor en entorno local (LAN), lo que minimiza la dependencia de una conexión permanente a Internet. Su funcionamiento cotidiano no requiere conectividad externa; sin embargo, el acceso a Internet puede ser necesario de forma ocasional para tareas como actualizaciones, respaldos, soporte técnico o futuras integraciones.

- **Limitaciones en la capacitación del personal:** El personal administrativo presenta niveles básicos de formación tecnológica, lo que implica una curva de aprendizaje durante la adopción del sistema. Esta restricción exige el diseño de interfaces intuitivas, procesos simplificados y la realización de capacitaciones básicas para facilitar su uso adecuado.
- **Tiempo reducido para la implementación:** El proyecto debe ejecutarse dentro de un cronograma académico y administrativo definido, lo que limita la incorporación de funcionalidades avanzadas en la fase inicial. En consecuencia, se priorizan las funciones esenciales del sistema, dejando mejoras y ampliaciones para etapas posteriores.
- **Seguridad dependiente de buenas prácticas:** Aunque el sistema incorpora mecanismos básicos de seguridad, la protección de la información depende en gran medida del uso adecuado por parte del personal. El manejo incorrecto de credenciales o accesos no autorizados puede comprometer la seguridad, por lo que se requiere promover buenas prácticas y sensibilizar a los usuarios sobre el cuidado de la información.

Casos de Uso

Los casos de uso describen de manera estructurada las interacciones entre los actores y el sistema, permitiendo comprender el comportamiento funcional del sistema propuesto. En la **Tabla 7** se presenta un resumen de los casos de uso

identificados, mientras que su especificación detallada, incluyendo flujos principales y alternativos, se encuentra desarrollada en el anexo correspondiente.

Tabla 7 - Resumen de Casos de Uso – Sistema de Información Cooperativa TransAndino Fuente: Propio.

ID	Nombre del Caso de Uso	Actor	Descripción	Resultado
CU-01	Registrar vehículo	Administrador	Permite registrar un nuevo vehículo ingresando su información técnica y documentación obligatoria.	Vehículo almacenado correctamente en el sistema.
CU-02	Consultar y actualizar vehículo	Administrador	Permite consultar y modificar la información registrada de un vehículo existente.	Información del vehículo actualizada.
CU-03	Registrar conductor	Administrador	Permite registrar un nuevo conductor, incluyendo datos personales, documentos y vinculación a un vehículo.	Conductor registrado y asociado correctamente.
CU-04	Consultar conductores	Administrador	Permite visualizar la información de conductores registrados en el sistema.	Información mostrada de forma clara y organizada.
CU-05	Gestionar asociados	Administrador	Permite registrar, editar y consultar información de los asociados y su relación con vehículos.	Asociados correctamente administrados en el sistema.
CU-06	Registrar mantenimiento	Administrador	Permite registrar mantenimientos preventivos o correctivos realizados a un vehículo.	Mantenimiento almacenado en el historial del vehículo.
CU-07	Consultar historial de mantenimientos	Administrador	Permite consultar el historial completo de mantenimientos de un vehículo específico.	Historial accesible y ordenado.

ID	Nombre del Caso de Uso	Actor	Descripción	Resultado
CU-08	Gestionar documentación	Administrador	Permite cargar, consultar y controlar documentos obligatorios de vehículos y conductores.	Documentación controlada y centralizada.
CU-09	Generar alertas de vencimiento	Sistema	Genera alertas automáticas cuando un documento está próximo a vencer.	Usuario notificado oportunamente.
CU-10	Inicio de sesión	Administrador / Asociado	Permite el acceso al sistema mediante credenciales y roles definidos.	Acceso seguro al sistema.
CU-11	Consultar información (asociado)	Asociado	Permite consultar información relacionada con sus vehículos sin modificar datos.	Consulta en modo solo lectura.

La especificación detallada de los casos de uso del sistema se presenta en el **Anexo A**, donde se describen las interacciones entre los actores y el sistema, incluyendo los flujos principales y alternativos, las pre y postcondiciones, así como las reglas de negocio asociadas. Estos casos de uso permiten complementar la definición de los requerimientos funcionales y facilitan la trazabilidad, validación y comprensión del comportamiento del sistema propuesto.

Capítulo 2 - Implementación de la Aplicación Web

Arquitectura del Sistema

El diseño de la arquitectura del sistema permite definir cómo se organizarán e integrarán los componentes tecnológicos que conforman la solución propuesta para la Cooperativa de Transporte TransAndino. A través de esta estructura se establecen los criterios que orientan la interacción entre la interfaz de usuario, la lógica de negocio y el almacenamiento de la información, asegurando que el sistema responda de manera eficiente a los requerimientos administrativos identificados y facilite su mantenimiento, seguridad y evolución futura.

La Figura 6 presenta el diagrama de flujo del sistema, el cual ilustra la interacción entre las distintas capas de la arquitectura a través de un ciclo de comunicación basado en el protocolo HTTP. El proceso se inicia en la **Capa de Presentación (React) – Front-End**, desde donde se envían las solicitudes al servidor utilizando un interceptor de **Axios (una de las bibliotecas de JavaScript más utilizadas para realizar solicitudes HTTP)**, encargado de adjuntar el token de seguridad en cada petición.

Posteriormente, la solicitud es procesada en la **Capa de Lógica de Negocio - Controllers (Node.js)**, donde los *middlewares* se encargan de validar los permisos y la autenticidad del acceso. Una vez superadas estas verificaciones, el controlador ejecuta las consultas SQL necesarias sobre la base de datos.

Finalmente, la **Capa de Datos (MySQL) - Models** retorna la información solicitada, completando el ciclo de comunicación con la actualización de la interfaz de usuario, permitiendo que el usuario visualice el resultado de la acción realizada.

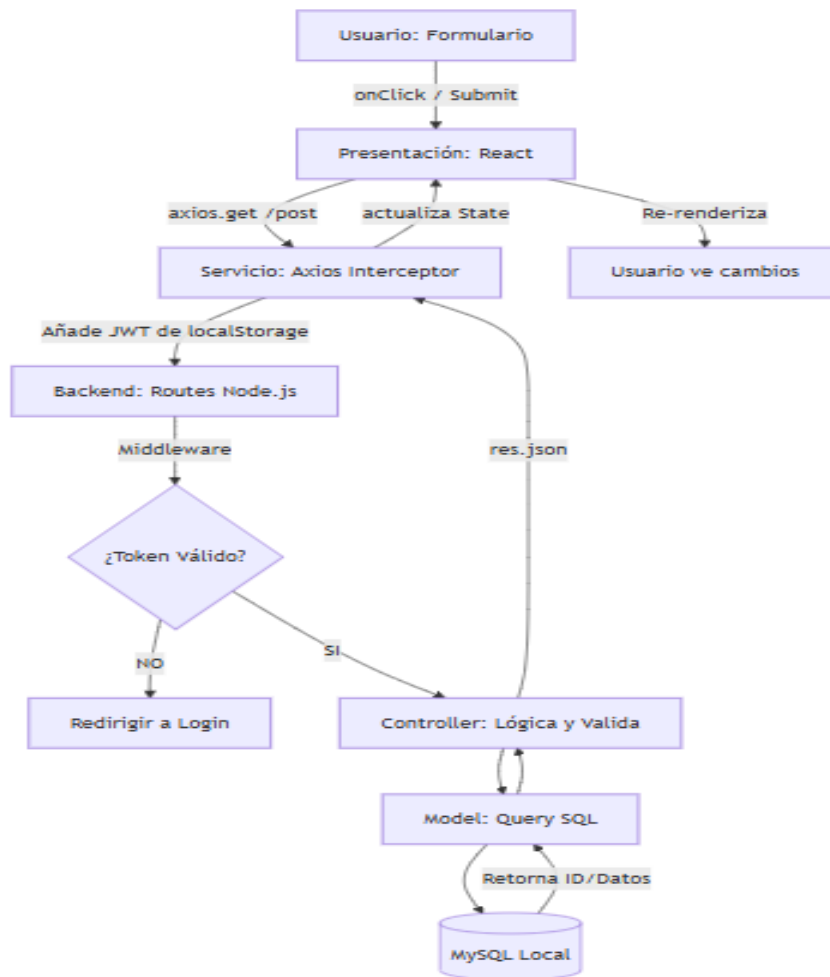


Figura 6 - Diagrama de flujo de la arquitectura del sistema. Fuente - Propio

Organización de la arquitectura

El sistema TransAndino implementa una arquitectura desacoplada **Cliente-Servidor**, donde el Front-End hecho en React gestiona la experiencia de usuario y el Backend Node.js/Express provee los servicios y la lógica de negocio. La separación en paquetes permite una alta cohesión interna y bajo acoplamiento entre módulos, facilitando la mantenibilidad, escalabilidad y evolución del sistema.

Organización del Frontend

El Frontend es la capa de presentación corresponde a la interfaz mediante la cual los usuarios interactúan con el sistema. Esta capa se encarga de mostrar la información y recibir los datos ingresados por el personal administrativo, permitiendo la ejecución de las funciones del sistema de manera clara y ordenada.

El diseño de la interfaz se orienta a la usabilidad, priorizando la claridad visual, la organización de los menús y la facilidad de navegación. Esto permite reducir errores en el ingreso de información y mejorar la experiencia del usuario, considerando que los usuarios finales no necesariamente cuentan con formación técnica especializada.

Su estructura sigue el enfoque del patrón **Atomic Design**, como muestra la figura 7 lo que favorece la reutilización, escalabilidad y mantenibilidad de los componentes de la interfaz.

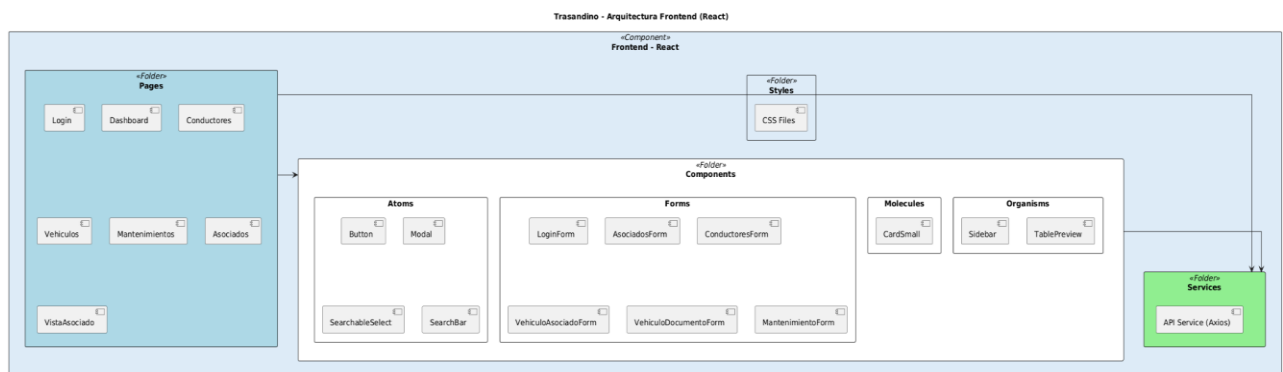


Figura 7 - Diagrama de paquetes front-end. Fuente - Propio

Cada uno de los paquetes del frontend se presentan a continuación:

El paquete *Pages* agrupa las vistas principales del sistema, las cuales representan pantallas completas accesibles por el usuario final: Dashboard, Login, Asociados, Conductores, Vehículos, Mantenimientos y Vista Asociado.

El paquete *Components* implementa el patrón Atomic Design, organizado en cuatro niveles:

Atoms que son los componentes básicos e indivisibles de la interfaz

Forms son los componentes especializados para la captura y validación de información.

Molecules Combinación de átomos con una funcionalidad concreta:

Organisms, son los componentes más complejos que integran múltiples moléculas y

átomos, Styles: Contiene los archivos CSS responsables de la presentación visual y

la coherencia gráfica del sistema. El paquete *Services* contiene el archivo *api.js*,

encargado de gestionar la comunicación con el backend mediante peticiones HTTP

REST, desacoplando la lógica de consumo de servicios del resto de la interfaz.

Backend – Node.js / Express

El backend está estructurado bajo una arquitectura en capas, una de la más importantes es la capa de lógica de negocio constituye el núcleo funcional del sistema

que está representada por el paquete Controller, ya que en ella se implementan las reglas, validaciones y procesos que controlan el comportamiento de la aplicación.

Esta capa se encarga de procesar la información proveniente del frontend, verifica su consistencia y aplicar las normas administrativas definidas por la cooperativa.

Sus funciones principales incluyen la validación que comprueba que los datos obtenidos sean precisos y se ajusten a las reglas de la cooperativa, seguridad por

medio de Middlewares para verificar que el JWT (son las sigl(de JSON Web Token, un estándar abierto (RFC 7519) utilizado para transmitir información de manera

segura entre dos partes) es auténtico y el usuario cuenta con los permisos necesarios

(Administrador o Asociado) y por último el control que procesa la lógica que organiza las interacciones entre el Frontend y la Base de Datos que determina la respuesta de éxito o error que se devuelve al usuario. La figura 8, muestra cada paquete que cumple una responsabilidad específica,

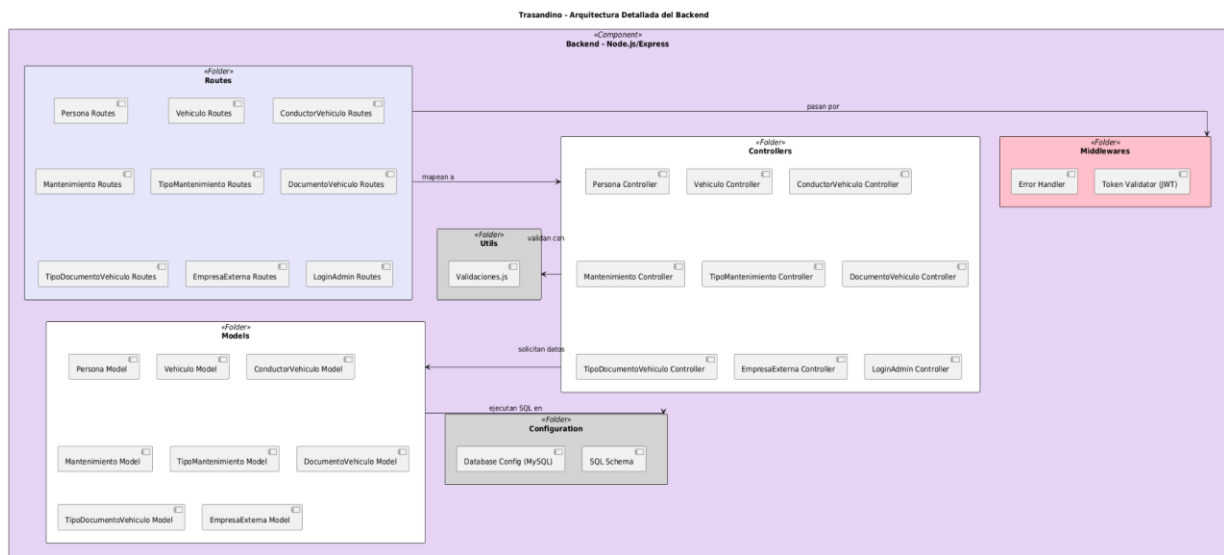


Figura 8 - Diagrama de paquetes Backend Fuente - Propio

La capa de datos es responsable del almacenamiento, organización y recuperación de la información gestionada por el sistema. Por medio del paquete **models**, en esta capa se almacenan los registros correspondientes a vehículos, conductores, asociados, documentación, mantenimientos y usuarios del sistema.

Para mantener la consistencia de la arquitectura, se utilizan otros paquetes tales como **Route**, el cual define las rutas del sistema y vincula las peticiones HTTP para mantener la comunicación entre el backend con los controladores y las rutas del frontend; **Utils**, que incluye funcionalidades transversales; **Middlewares**, cuya función es gestionar las responsabilidades transversales al flujo de ejecución; y, finalmente, **Configuration**, que se encarga de centralizar la configuración del sistema.

Diagrama Entidad Relación

Los diagramas UML que permiten modelar la estructura y el comportamiento del sistema propuesto. Se incluyen el diagrama de clases, el diagrama de casos de uso y el diagrama de secuencia, los cuales facilitan la comprensión de las entidades del sistema, las interacciones entre los actores y el flujo de ejecución de las funcionalidades principales.

El **modelado de datos** permite definir la estructura sobre la cual se almacenará y gestionará la información del sistema, estableciendo las entidades, relaciones y restricciones necesarias para asegurar un manejo confiable de los datos administrativos de la Cooperativa Integral de Transporte Andino TransAndino. A partir de este diseño se garantiza la coherencia entre los registros de vehículos, conductores, asociados y mantenimientos, facilitando la trazabilidad de la información y respaldando las operaciones del sistema sin inconsistencias ni duplicidad de datos.

Este modelado se deriva del análisis de los procesos administrativos actuales de la cooperativa, en los cuales se evidenció la necesidad de organizar información relacionada con vehículos, conductores, asociados, documentación y mantenimientos. La figura 9 muestra el diagrama entidad relación.

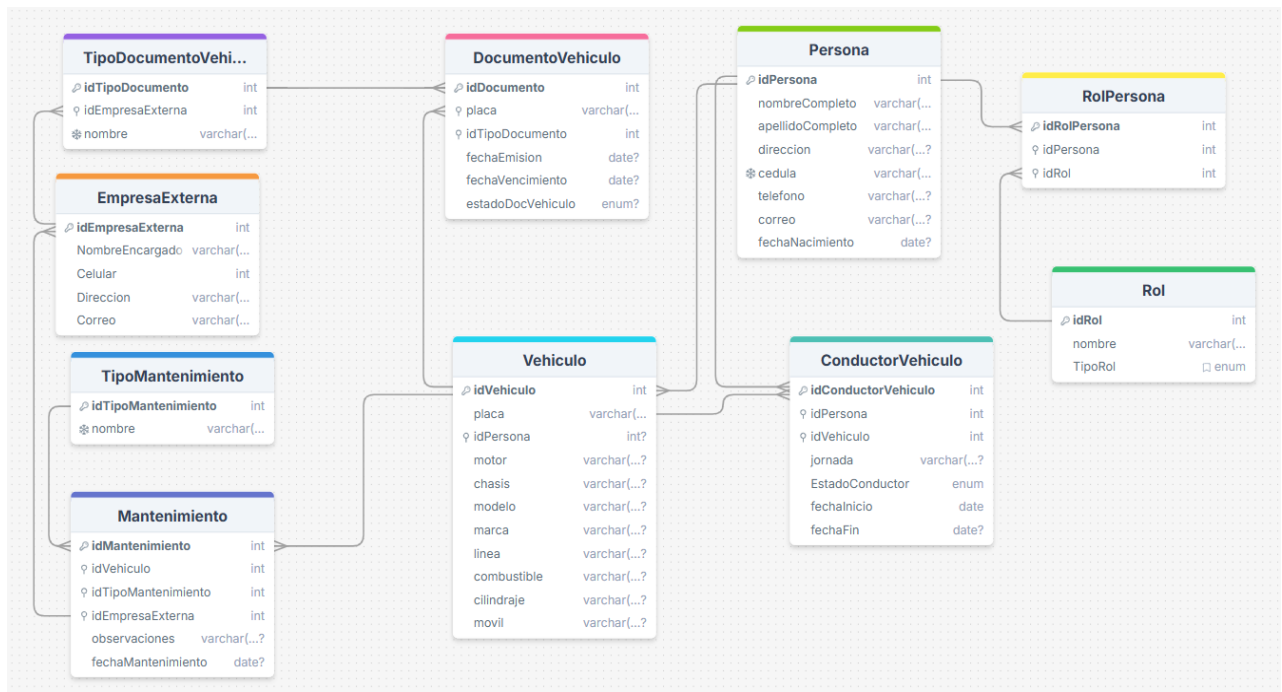


Figura 9 - Diagrama entidad relación Fuente - Propio

Las entidades principales definidas dentro del modelo de datos incluyen vehículos, conductores, asociados, documentos, mantenimientos. Cada entidad contiene atributos específicos que describen sus características y permiten su correcta identificación dentro del sistema. Por ejemplo, la entidad vehículo almacena información como placa, tipo y estado, mientras que la entidad conductora contiene datos personales y documentales relevantes para su control administrativo.

Las relaciones entre las entidades reflejan el funcionamiento real de la cooperativa. Un asociado puede vincularse a uno o varios vehículos, un vehículo puede registrar múltiples mantenimientos y un conductor puede asociarse a distintos vehículos según las necesidades operativas. Estas relaciones permiten conservar la trazabilidad de la información y facilitan la consulta de registros históricos.

El **diagrama de clases** representa la estructura estática del sistema, mostrando las principales entidades del dominio (por ejemplo, vehículos, conductores, asociados, mantenimientos, usuarios y documentación), sus atributos, relaciones y dependencias. Este diagrama presentado en la figura 10, permite comprender cómo se modela la información en el sistema y sirve como base para el diseño de la capa de datos y la implementación de la lógica de negocio

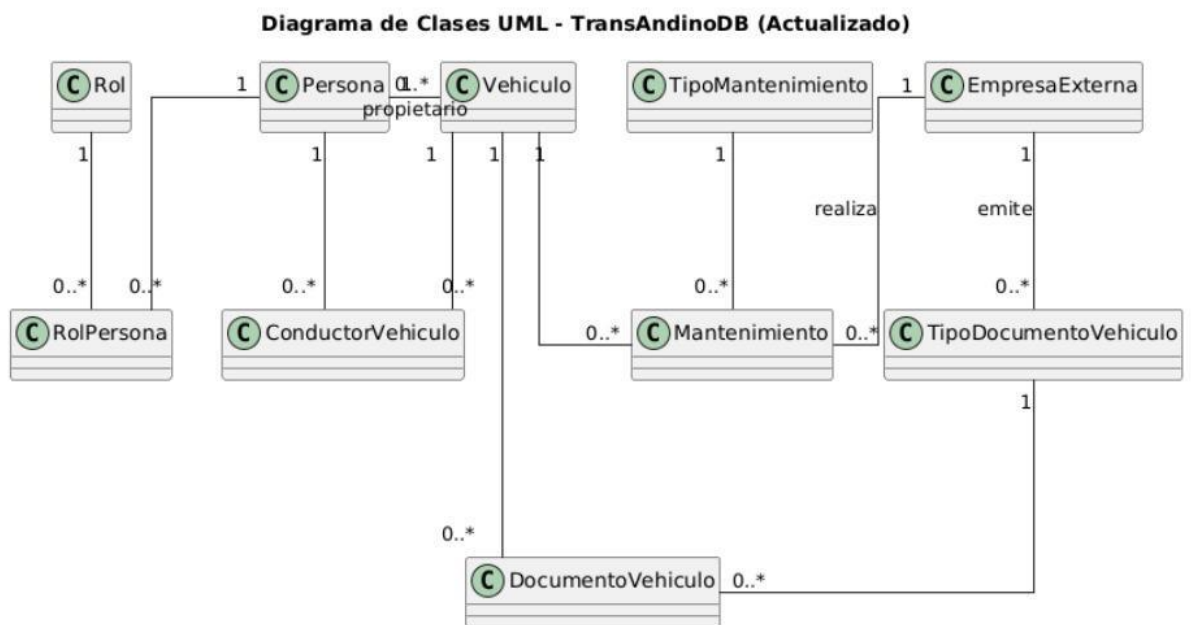


Figura 10 - Diagrama Clases en alto Nivel Fuente - propio

Por otro lado, **el diagrama de casos de uso** describe las funcionalidades del sistema desde la perspectiva de los actores, identificando las interacciones entre el Administrador, el Asociado y el Sistema con los principales módulos del sistema. Adicionalmente, presenta la relación entre los casos de uso de gestión (casos padres) y sus operaciones específicas (casos hijo) mediante relaciones UML como «include», lo que facilita la comprensión del alcance funcional del sistema y su trazabilidad con los requerimientos definidos. La Figura 11 muestra el diagrama de casos de uso con

todas gestiones, el cual mantiene una relación directa con las funcionalidades del sistema.

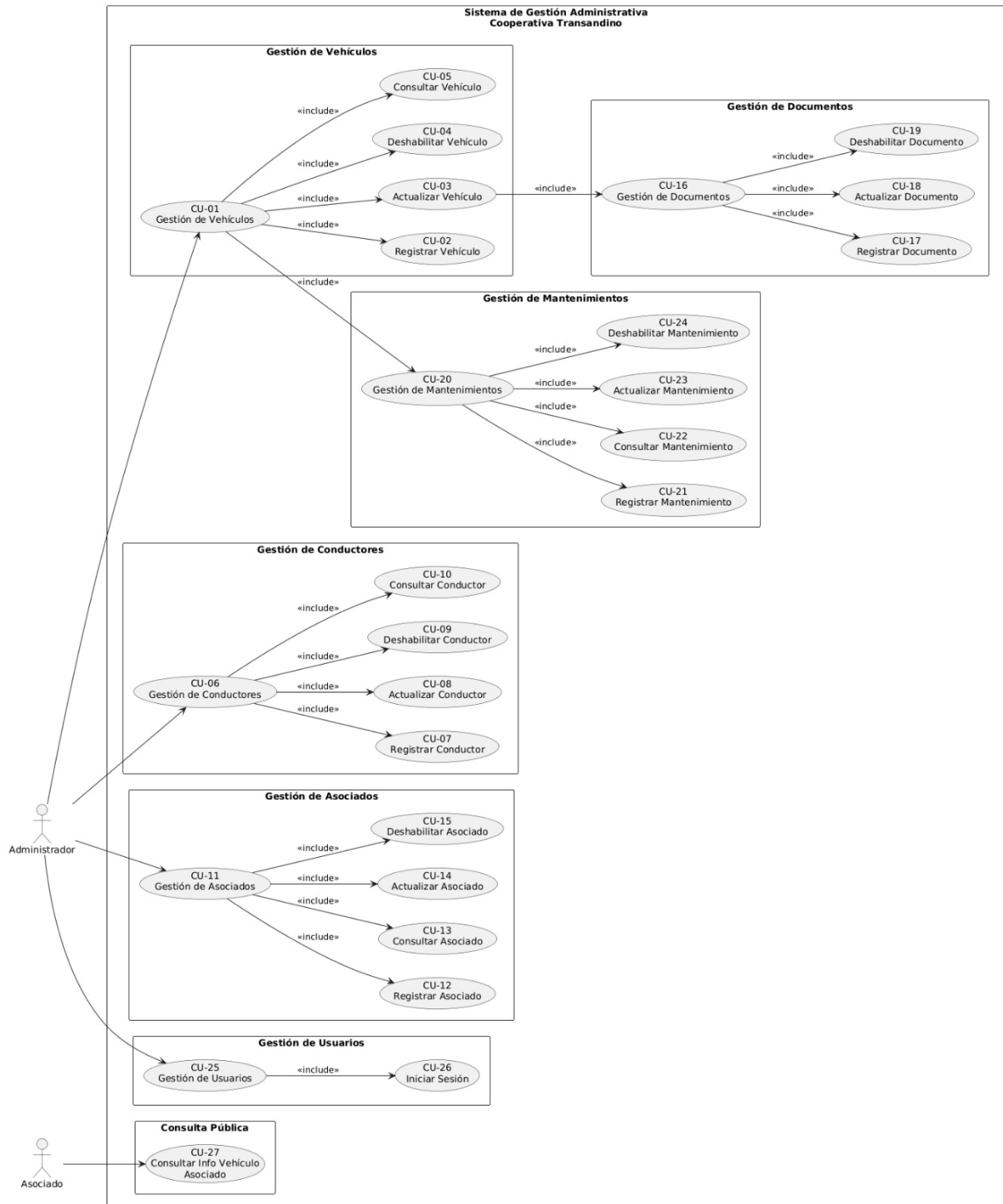


Figura 11 - Diagrama Casos de Uso- Gestión de Vehículos. Fuente – Propio .

El diagrama de secuencia ilustra el flujo de interacción entre el cliente (frontend), el servidor Express, los middlewares de seguridad, los controladores, los modelos y la base de datos MySQL para las operaciones del sistema. Este diagrama que se muestra en la figura 12 permite visualizar el orden en que se ejecutan las validaciones, el procesamiento de la lógica de negocio y el acceso a los datos, ejemplificando el manejo de una operación típica bajo la arquitectura cliente–servidor implementado.

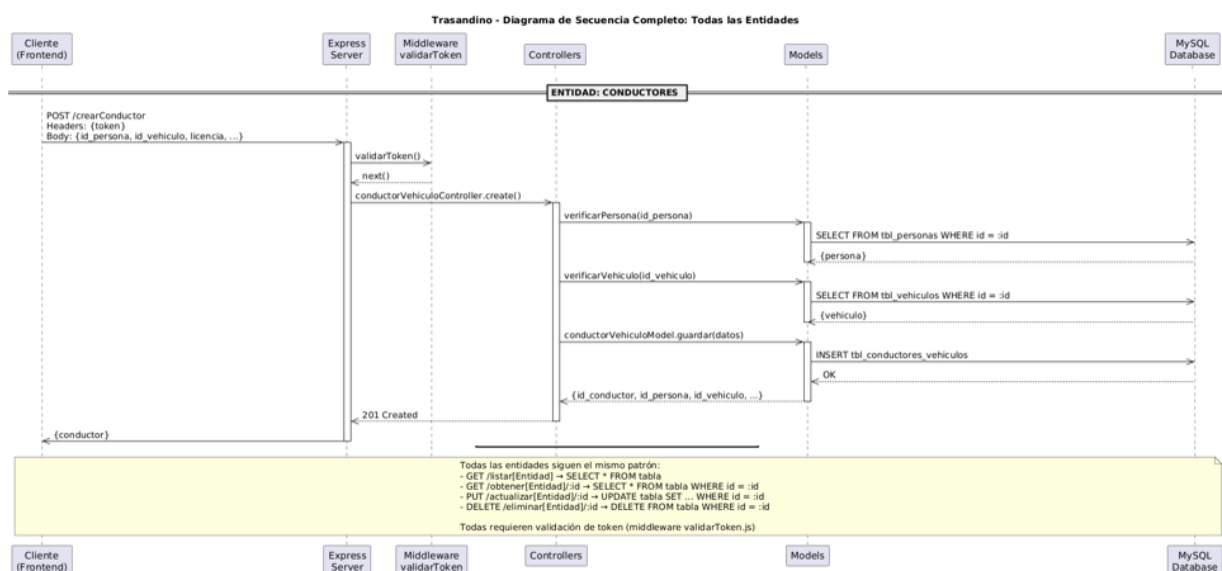


Figura 12 - Diagrama de secuencia. Fuente - Propio

El detalle completo de los diagramas UML del sistema, incluyendo diagramas de clases, casos de uso (separado por gestiones) y secuencia para las distintas entidades, se presenta en el Anexo B.

Mockups e Interfaces

El diseño de mockups e interfaces permite definir de manera anticipada cómo se organizarán y presentarán las funcionalidades del sistema de información propuesto para la Cooperativa Integral de Transporte AndinoTransAndino. A partir de

estas representaciones visuales se establecen criterios de navegación y disposición de la información, los cuales orientan el desarrollo de la capa de presentación. Un ejemplo de esta planificación inicial se observa en la figura 13, donde se define una interfaz de acceso simplificada que asegura que la solución responda de forma coherente a las necesidades administrativas desde el primer contacto con el usuario.

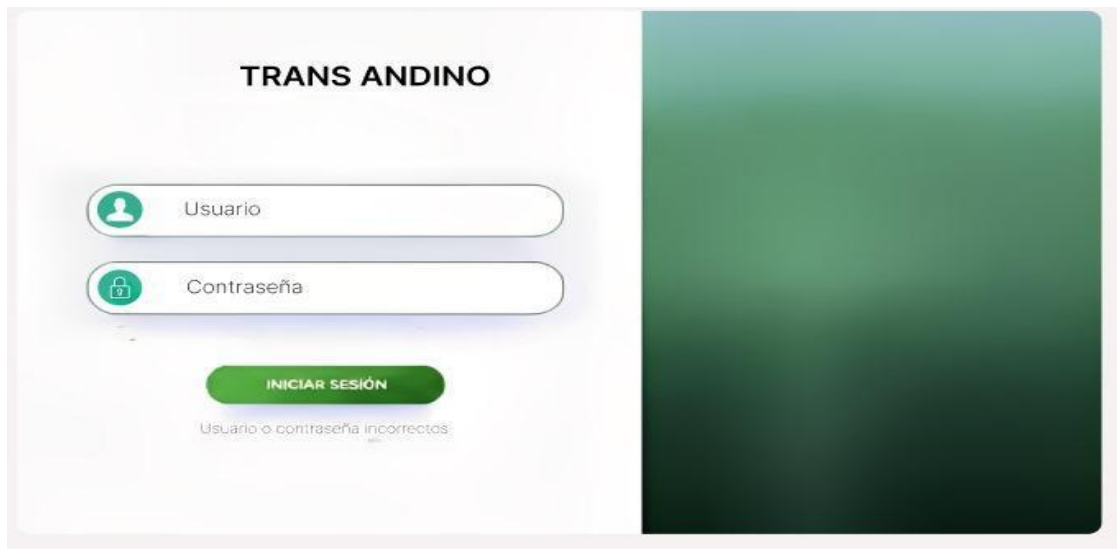


Figura 13 - Mockup preliminar. Fuente - Propio.

Desde un enfoque de usabilidad, el diseño se orienta a la simplicidad y claridad, considerando que los usuarios del sistema no necesariamente cuentan con formación técnica especializada.

Asimismo, el uso de mockups permite incorporar procesos de retroalimentación antes del desarrollo definitivo, La comparación entre el acceso principal figura 13 y el panel de administración figura 14 favorece la identificación de ajustes necesarios en la secuencia de acciones y la claridad de los mensajes. la secuencia de acciones o la claridad de los mensajes presentados al usuario. En conclusión, los mockups constituyen un insumo fundamental que garantiza la

coherencia entre los requerimientos funcionales y la experiencia del usuario, resultando en una plataforma accesible y adecuada a las condiciones reales de la cooperativa.

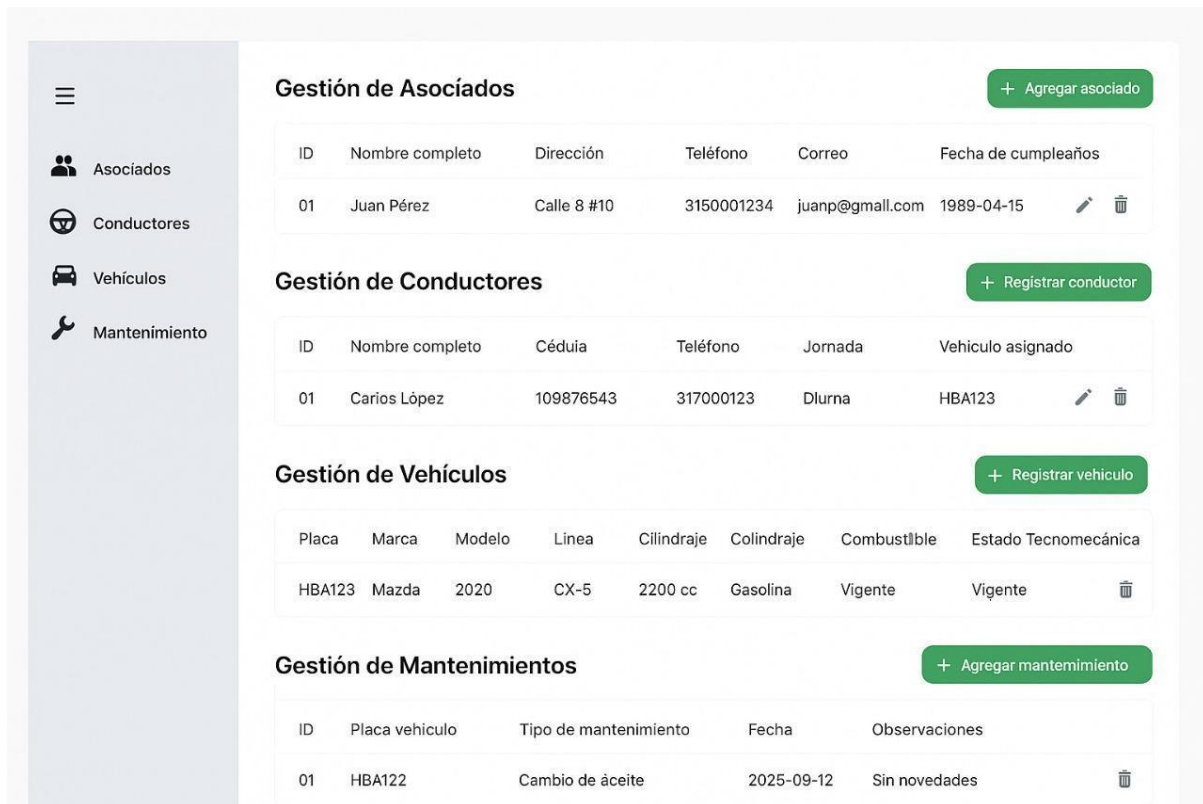


Figura 14 - Mockup Secciones. Fuente - Propio.

Desarrollo del Sistema

El desarrollo del sistema permite implementar y validar las funcionalidades definidas para la gestión administrativa de la Cooperativa Integral de Transporte Andino TransAndino, asegurando que los módulos construidos respondan a los requerimientos funcionales y a las reglas de negocio establecidas. En esta fase se integran la lógica del sistema, el acceso a la base de datos y la interfaz de usuario,

garantizando un funcionamiento coherente, controlado y alineado con los procesos reales de la organización.

El proceso de desarrollo se lleva a cabo de manera incremental, priorizando la implementación de las funcionalidades esenciales para la gestión administrativa de la cooperativa. Esta estrategia permite obtener versiones funcionales del sistema en etapas tempranas, facilitando la validación continua con los usuarios y la identificación oportuna de ajustes necesarios. De acuerdo con Pressman (2015), un desarrollo iterativo contribuye a reducir riesgos y mejora la calidad del software al incorporar retroalimentación constante durante su construcción.

Durante esta fase se implementan los módulos de gestión de personas, vehículos, conductores, documentación y mantenimientos, respetando las relaciones establecidas en el modelado de datos y los flujos definidos en los mockups e interfaces. Cada módulo es desarrollado considerando reglas de negocio específicas, controles de validación y mecanismos que garantizan la integridad y consistencia de la información almacenada en la base de datos.

Igualmente, el desarrollo del sistema contempla la aplicación de buenas prácticas de programación orientadas a la mantenibilidad y escalabilidad de la solución. Esto incluye la separación de responsabilidades entre la interfaz de usuario, la lógica de negocio y la capa de acceso a datos, lo cual facilita futuras mejoras, correcciones y posibles ampliaciones del sistema sin afectar su funcionamiento general. Sommerville (2016) señala que una estructura bien organizada permite prolongar la vida útil del software y adaptarlo a nuevas necesidades organizacionales.

Implementación del Frontend

La implementación del frontend permite establecer cómo el usuario ejecuta las operaciones administrativas del sistema, definiendo la estructura de navegación, los puntos de interacción y la forma en que se presentan los datos. Para garantizar una experiencia de usuario óptima, se utilizó Figma como herramienta de diseño UI/UX en la creación de los mockups y así prototipar la interfaz antes de su codificación en React, como muestra la figura 15, puede apreciarse el tablero de trabajo. A través de esta fase se garantiza que funcionalidades como la gestión de conductores, vehículos, mantenimientos y asociados puedan ser utilizadas de manera intuitiva y coherente con los procesos reales de la Cooperativa Integral de Transporte Andino TransAndino.

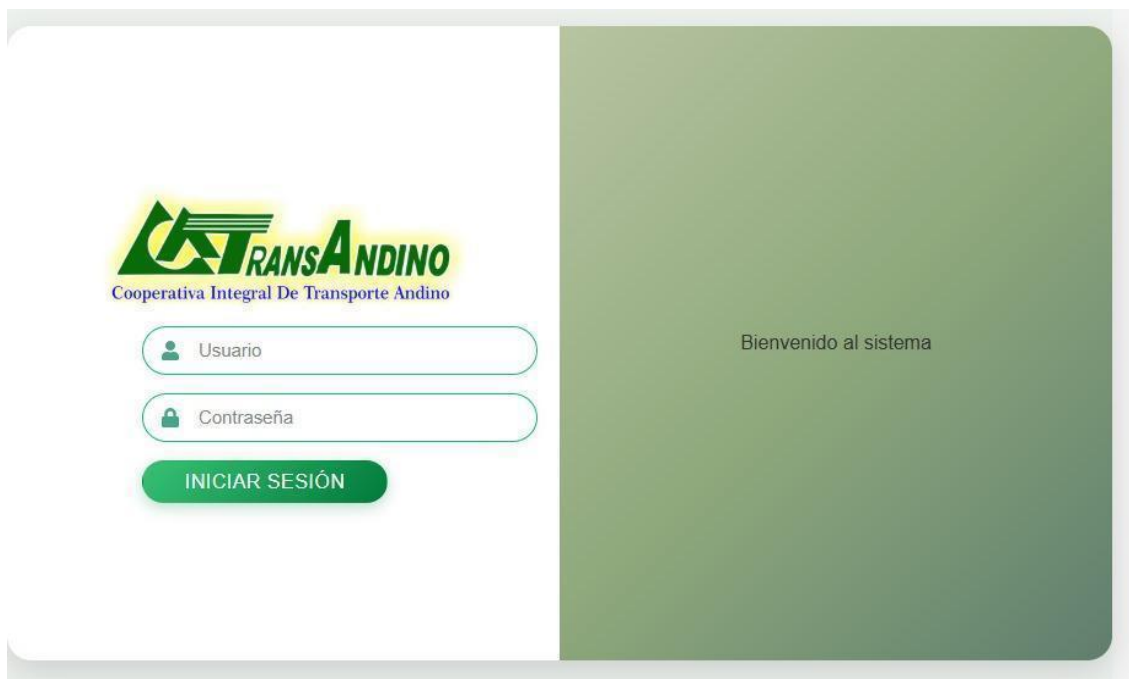


Figura 15 - *Página Principal*. Fuente - *Propio*.

El frontend se desarrolla a partir de los mockups previamente diseñados, respetando la estructura funcional del sistema y el modelo de datos establecido. Las

interfaces implementadas incluyen el módulo de autenticación, el panel general y los módulos de gestión de conductores, vehículos, mantenimientos y asociados, permitiendo al usuario acceder de forma organizada a la información relevante según su rol dentro de la cooperativa. Esta organización visual facilita la comprensión del sistema y reduce la posibilidad de errores durante el uso cotidiano, como muestra la figura 16.

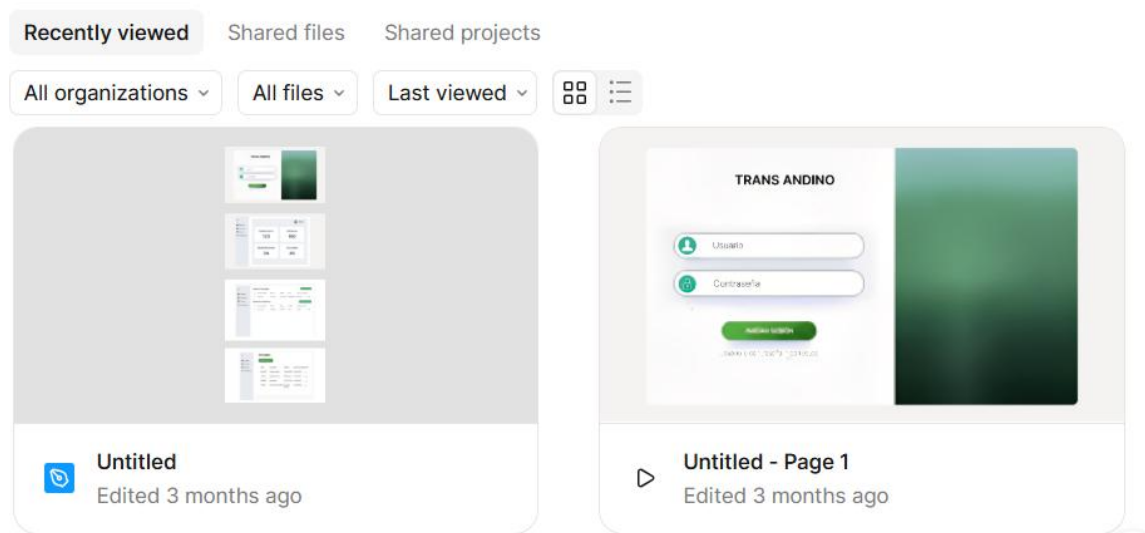


Figura 16 - Tablero de trabajo Figma. Fuente - Propio.

La estructura de paquetes del proyecto frontend TransAndino-frontend, está organizado bajo el directorio src siguiendo el enfoque de **Atomic Design** para el desarrollo de interfaces en React, como muestra la figura 17.

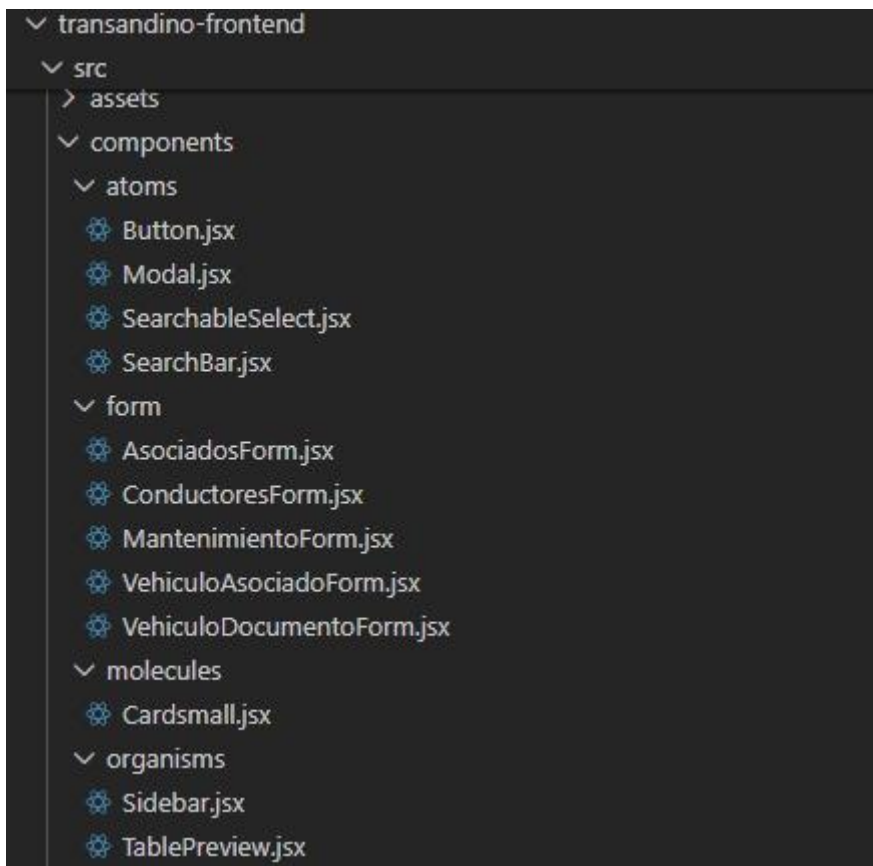


Figura 17 - *paquetes Form-end*. Fuente: *Propio*.

En el nivel superior se encuentra la carpeta **assets**, destinada al almacenamiento de recursos estáticos. La carpeta **components** agrupa los componentes reutilizables del sistema y se subdivide en:

- **atoms**: como botones, modales, barras de búsqueda y selectores, que constituyen los elementos más simples de la interfaz, como muestra la figura 18.

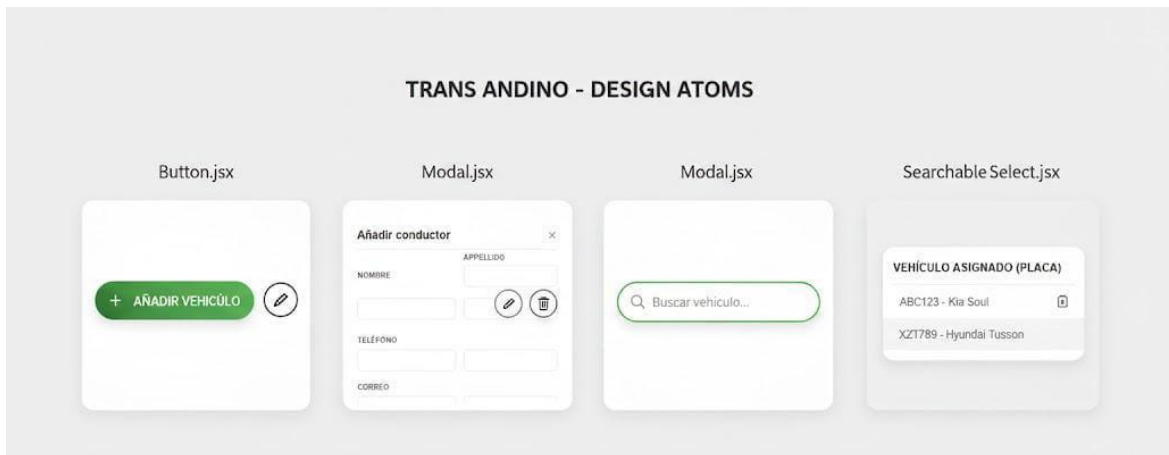


Figura 18 - Átomos – Atomic Design Frontend.

- **Moléculas:** componentes compuestos por la combinación de átomos, como tarjetas de visualización, que permiten construir interfaces con mayor funcionalidad. Como muestra la Figura 19.

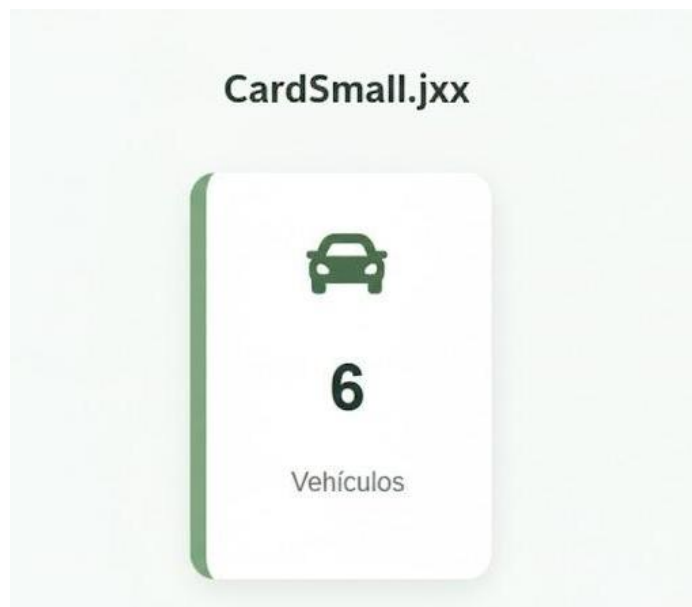


Figura 19 - Formularios registro de vehículo. Fuente - Propio.

- **organismos:** componentes de mayor complejidad estructural, como el menú lateral y las tablas de previsualización, que integran múltiples átomos y moléculas para conformar secciones completas de la interfaz. La figura 20 .

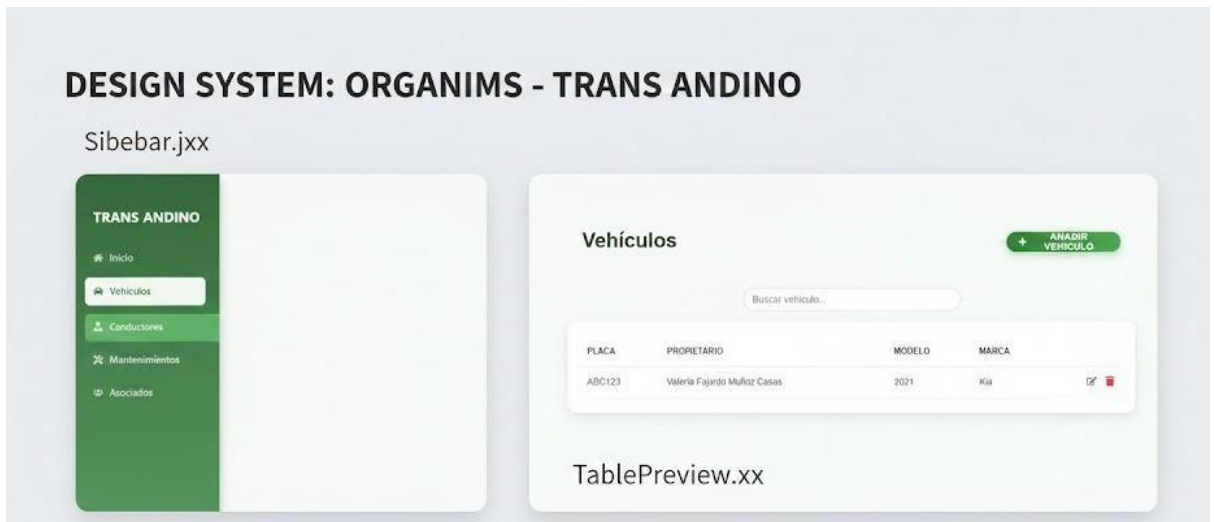


Figura 20 – Molécula Atomic design Frontend. Fuente – Propio.

- **Page**, muestra el armado de páginas completas como formularios específicos del dominio del sistema, orientados a la gestión de asociados, conductores, mantenimientos, vehículos y documentación, los cuales encapsulan la lógica de captura de datos, tal como se representa en la figura 21. Estas funcionalidades aseguran que el usuario pueda registrar, consultar y actualizar los datos de forma inmediata, manteniendo coherencia con las reglas de negocio definidas en el sistema y con los permisos establecidos para cada tipo de usuario.

Desde el enfoque de usabilidad, la implementación prioriza una navegación intuitiva, el uso de iconografía representativa y una distribución coherente de los elementos visuales, los íconos incluidos en el sistema son parte de la librería de React para así mantener un solo estándar de tamaño y forma

Añadir Vehículo y Documentos ×

Propietario *

Datos Técnicos

PLACA	MARCA	MODELO
<input type="text"/>	<input type="text"/>	<input type="text"/>
LÍNEA	MOTOR	CHASIS
<input type="text"/>	<input type="text"/>	<input type="text"/>
CILINDRAJE	COMBUSTIBLE	MÓVIL
<input type="text"/>	<input type="text"/>	<input type="text"/>

Documentos

+ AÑADIR DOCUMENTO

REGISTRAR

Figura 21 - Formularios registro de vehículo. Fuente: Propio.

El menú lateral permite un acceso rápido a los módulos principales como muestra la figura 22, mientras que los paneles informativos presentan datos resumidos que apoyan la supervisión administrativa y la toma de decisiones. De acuerdo con Nielsen (2012), una interfaz bien estructurada contribuye a mejorar la eficiencia del usuario y a disminuir la carga cognitiva durante la ejecución de tareas repetitivas.

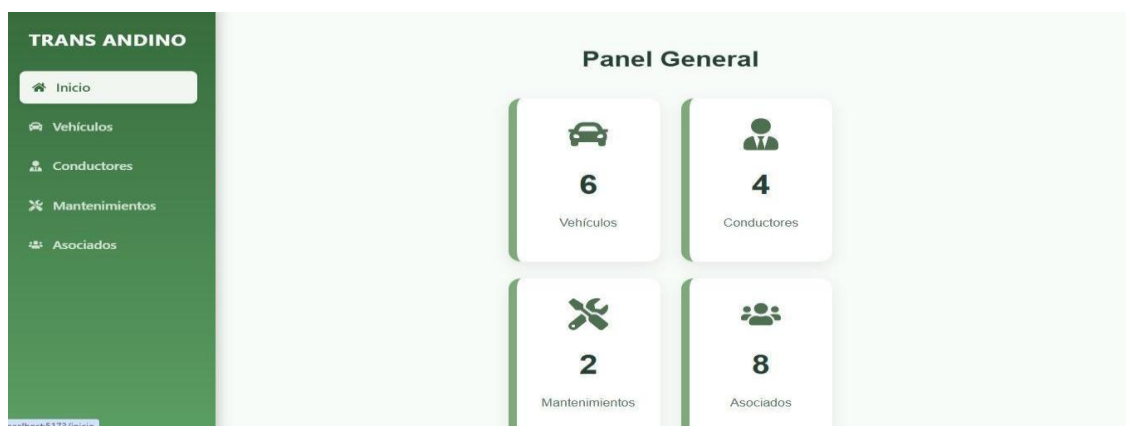


Figura 22 -Dashboard. Fuente -Propio.

Esta organización facilita la **modularidad, reutilización y mantenibilidad del código**, permitiendo un desarrollo escalable y alineado con buenas prácticas de diseño de interfaces en aplicaciones web modernas.

Finalmente, la implementación del frontend actúa como el punto de conexión entre el usuario y la lógica del sistema, garantizando que la información almacenada en la base de datos sea presentada de forma clara y accesible. Su correcta ejecución contribuye a que el sistema de información resulte funcional y alineado con los procesos administrativos reales de la Cooperativa Integral de Transporte TransAndino, fortaleciendo el proceso de transformación digital de la organización.

Implementación del Backend

El backend fue estructurado bajo una arquitectura basada en capas, lo que facilitó la separación de responsabilidades entre la gestión de rutas, controladores, modelos y configuración del sistema. Esta organización permitió un desarrollo más ordenado, mejoró la mantenibilidad del código y facilitó la incorporación de nuevas funcionalidades sin afectar el funcionamiento general del sistema. Según Sommerville (2016), este tipo de estructuración contribuye a la escalabilidad y evolución de los sistemas de software.

La implementación del backend del sistema se desarrolló utilizando **Node.js** junto con el framework **Express**, siguiendo una arquitectura basada en capas que permite separar claramente las responsabilidades de configuración, enrutamiento, lógica de negocio, y acceso a datos. Esta organización facilita la mantenibilidad, escalabilidad y comprensión del sistema, además de favorecer la aplicación de

buenas prácticas de desarrollo de software en aplicaciones web. La estructura del backend está diseñada para soportar la comunicación con el frontend mediante servicios REST, garantizar la validación de accesos y gestionar de forma eficiente la persistencia de la información en la base de datos relacional. La figura 22 muestra la estructura del sistema siguiendo la arquitectura planteada.

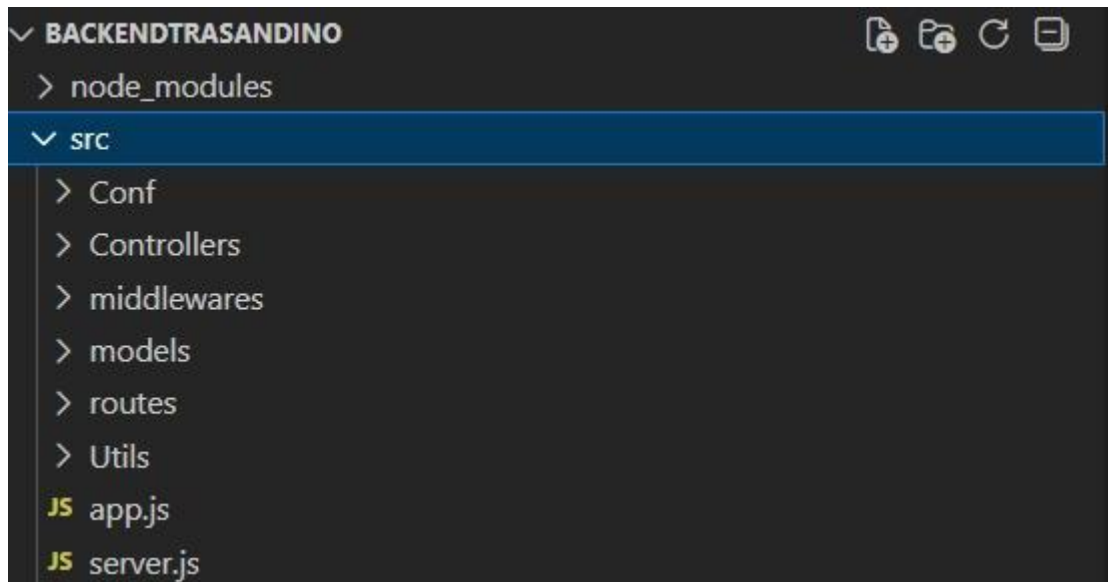


Figura 23 - Estructura del Backend. Fuente - Propio.

A continuación se describen cada uno de los paquetes del Backend

- **Conf:** Este contiene los archivos de configuración general del sistema. En este módulo se definen parámetros esenciales como la conexión a la base de datos, variables de entorno, puertos de ejecución, claves de seguridad y configuraciones globales del servidor. Centralizar esta información permite modificar el comportamiento del sistema sin afectar directamente la lógica de negocio, facilitando la adaptación a distintos entornos (desarrollo, pruebas o producción).

- **Controllers:** implementa la lógica de control del sistema. Cada controlador se encarga de recibir las solicitudes HTTP provenientes de las rutas, procesar la información validada y coordinar las operaciones necesarias con los modelos. Los controladores actúan como intermediarios entre la capa de presentación y la capa de datos, garantizando que las reglas de negocio se ejecuten de manera coherente antes de generar una respuesta al cliente.
- **Middlewares:** agrupa los componentes encargados de interceptar las solicitudes antes de que lleguen a los controladores. En este módulo se implementan funciones transversales como la validación de tokens de autenticación, control de acceso por roles, manejo de errores y validación de datos. El uso de middlewares contribuye a reforzar el sistema y a mantener una lógica limpia y reutilizable en los controladores.
- **Models:** define la estructura de los datos y la interacción directa con la base de datos. En este módulo se representan las entidades del sistema, tales como vehículos, conductores, asociados, mantenimientos y usuarios, así como las operaciones de consulta, inserción, actualización y eliminación. Los modelos permiten encapsular el acceso a los datos, garantizando la integridad de la información y facilitando el mantenimiento del código. La figura 24 muestra el diagrama físico en MYSQL de la base de datos, la cual se mapea a node js para el funcionamiento del sistema.

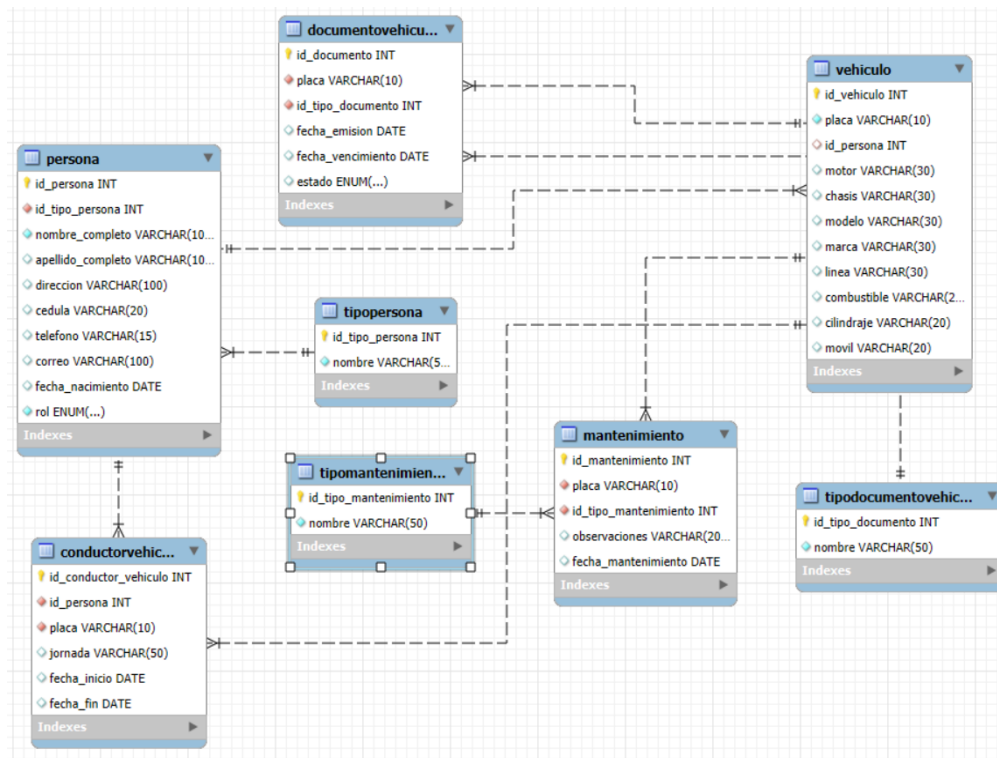


Figura 24 – Diagrama físico en MySQL de la base de datos. Fuente: Propio.

- **Routes:** El paquete **routes** contiene la definición de las rutas del API REST del sistema. Cada archivo de rutas asocia una URL y un método HTTP específico con el controlador correspondiente. Este módulo permite organizar de manera clara los endpoints disponibles, facilitando la trazabilidad entre las funcionalidades del sistema y los servicios expuestos al frontend.
- **Utils:** Agrupa funciones auxiliares reutilizables que apoyan la lógica general del sistema. En este módulo se incluyen utilidades para el manejo de fechas, validaciones comunes, formateo de datos, generación de respuestas estándar y otras operaciones transversales que no pertenecen directamente a la lógica de negocio principal.
- **node_modules:** Contiene las dependencias externas utilizadas por el backend, instaladas mediante el gestor de paquetes npm. Aunque no

forma parte directa del desarrollo del sistema, es fundamental para el correcto funcionamiento del proyecto, ya que incluye las librerías necesarias para la ejecución del servidor y la comunicación con la base de datos.

Flujo de Funcionamiento del Backend

El funcionamiento del Backend inicia con la ejecución del archivo `server.js`, el cual actúa como punto de entrada principal del sistema. En este archivo se define el puerto de escucha y se pone en marcha la aplicación configurada en `app.js`, permitiendo que el servidor comience a recibir las solicitudes provenientes del cliente.

Una vez iniciado el servidor, el archivo `app.js` se encarga de la inicialización y configuración de la aplicación Express, registrando los middlewares globales, las rutas del sistema y el manejador centralizado de errores. A partir de esta configuración, cada solicitud HTTP es procesada siguiendo una arquitectura basada en capas.

Cuando el servidor Express recibe una solicitud, ésta incluye un token JWT que es validado por un middleware de autenticación contra la clave secreta del sistema. Si el token es íntegro y válido, la solicitud es autorizada y transferida al Controller, correspondiente a la capa de lógica de negocio, donde se ejecutan las operaciones principales del sistema.

El controlador actúa como el núcleo del procesamiento, extrayendo los datos de la solicitud y sometiéndolos a validaciones estrictas mediante módulos de utilidad, con el fin de garantizar la consistencia y confiabilidad de la información antes de cualquier interacción con la capa de persistencia. Superada esta etapa, el controlador

invoca al Model, responsable de gestionar la comunicación directa con la base de datos MySQL, ejecutando consultas SQL parametrizadas que protegen la integridad de los datos almacenados.

Finalmente, el sistema genera una respuesta estructurada en formato JSON, la cual es interceptada por el manejador centralizado de errores definido en app.js. Este componente analiza las excepciones generadas durante el procesamiento ya sean de validación, autenticación o de tipo operativo y retorna una respuesta uniforme al frontend, permitiendo informar adecuadamente al usuario o redirigirlo según la naturaleza del fallo., el flujo se muestra en la figura 25.

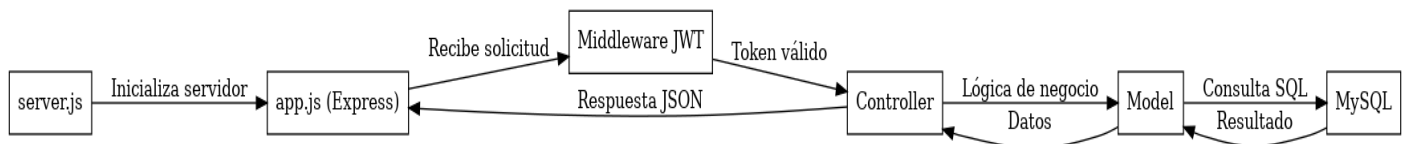


Figura 25 – Flujo del Backend. Fuente - Propio.

La figura 26 representa la estructura general del backend del sistema, organizada bajo una arquitectura modular por capas. Se identifican los módulos de configuración, rutas, controladores, modelos, middlewares y utilidades, los cuales permiten separar responsabilidades como la inicialización del servidor, la validación de accesos, la lógica de negocio y el acceso a la base de datos. Esta organización facilita la mantenibilidad del sistema, la reutilización del código y la correcta implementación de los servicios REST que soportan las funcionalidades del aplicativo.

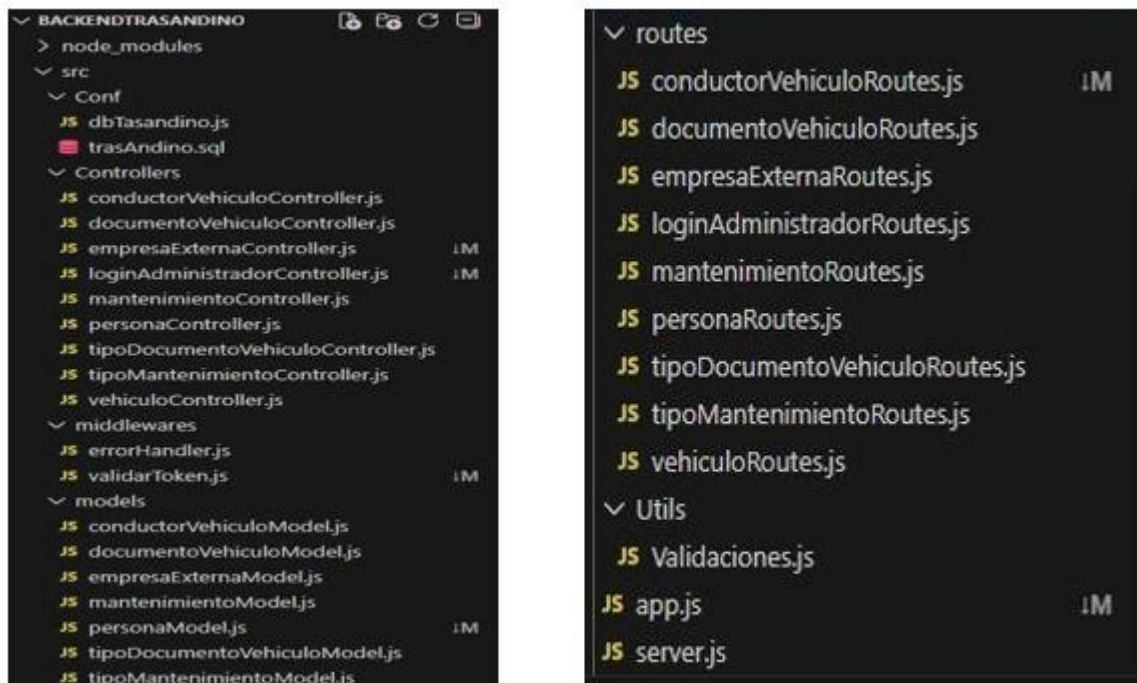


Figura 26 – Estructura backend Fuente - Propio.

En conclusión, el flujo del backend se estructura sobre una arquitectura por capas que permite gestionar de manera ordenada las solicitudes del sistema, desde la inicialización del servidor hasta la persistencia de la información. La integración de middlewares para la validación de credenciales , controladores para la lógica de negocio y modelos para el acceso a la base de datos garantiza un procesamiento consistente, mantenible y escalable. Este enfoque facilita la trazabilidad de las funcionalidades y sienta las bases para futuras extensiones o mejoras, manteniendo la coherencia entre el diseño y la implementación del backend.

A continuación se adjunta el ANEXO B correspondiente a los Diagramas UML el cual consolida el cumplimiento del objetivo.

Capítulo 3. Validación del sistema TransAndino

Pruebas de Validación del Backend

La implementación de las pruebas realizadas sobre el backend del sistema, utilizando la herramienta Thunder Client que es una extensión ligera de cliente API REST para Visual Studio Code y sirve para probar APIs (HTTP/REST) sin necesidad de salir de Visual Studio Code, editor de código fuente empleado durante el desarrollo del proyecto. A través de esta herramienta se ejecutaron diferentes peticiones HTTP, específicamente los métodos GET, POST, PUT y DELETE, con el fin de validar el comportamiento de las distintas entidades que conforman el sistema.

Durante la ejecución de las pruebas, se analizaron los códigos de respuesta HTTP generados por el servidor, siendo los más comunes el código 200, que indica que la solicitud fue procesada correctamente y el servidor respondió de manera exitosa, y el código 400, que señala que el cliente envió una solicitud mal formada o inválida, generando un error controlado por el sistema. Aunque existen otros códigos de estado, estos fueron los más recurrentes durante las pruebas funcionales realizadas.

A continuación, se presentan los resultados obtenidos al ejecutar diferentes pruebas correspondientes a las entidades del backend, evidenciando el correcto funcionamiento de los servicios implementados de la clase persona.

La clase **Persona** se considera una de las principales dentro del sistema, ya que el primer proceso fundamental consiste en el registro de las personas, ya sean **Asociados, Conductores o empleados**. Este registro es indispensable para

garantizar la correcta ejecución de los demás procesos del sistema, evitando inconsistencias o fallos en el manejo de la información.

a. Petición ListarPersonas

La figura 27 muestra esta petición que permite visualizar la información de las personas registradas en el sistema, incluyendo datos como nombre, apellido, número de identificación, entre otros.

Se trata de una petición de tipo GET, utilizada para consultar y obtener datos almacenados, sin modificar el estado de la información.

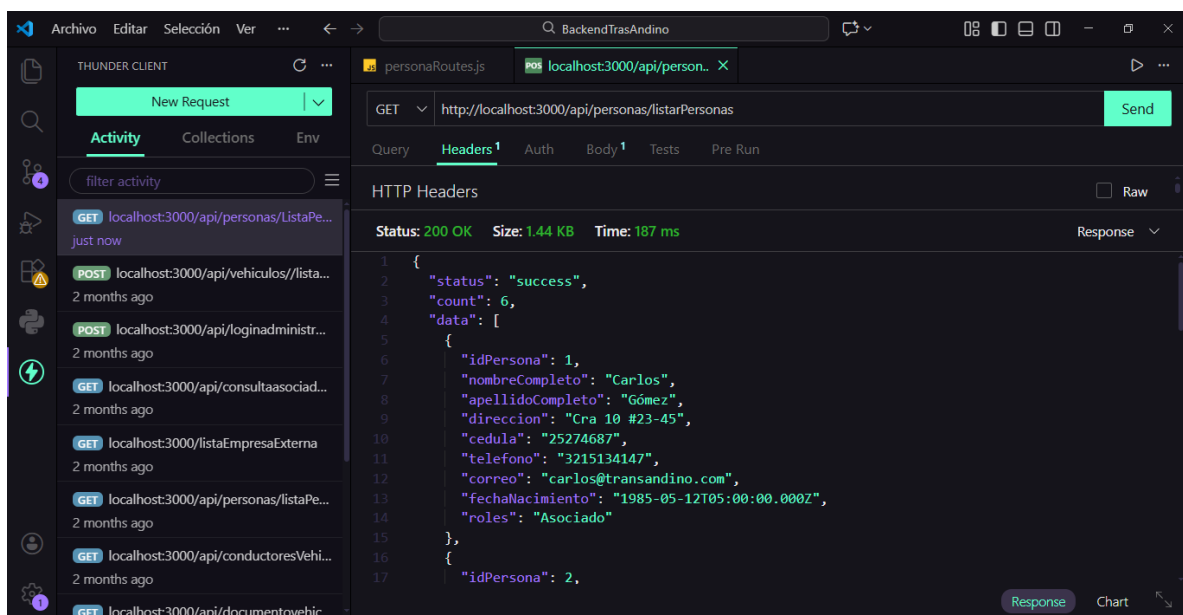


Figura 27 – Pruebas backend _ Listar Personas. Fuente - Propio

Como se puede observar en la imagen, esta petición retorna diferentes tipos de datos, además del total de personas registradas en el sistema. Asimismo, la respuesta presenta el código de estado HTTP 200, lo cual indica que la solicitud se ejecutó de manera exitosa, sin presentar ningún tipo de error.

b. Petición “BuscarPersonaPorCedula”

Esta funcionalidad corresponde a un filtro de búsqueda, el cual permite consultar a cualquier persona registrada en el sistema a través de su número de cédula, con el fin de visualizar información relevante y otros datos de interés que se detallarán más adelante. La figura 28 muestra la ejecución de la prueba.

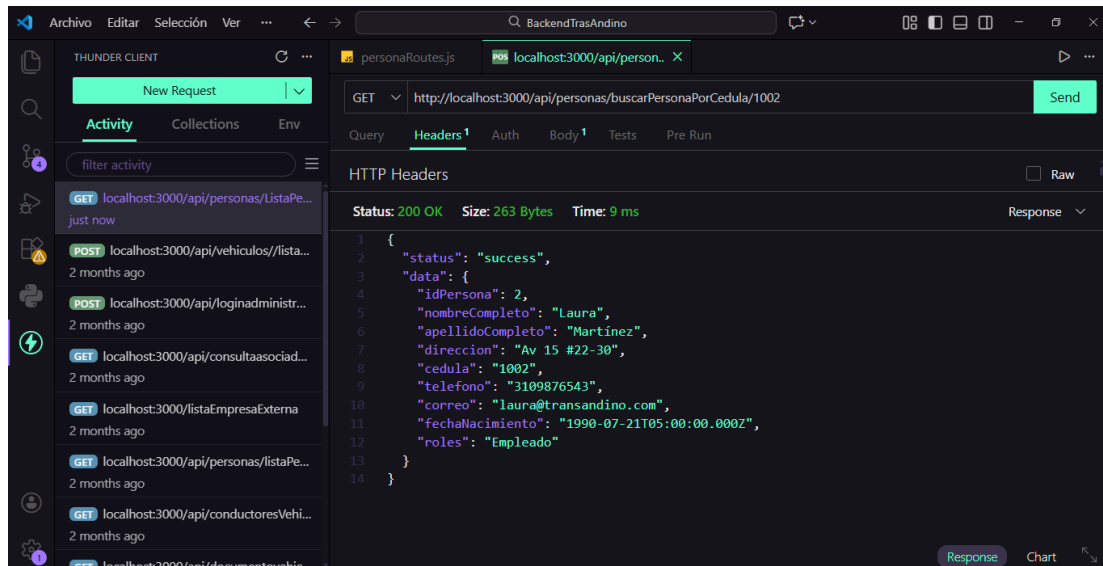


Figura 28 – Pruebas backend _ Buscar Persona Por Cédula. Fuente - Propio

Como se puede apreciar en la imagen, la consulta se realiza mediante el método GET y se ejecuta de manera exitosa, lo cual se evidencia en la respuesta con el código de estado HTTP 200.

c. petición “CrearPersonas”

Esta petición es de gran importancia, ya que a través de ella se realiza la creación de las diferentes personas dentro del sistema, tales como asociados, conductores y empleados. Cada persona es registrada con su respectivo rol, el cual permite identificar su función dentro de la plataforma. En este proceso también se definen

otros datos relevantes, como nombre, número de identificación, número de cédula, entre otros, tal como se muestra en la Figura 29.

Para esta operación se utiliza el método POST, el cual permite crear nuevos registros en el sistema. A diferencia de las peticiones anteriores, la respuesta retorna el código de estado HTTP 201, indicando que el recurso fue creado exitosamente.

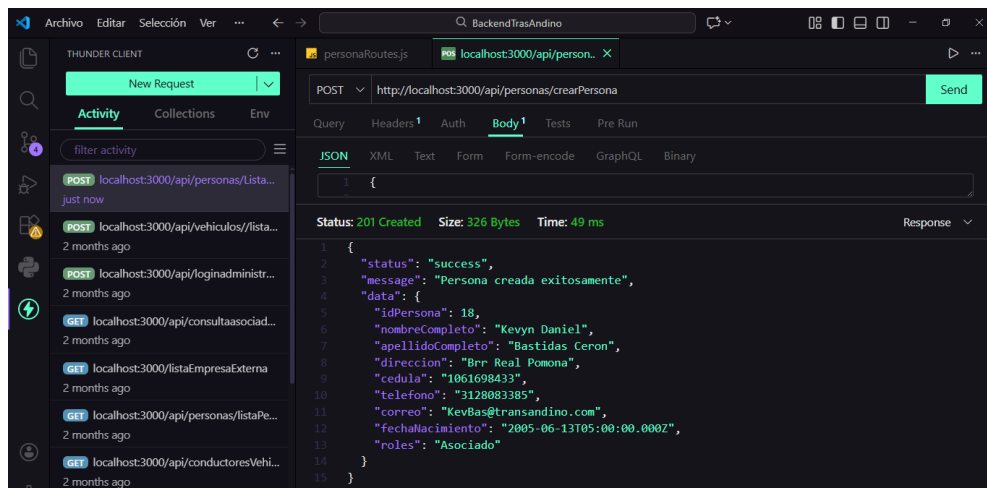


Figura 29 – Pruebas backend _ Crear Persona. Fuente- Propio

Adicionalmente, durante la ejecución de esta petición pueden presentarse errores, los cuales serán descritos a continuación. En este caso, el error se presenta debido a que en el campo roles se ingresó el valor “asociados”, mientras que en la base de datos el valor permitido es “asociado” en singular. Esta inconsistencia provoca que la solicitud no pueda ser procesada correctamente.

Como resultado, el sistema retorna el código de estado HTTP 400, lo cual indica que la solicitud enviada por el cliente es incorrecta o inválida, ya que no cumple con las reglas de validación establecidas.

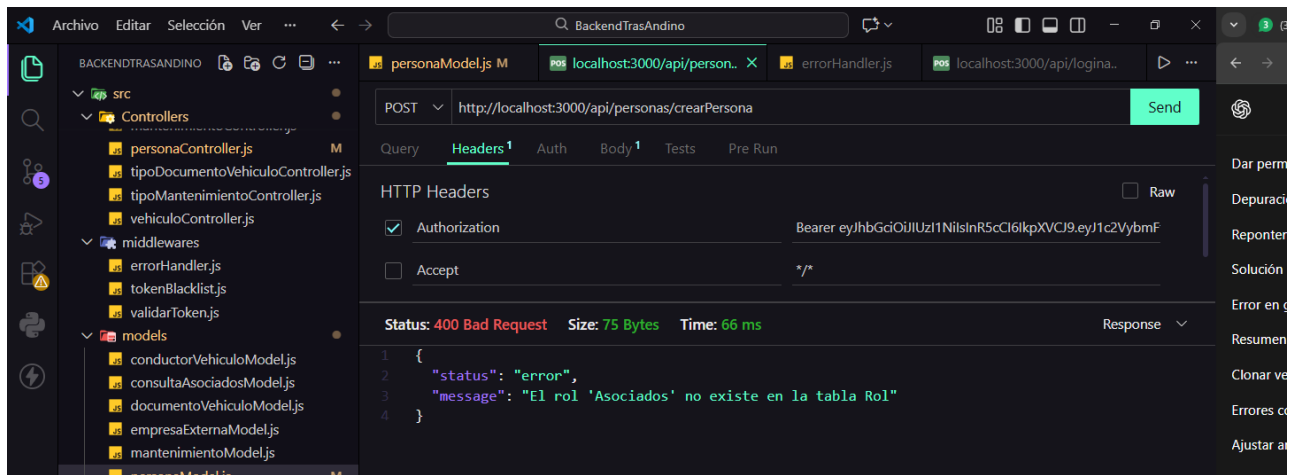


Figura 30– Pruebas backend_ Crear Persona Error Fuente - Propio

d. Petición “ActualizarPersonas”

Esta es una petición de tipo PUT, la cual se utiliza para actualizar recursos o datos que ya existen en el sistema. La figura 31 La actualización se realiza a través del identificador (ID) de la persona, lo que permite ubicar el registro de manera precisa y facilitar el proceso de modificación.

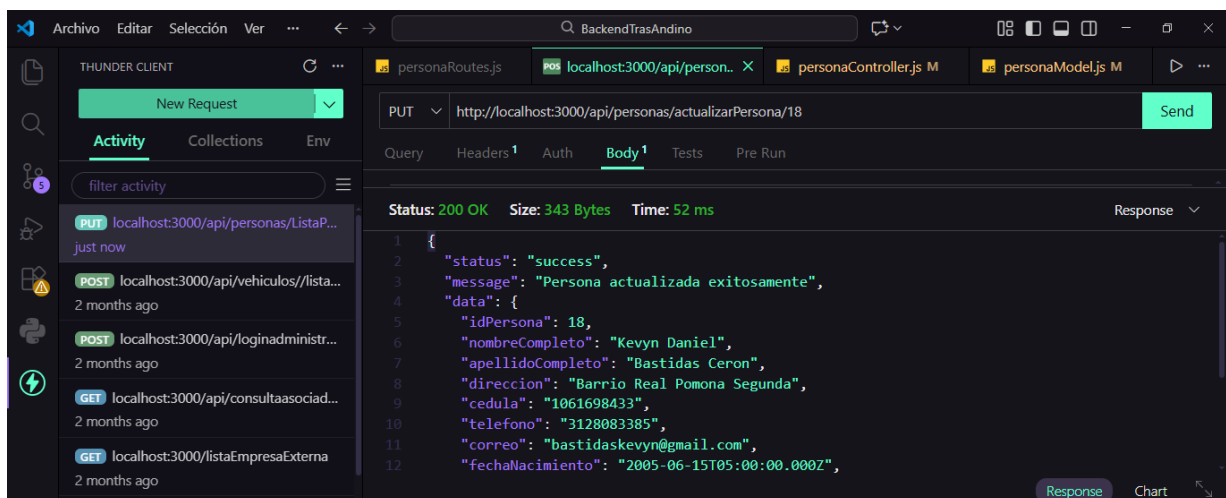


Figura 31– Pruebas backend Actualizar Personas Fuente: Propio

Al comparar con la imagen anterior, se puede observar que se han modificado algunos campos, como la dirección y el correo electrónico, con el fin de ilustrar el funcionamiento de la operación. Como resultado, la respuesta del sistema retorna el

código de estado HTTP 200, lo cual indica que la solicitud se ejecutó correctamente y que el recurso fue actualizado sin inconvenientes.

e. petición “DeshabilitarPersonas ”

Esta petición es de tipo DELETE y se utiliza para deshabilitar un registro existente dentro del sistema. En este caso, la operación permite borrar toda la información asociada a una persona, utilizando su identificador único (ID) como criterio de búsqueda, lo que garantiza que el registro correcto sea eliminado.

Como se puede observar en la Figura 32, al ejecutarse correctamente la solicitud, el sistema procesa la eliminación del recurso sin presentar errores. Este tipo de operación es fundamental para la gestión y depuración de la información, ya que permite mantener la base de datos actualizada y libre de registros innecesarios o incorrectos.

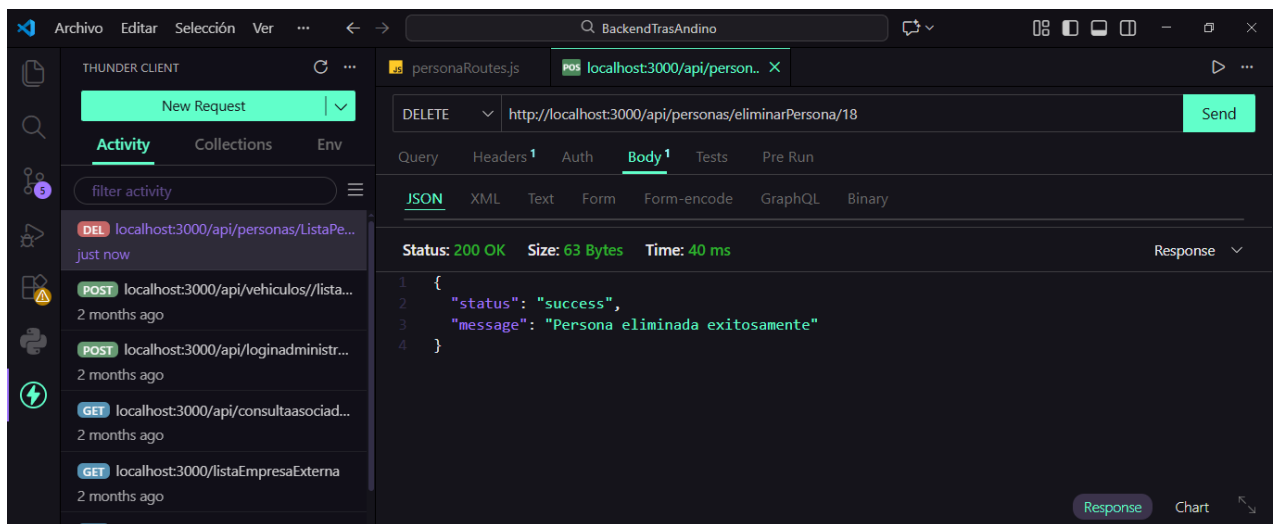


Figura 32– Pruebas backend Eliminar Personas Fuente: *Propio*

Integración frontend–backend

Con el fin de profundizar en la arquitectura técnica del sistema, se implementó un patrón cliente–servidor organizado en capas, adaptado a un entorno Single Page

Application (SPA). En este enfoque, la comunicación entre el frontend y el backend se realiza mediante el protocolo HTTP, operando dentro de un entorno de red local. Esta arquitectura permite una separación clara de responsabilidades y favorece la escalabilidad y mantenibilidad del sistema.

Un aspecto técnico relevante en la capa de presentación es el uso de interceptores de Axios, los cuales actúan como un mecanismo centralizado para la gestión de la seguridad en las solicitudes. Estos interceptores inyectan de forma automática el token de autenticación (Bearer Token) en el encabezado de cada petición, eliminando la necesidad de replicar lógica en cada componente del frontend y garantizando que todas las solicitudes hacia el backend se realicen de manera protegida.

En la capa de servidor, la robustez del sistema se garantiza mediante el manejo de operaciones asíncronas utilizando Async/Await en Node.js, evitando el bloqueo del hilo principal de ejecución mientras se procesan las respuestas de la base de datos MySQL. El acceso a los datos se realiza a través de sentencias preparadas (Prepared Statements), lo que permite separar la lógica SQL de los datos suministrados por el usuario y mitigar riesgos asociados a inyección de código malicioso. Adicionalmente, la persistencia se apoya en un pool de conexiones, que reutiliza un conjunto controlado de conexiones abiertas, reduciendo la sobrecarga de memoria y procesamiento asociada a la apertura y cierre constante de sesiones con la base de datos.

La integridad de la sesión se mantiene mediante un sistema de validación de estado implementado en el frontend. Cuando el backend retorna un código de estado HTTP 401, asociado a un token expirado o inválido, el interceptor de respuestas

captura este evento de forma global, elimina la información almacenada en el local storage y fuerza la redirección inmediata del usuario a la pantalla de inicio de sesión. Este mecanismo asegura que el flujo de información entre la lógica de negocio y el almacenamiento local sea atómico, seguro y confiable. La figura 33 muestra los datos que se traen en el Frontend desde el Backend de la funcionalidad editar conductor.

The screenshot shows a web application interface for 'TRANS ANDINO'. A modal window titled 'Editar conductor' is open, displaying a form with the following fields and values:

Datos del conductor	
NOMBRE	David
APELLIDO	Miranda
CÉDULA	34234890
DIRECCIÓN	Barrio José Maria Obando
TELÉFONO	3284049289
CORREO	davidmira@gmail.com
FECHA DE NACIMIENTO	23/05/1988
VEHÍCULO ASIGNADO (PLACA)	ANB452
JORNADA	Nocturna
FECHA DE INICIO	
FECHA DE FIN	

Figura 33 – Pruebas integración Frontend y Backend - Actualizar ConductorPersonas Fuente -Propio.

En términos de seguridad, el sistema incorpora mecanismos orientados al control de acceso a las funcionalidades administrativas y a la información gestionada por la Cooperativa Integral de Transporte AndinoTrasandino. El proceso de autenticación se realiza mediante la validación de credenciales, y el backend implementa middlewares de seguridad encargados de verificar los permisos del usuario en cada solicitud, garantizando que únicamente personal autorizado acceda a los recursos del sistema.

Estas medidas contribuyen a reducir los riesgos asociados al acceso no autorizado, la manipulación indebida de la información y la pérdida de datos sensibles.

De acuerdo con Sommerville (2016), la implementación de controles de seguridad constituye un elemento esencial en los sistemas de información que gestionan datos administrativos y operativos.

La figura 34 presenta el diagrama de integración frontend–backend del sistema, el cual muestra de manera conceptual el flujo de comunicación entre la capa de presentación, la lógica de negocio y la capa de persistencia. Este diagrama permite identificar los componentes responsables de la seguridad, validación, procesamiento y almacenamiento de la información, sin detallar la secuencia temporal de ejecución.

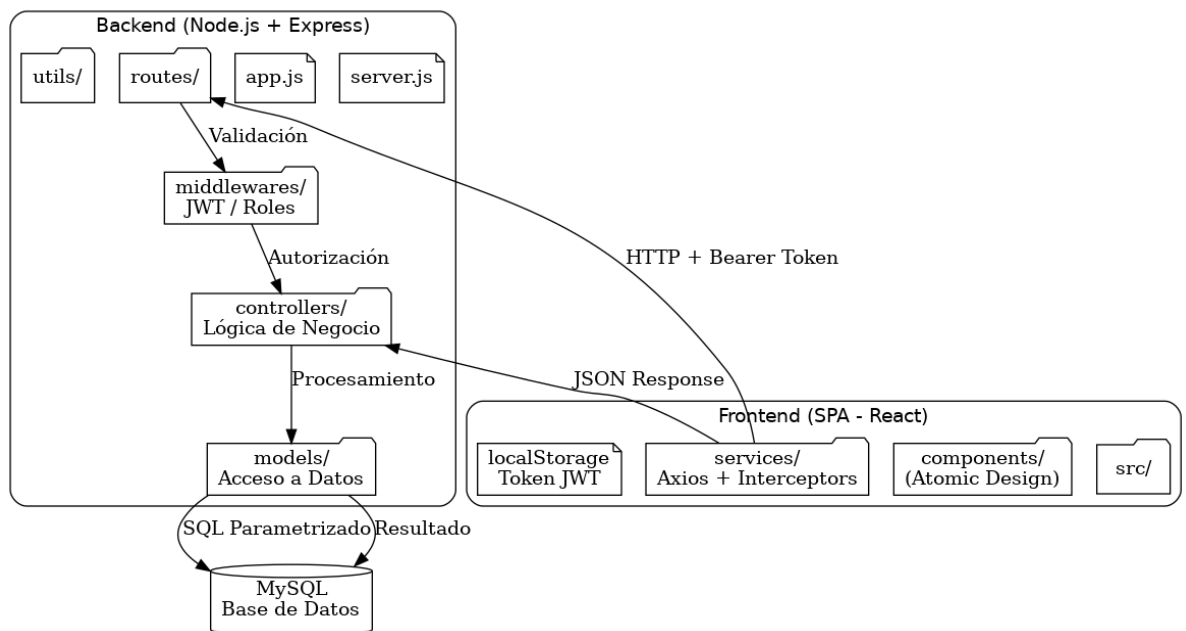


Figura 34– Diagrama de integración – Comunicación del sistema backend Fuente - Propio.

Plan de Pruebas

El plan de pruebas se diseñó con el objetivo de verificar el correcto funcionamiento del sistema de información desarrollado para la Cooperativa Integral de Transporte Andino TransAndino, asegurando que cada uno de los módulos cumpla con los requerimientos funcionales y no funcionales definidos durante las fases de análisis y diseño. Este proceso permitió evaluar la calidad del sistema antes de su puesta en funcionamiento, reduciendo riesgos y garantizando una operación confiable.

El objetivo del plan de pruebas es verificar que las funcionalidades implementadas cumplan los requerimientos del sistema, validando el comportamiento del backend (API REST) y su integración con la base de datos, incluyendo controles de autenticación y autorización.

Se cubren pruebas funcionales y técnicas sobre:

- Autenticación e inicio de sesión (JWT).
- Gestión de vehículos, conductores, asociados, mantenimientos, documentación.
- Manejo de errores, validaciones y códigos HTTP.
- Persistencia y consultas en MySQL.
- Operación CRUD (crear, consultar, actualizar, deshabilitar lógico donde aplique).

Pruebas Funcionales

ID	Requisito	Funcionalidad	Validación	Resultado (%)
RF-01	Gestión de vehículos	Registro, consulta y actualización de datos del vehículo; control de documentos obligatorios (SOAT, tecnomecánica, tarjeta de propiedad) y alertas de vencimiento	Pruebas funcionales sobre registro, edición y consulta de vehículos; verificación de alertas	95 %
RF-02	Gestión de conductores	Registro y administración de información personal, licencias, vinculación a vehículos y jornadas asignadas	Pruebas funcionales de formularios y vinculación conductor-vehículo	93 %
RF-03	Gestión de asociados	Registro, edición y consulta de propietarios; relación entre asociados, vehículos y conductores	Validación mediante casos de uso y revisión de integridad de datos	94 %
RF-04	Módulo de mantenimientos	Registro de mantenimientos preventivos y correctivos; consulta de historial y fechas programadas	Pruebas funcionales de registro y consulta histórica	96 %
RF-05	Gestión de usuarios y accesos	Inicio de sesión con credenciales; roles de administrador y asociado; control básico de actividades	Pruebas de autenticación, roles y permisos	92 %

RF-06	Interfaz amigable e intuitiva	Menús organizados, formularios claros, botones identificables y mensajes de confirmación o error	Pruebas de usabilidad con usuarios finales	97 %
RNF-01	Usabilidad	Facilidad de navegación, claridad visual y adaptación al flujo administrativo	Pruebas de usabilidad y encuestas de satisfacción	96 %
RNF-02	Rendimiento	Respuesta rápida del sistema y carga eficiente de información con grandes volúmenes de datos	Pruebas de carga y tiempos de respuesta	94 %
RNF-03	Escalabilidad	Capacidad de crecimiento en usuarios, registros y módulos usando MySQL y arquitectura modular	Validación estructural y pruebas de crecimiento de datos	90 %

Tabla 8: *Requisitos funcionales y no Funcionales.* **Fuente:** *Propio.*

Las pruebas se realizaron con el propósito de verificar que las funcionalidades del sistema de información desarrollado para la Cooperativa Integral de Transporte Andino TransAndino operen conforme a los requerimientos establecidos. Estas pruebas permitieron validar que cada módulo del sistema responda de manera correcta a las acciones realizadas por el usuario y que los resultados obtenidos coinciden con el comportamiento esperado.

Datos del conductor

VEHÍCULO ASIGNADO (PLACA) **JORNADA**

Escribe para buscar por placa... Seleccione...

FECHA DE INICIO **FECHA DE FIN**

dd/mm/aaaa dd/mm/aaaa

CANCELAR **GUARDAR**

Figura 35- Funciones. Fuente - Propio

Durante esta fase se evaluaron los principales procesos administrativos del sistema, tales como el registro, consulta, actualización y control de información relacionada con vehículos, conductores, asociados, mantenimientos y documentación. Cada prueba se ejecutó siguiendo los flujos definidos en el diseño del sistema, comprobando que la información ingresada fuera procesada adecuadamente y almacenada de forma consistente en la base de datos.

The image shows a web form titled "Nuevo Asociado" (New Associate) with a close button (x) in the top right corner. The form contains the following fields:

- NOMBRE COMPLETO ***: Text input field.
- APELLIDO COMPLETO ***: Text input field.
- CÉDULA ***: Text input field.
- TELÉFONO**: Text input field.
- CORREO ELECTRÓNICO**: Text input field.
- DIRECCIÓN**: Text input field.
- FECHA DE NACIMIENTO**: Date picker with a calendar icon and a placeholder "dd/mm/aaaa".
- ROL**: Dropdown menu with "Asociado" selected.

At the bottom of the form are two buttons: "CANCELAR" (light gray) and "GUARDAR" (green).

Figura 36 - Funcionalidad. Fuente -Propio.

Además, se verificó la correcta interacción entre el frontend y el backend, asegurando que las solicitudes generadas desde la interfaz fueran atendidas por los servicios correspondientes y que las respuestas obtenidas reflejaran los cambios realizados en el sistema. Estas pruebas permitieron identificar y corregir errores relacionados con validaciones incompletas, mensajes de respuesta y manejo de datos.

Los resultados obtenidos a partir de las pruebas funcionales evidenciaron que el sistema cumple con las funcionalidades previstas y puede ser utilizado para apoyar los procesos administrativos de la cooperativa. De acuerdo con Pressman (2015),

este tipo de pruebas resulta fundamental para garantizar que el software satisface las necesidades del usuario y funciona conforme a lo especificado.

Las pruebas se realizaron con el objetivo de evaluar aspectos del sistema que no están relacionados directamente con una funcionalidad específica, pero que influyen de manera significativa en su calidad, desempeño y confiabilidad. Estas pruebas permitieron analizar el comportamiento del sistema de información de la Cooperativa Integral de Transporte Andino TransAndino en condiciones reales de uso, garantizando que su operación sea estable y adecuada para el entorno administrativo en el que será utilizado.

Durante esta fase se evaluaron criterios como el rendimiento del sistema, los tiempos de respuesta ante solicitudes frecuentes de la plataforma durante la ejecución de múltiples operaciones consecutivas.

Los resultados obtenidos a partir de las pruebas evidenciaron que el sistema presenta un comportamiento estable, tiempos de respuesta adecuados y condiciones de seguridad acordes con los requerimientos definidos. Según Sommerville (2016), la evaluación de estos atributos resulta esencial para asegurar la calidad global del software y su viabilidad en entornos organizacionales.

Pruebas de Usabilidad con Usuario Administrador

Las pruebas de usabilidad con el usuario administrador se realizaron con el objetivo de evaluar de manera integral el funcionamiento, la facilidad de uso y la adecuación del sistema de información para la gestión administrativa de la Cooperativa Integral de Transporte Andino TransAndino. Para esta evaluación se contó con la participación del gerente de bienestar de la cooperativa, quien

desempeña un rol clave dentro de los procesos administrativos y operativos de la organización.

La prueba consistió en la ejecución de cinco tareas representativas, diseñadas a partir de escenarios reales de uso del sistema. Estas tareas incluyeron el registro de un asociado, el registro de un vehículo vinculado al asociado, la edición de información de un conductor, la eliminación de un registro de mantenimiento previamente existente y la consulta pública de información mediante la placa del vehículo y el número de identificación del asociado. Las actividades fueron ejecutadas de forma autónoma, con el fin de evaluar la comprensión de la interfaz y la navegación del sistema en la Figura 37 se puede observar una parte del proceso.

Durante la ejecución de las tareas se observó que el usuario logró completar satisfactoriamente todas las actividades propuestas, evidenciando un alto nivel de eficacia en el uso del sistema. Asimismo, los tiempos empleados para la realización de cada tarea se mantuvieron dentro de rangos adecuados, lo que permitió evaluar positivamente la eficiencia del sistema y la organización de sus módulos.

Los resultados obtenidos permitieron validar que el sistema responde de manera correcta a las necesidades administrativas de la cooperativa, presentando una interfaz clara, flujos de navegación coherentes y funcionalidades alineadas con los procesos reales de gestión. Esta prueba constituyó la principal base para la validación de la usabilidad del sistema desarrollado.

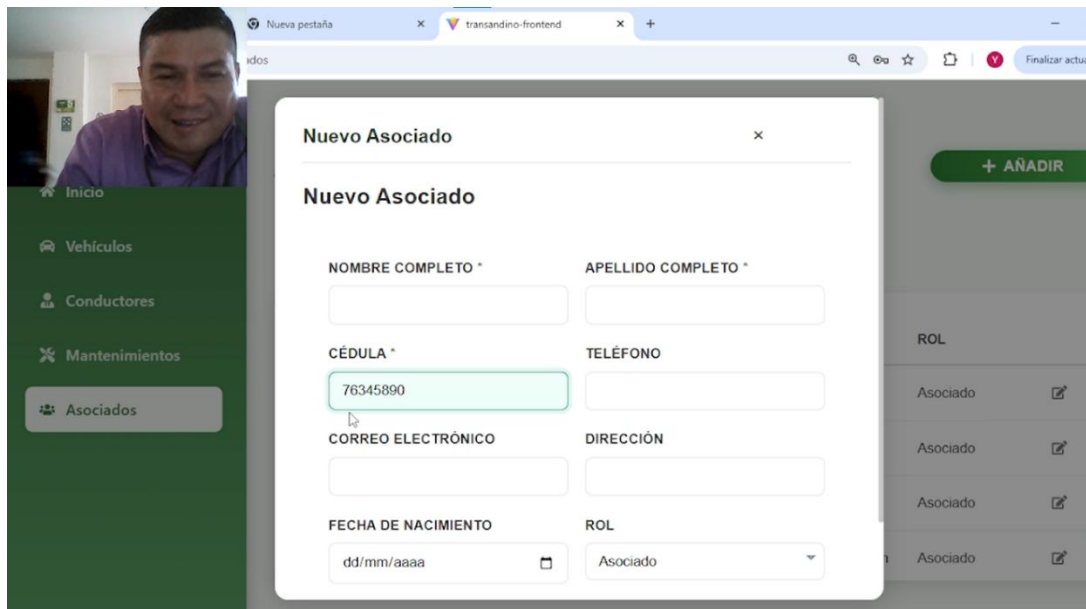


Figura 37- Proceso del test de Usabilidad con Usuario Administrador. Fuente - Propio

Pruebas de Usabilidad con Usuarios Finales

Las pruebas de usabilidad con usuarios finales se realizaron con el objetivo de verificar el funcionamiento del sistema de información cuando es utilizado por personas externas a la Cooperativa Integral de Transporte Andino TransAndino, es decir, usuarios civiles sin vinculación directa con los procesos administrativos ni experiencia previa con la plataforma.

En esta etapa participaron tres usuarios civiles, correspondientes a dos mujeres y un usuario masculino, con rangos de edad diferentes, quienes realizaron las pruebas simulando el uso del sistema desde un perfil administrativo. El propósito de este enfoque fue evaluar si el sistema puede ser comprendido y utilizado correctamente por usuarios no expertos, permitiendo validar la claridad de la interfaz, la lógica de los flujos de navegación y la correcta ejecución de las funcionalidades, independientemente del conocimiento previo del contexto organizacional.

Los participantes ejecutaron un conjunto de tareas previamente definidas, similares a las realizadas por el usuario administrador, tales como el registro y consulta de información, la edición de datos y la navegación entre los distintos módulos del sistema. La ejecución de estas actividades permitió observar el comportamiento de los usuarios frente a las funcionalidades principales y detectar posibles dificultades en la interacción con la plataforma.

Durante las pruebas se identificaron algunos aspectos de mejora relacionados con la disposición de ciertos elementos visuales, la claridad de algunos mensajes del sistema y la secuencia de acciones en determinados formularios. A partir de esta retroalimentación se realizaron ajustes orientados a mejorar la comprensión del sistema, simplificar los flujos de uso y reforzar la experiencia de usuario.

En términos generales, las pruebas con usuarios finales permitieron corroborar que el sistema puede ser utilizado de manera adecuada por personas externas a la cooperativa, cumpliendo correctamente con sus funcionalidades principales y sirviendo como validación complementaria a la prueba realizada con el usuario administrador.

Análisis de Resultados

Análisis de los Resultados de las Pruebas de Usabilidad

El análisis de resultados de las pruebas de usabilidad se realizó a partir de los criterios establecidos por la norma **ISO 9241-11**, considerando las dimensiones de **eficacia, eficiencia y satisfacción del usuario**. Este análisis permitió evaluar el desempeño del sistema tanto desde el rol administrativo como desde la perspectiva de usuarios finales externos.

En términos de **eficacia**, los resultados evidencian que el usuario administrador logró completar el 100 % de las tareas propuestas (Figura 38), mientras que los usuarios finales alcanzaron un alto nivel de cumplimiento, con ligeras variaciones en algunas actividades. Estos resultados confirman que el sistema cumple funcionalmente con los procesos definidos y permite la ejecución correcta de las acciones principales.

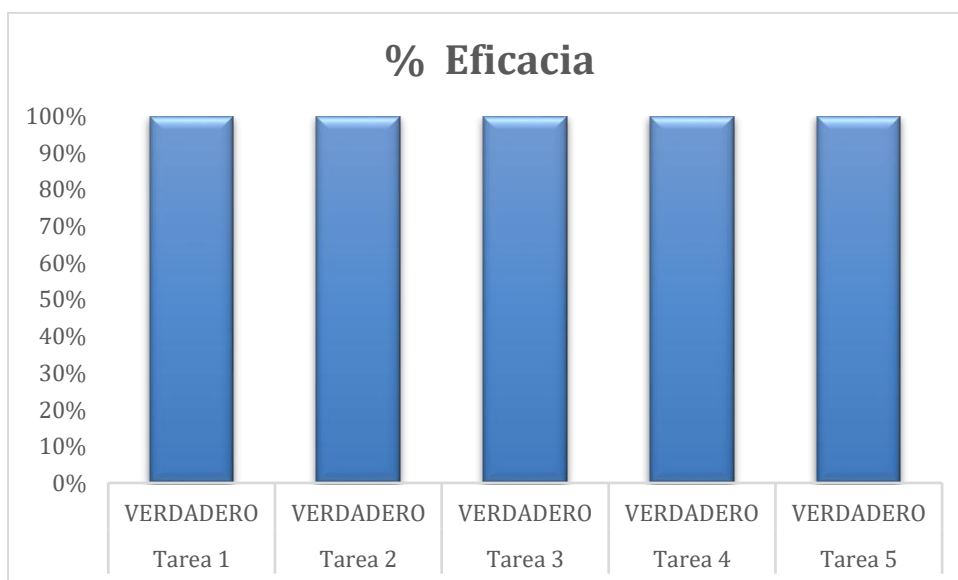


Figura 38- Gráfica de Eficacia en Test de Usuario Administrador. Fuente – Propio.

Respecto a la **eficiencia**, los tiempos de ejecución registrados muestran un comportamiento estable para el usuario administrador (Figura 39) y una mayor variabilidad en el caso de los usuarios finales (Figura 40 y Figura 41), lo cual es coherente con las diferencias en experiencia y familiaridad con el sistema. Aun así, los tiempos obtenidos se consideran adecuados, lo que demuestra que la plataforma permite realizar las tareas sin requerir esfuerzos excesivos.

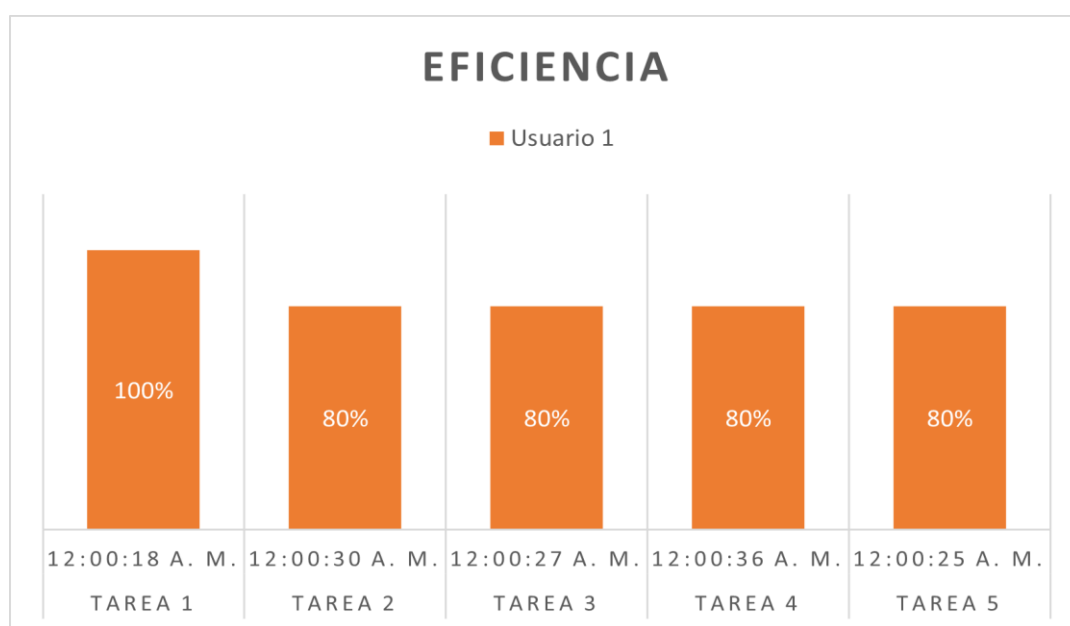


Figura 39- Gráfica de resultados de Eficiencia en Test de Usuario Administrador. Fuente – Propio.

Tempo(minuto)	Usuario 1	Usuario 2	Usuario 3	Usuario 1	Usuario 2	Usuario 3
Tarea 1	12:00:20 a. m.	12:00:17 a. m.	12:00:29 a. m.	80%	100%	80%
Tarea 2	12:00:13 a. m.	12:00:10 a. m.	12:00:40 a. m.	100%	100%	80%
Tarea 3	12:00:35 a. m.	12:00:51 a. m.	12:01:40 a. m.	80%	60%	20%
Tarea 4	12:00:28 a. m.	12:00:28 a. m.	12:01:46 a. m.	80%	60%	80%
Tarea 5	12:00:24 a. m.	12:00:21 a. m.	12:01:47 a. m.	80%	80%	60%

Figura 40 - Tabla de tiempos con usuarios finales . Fuente - Propio.

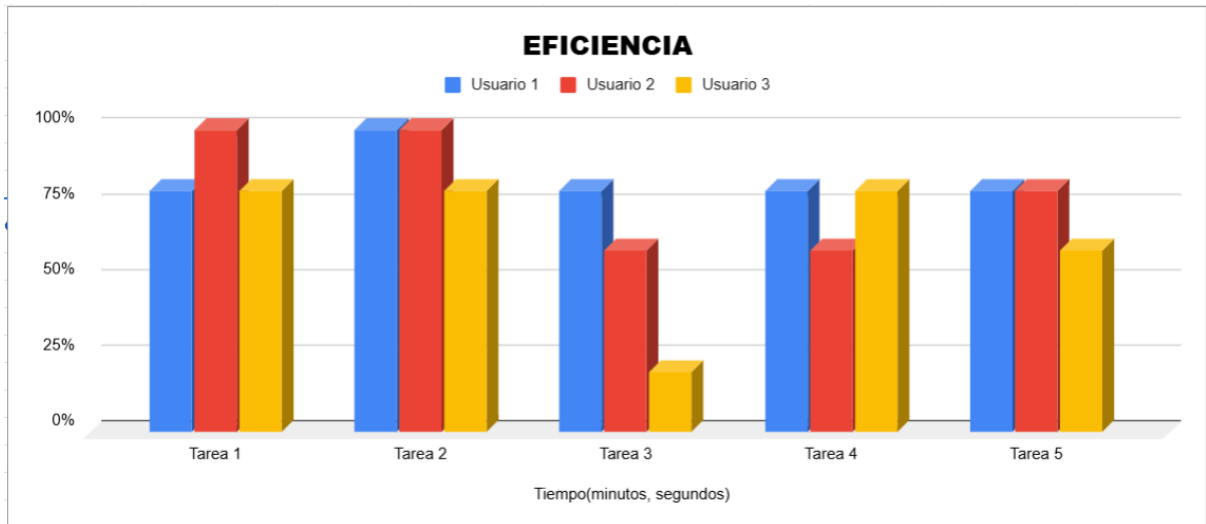


Figura 41 – Gráfico de Eficiencia Test con Usuarios Finales. Fuente -Propio.

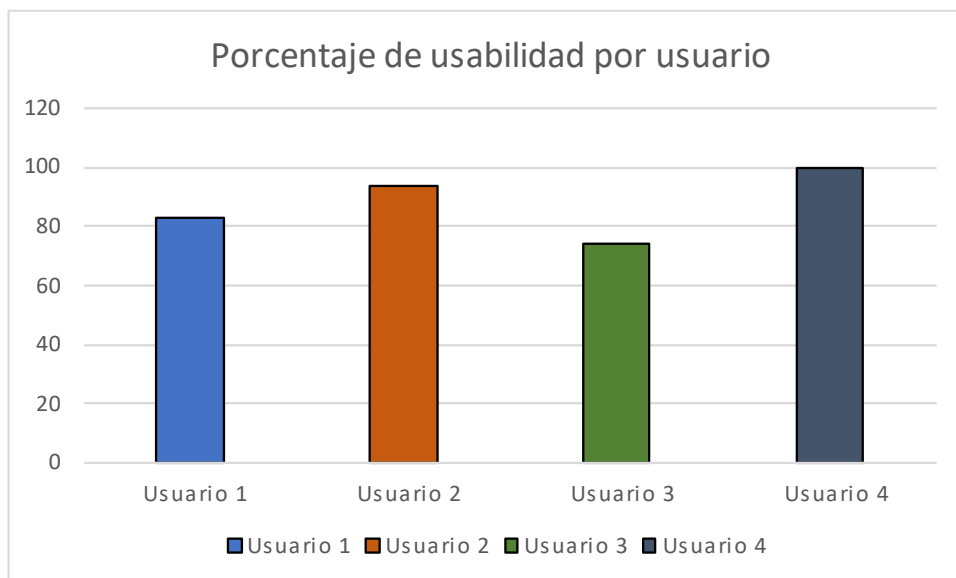


Figura 42– Satisfacción de usuario en general con todos los usuarios. Fuente -Propio.

En cuanto a la Satisfacción de Usuario la gráfica plasma un rango entre el 78% y 100% asegurando que la experiencia usando el sistema no genera inconformidad o emociones de frustración al realizar actividades dentro de los módulos.

Los resultados de eficacia y eficiencia se presentan mediante gráficas, las cuales facilitan la visualización comparativa del desempeño de cada grupo de usuarios y permiten identificar de manera clara el comportamiento del sistema

durante las pruebas. Estos resultados sirvieron como base para la identificación de ajustes menores orientados a mejorar la experiencia de uso y fortalecer la coherencia de la interacción con el sistema.

Análisis de funcionamiento del sistema

Uno de los principales beneficios obtenidos fue la centralización de la información administrativa. Antes del desarrollo del sistema, los datos se encontraban dispersos en hojas de cálculo y registros manuales, lo que dificulta su control y actualización. Con la implementación del sistema, la información de vehículos, conductores, asociados y documentación se gestiona desde una única plataforma, lo que reduce la duplicidad de datos y minimiza el riesgo de errores humanos. De acuerdo con Stair y Reynolds (2017), la centralización de la información contribuye a mejorar la eficiencia operativa y la confiabilidad de los datos dentro de las organizaciones.

Del mismo modo, la solución tecnológica aporta una mejora notable en el rendimiento del personal administrativo. Al automatizar tareas repetitivas y simplificar los procesos de consulta y registro, se redujeron los tiempos de gestión y se optimizó el uso de los recursos humanos disponibles. Según Laudon y Laudon (2020), los sistemas de información bien implementados permiten a las organizaciones enfocar sus esfuerzos en actividades de mayor valor estratégico, en lugar de destinar tiempo a procesos manuales.

Desde la perspectiva organizacional, el sistema representa un avance significativo hacia la transformación digital de la cooperativa. La adopción de esta

herramienta tecnológica no solo moderniza la gestión administrativa, sino que también sienta las bases para futuras mejoras, como la integración con otros sistemas o la incorporación de nuevas funcionalidades. Vial (2019) destaca que la transformación digital fortalece la competitividad institucional y facilita la adaptación a entornos cada vez más exigentes y dinámicos.

Durante el proyecto, los datos analizados corresponden principalmente a la información administrativa relacionada con vehículos, conductores, asociados, documentación obligatoria y mantenimientos. Estos datos fueron estructurados y almacenados en una base de datos relacional, lo que facilitó su consulta, actualización y análisis de manera ordenada y consistente. Según Elmasri y Navathe (2015), el uso de bases de datos estructuradas permite garantizar la integridad de la información y optimizar los procesos de análisis en sistemas de información organizacionales.

El sistema desarrollado permite realizar análisis que apoyan la gestión administrativa, tales como la identificación de documentos próximos a vencer, el seguimiento de mantenimientos y la verificación del estado operativo de los vehículos. Este tipo de análisis contribuye a la reducción de riesgos operativos y legales, al permitir una supervisión más efectiva de los procesos críticos de la cooperativa. De acuerdo con Laudon y Laudon (2020), el análisis de datos operativos proporciona una base confiable para mejorar la eficiencia y el control organizacional.

Desde una perspectiva estratégica, el análisis de datos obtenido a través del sistema de información sienta las bases para futuros procesos de mejora continua y transformación digital. Tal como señala Davenport (2014), las organizaciones que

aprovechan sus datos de manera sistemática logran un mayor nivel de competitividad y sostenibilidad en entornos dinámicos y regulados.

La visualización y soporte de las validaciones se encuentra complementada en el ANEXO C que corresponde a las pruebas con usuarios finales, acuerdos de confidencialidad y cuestionarios post-text, ANEXO D de las pruebas del backend y finalmente ANEXO E pruebas de Usabilidad con el Usuario administrador que contiene acuerdo de confidencialidad, captura de imágenes del cumplimiento de tareas, pretest, test y posttest.

Conclusiones

El desarrollo del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino permitió evidenciar la importancia de la tecnología como un elemento estratégico para fortalecer la gestión administrativa y operativa de las organizaciones del sector transporte. A partir del análisis de las necesidades reales de la cooperativa, fue posible diseñar e implementar una solución tecnológica alineada con sus procesos, contribuyendo a la modernización de la gestión de la información.

El proyecto demostró que la digitalización de los procesos administrativos favorece la centralización, integridad y trazabilidad de los datos, reduciendo los riesgos asociados al manejo manual de la información. La implementación de una base de datos relacional y de un sistema de información estructurado permitió mejorar el control sobre la documentación, los mantenimientos y la información de los vehículos y asociados, fortaleciendo la toma de decisiones basada en datos confiables y actualizados.

En este sentido, la adopción de metodologías de desarrollo de software permitió una construcción progresiva y controlada del sistema, facilitando la identificación temprana de ajustes y garantizando el cumplimiento de los requisitos establecidos. La integración entre el frontend, el backend y la base de datos evidenció la viabilidad técnica del proyecto y aseguró un funcionamiento coherente y estable de la plataforma.

Desde una perspectiva organizacional, el sistema desarrollado representa un avance significativo hacia la transformación digital de la cooperativa. Más allá de la implementación tecnológica, el proyecto contribuyó a generar una mayor conciencia sobre la importancia del uso adecuado de la información y del aprovechamiento de herramientas digitales para mejorar la eficiencia institucional. Tal como señalan Laudon y Laudon (2020), los sistemas de información bien implementados generan valor cuando se integran de manera efectiva en los procesos organizacionales.

En suma, el sistema de información desarrollado cumple con los objetivos planteados y se consolida como una solución funcional, confiable y adaptable a las necesidades actuales de la Cooperativa Integral de Transporte Andino TransAndino, sentando las bases para futuras mejoras tecnológicas y fortaleciendo su capacidad de gestión administrativa.

Recomendaciones

A partir del desarrollo e implementación del sistema de información para la Cooperativa Integral de Transporte Andino TransAndino, se formulan una serie de recomendaciones orientadas a fortalecer el uso de la solución tecnológica, garantizar su sostenibilidad en el tiempo y potenciar su impacto dentro de la organización. Estas

recomendaciones se fundamentan en los resultados obtenidos durante el proyecto y en buenas prácticas relacionadas con la gestión de sistemas de información.

En primer lugar, se sugiere establecer políticas internas claras para el manejo y actualización de la información. La calidad de los datos almacenados en el sistema depende en gran medida de la disciplina administrativa y del compromiso del personal encargado de su gestión. Definir responsables, frecuencias de actualización y protocolos de validación permitirá mantener la integridad y confiabilidad de la información a lo largo del tiempo.

Otra recomendación relevante es realizar mantenimientos periódicos al sistema, tanto a nivel técnico como funcional. Esto incluye la actualización de la base de datos, la revisión de la seguridad, la optimización del rendimiento y la incorporación de mejoras conforme evolucionen las necesidades de la cooperativa. Pressman y Maxim (2020) señalan que el mantenimiento del software es una etapa esencial para garantizar su continuidad operativa y su adaptación a nuevos requerimientos.

De igual manera, se recomienda considerar la ampliación futura del sistema mediante la integración con otras herramientas tecnológicas, como módulos de reportes avanzados, notificaciones automatizadas o integración con sistemas externos del sector transporte. Estas mejoras permitirían fortalecer la transformación digital de la cooperativa y ampliar el alcance de la solución desarrollada.

Para concluir, se sugiere realizar evaluaciones periódicas del desempeño del sistema, apoyadas en la retroalimentación de los usuarios finales. Este proceso permitirá identificar oportunidades de mejora, corregir posibles falencias y asegurar que la solución tecnológica continúe alineada con los objetivos organizacionales de la Cooperativa Integral de Transporte Andino TransAndino.

Bibliografía

Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business*. Blue Hole Press.

Barrera, J. R. A., Torres, Y. E. S., & Alarcón, L. A. V. (2024). La Tecnología y su Impacto en la Gestión de Procesos y Estrategias de Automatización. *Ciencia Latina Revista Científica Multidisciplinar*, 8(4), 6198–6221.
https://doi.org/10.37811/cl_rcm.v8i4.12822

CEPAL.(2022). *La digitalización del transporte y la logística en América Latina y el Caribe*. Naciones Unidas.
<https://www.cepal.org/es/publicaciones/digitalizacion-transporte-logistica>

Crespo Postigo, J. (2025). *Digitalización y gestión tecnológica en el sector transporte*. Editorial Académica Española.

Davenport, T. H. (2014). *Big data at work: Dispelling the myths, uncovering the opportunities*. Harvard Business Review Press.

Dini, M., Gligo, N., & Patiño, A. (2021). *Transformación digital de las pymes en América Latina*. Comisión Económica para América Latina y el Caribe (CEPAL).

Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of database systems* (7th ed.). Pearson.

Express.js Foundation. (2023). *Express – Node.js web application framework*.
<https://expressjs.com>

Facebook Open Source. (2023). *React: A JavaScript library for building user interfaces*. <https://react.dev>

Ferrente,J.,& Rodríguez,D.(2022). *La telemetría como herramienta de IoT en la gestión de seguridad de flotas vehiculares*. Revista de investigación en Ingeniería y Logística.

<https://periodicos.newsciencepubl.com/arace/article/view/1414>

International Organization for Standardization. (2018). *ISO/IEC 27001: Information security management systems*. ISO.

Koontz, H., & Weihrich, H. (2012). *Administración: Una perspectiva global y empresarial* (13.^a ed.). McGraw-Hill.

Koontz, H., & Weihrich, H. (2012). *Essentials of management* (9th ed.). McGraw-Hill Education.

Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing the digital firm* (16th ed.). Pearson.

Ministerio de Transporte. (2020). *Política y lineamientos para la modernización del sector transporte en Colombia*. Gobierno de Colombia.

Ministerio de Transporte. (2023). *Normatividad del transporte terrestre y control documental vehicular*. Gobierno de Colombia.

Ministerio de Transporte. (2023). Normatividad del sector transporte en Colombia.

<https://www.mintransporte.gov.co>

Nielsen, J. (2012). *Usability engineering*. Morgan Kaufmann.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.

Stair, R. M., & Reynolds, G. W. (2017). *Principles of information systems* (13th ed.). Cengage Learning.

Stair, R. M., & Reynolds, G. W. (2017). *Principles of information systems* (13th ed.). Cengage Learning..

OpenJS Foundation. (2023). *Node.js documentation*. <https://nodejs.org>

Oracle Corporation. (2023). *MySQL documentation*. <https://dev.mysql.com/doc/>

Perez, P., & Enrique, J. (2024). *Optimización de tareas rutinarias y mejora de la toma de decisiones en el senior management mediante ciencia de datos y herramientas de inteligencia artificial accesibles*.
<http://repository.unad.edu.co/handle/10596/69257>

Pressman, R. S. (2015). *Ingeniería de software: Un enfoque práctico* (7.^a ed.). McGraw-Hill Education.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Vial, G. (2019). Understanding digital transformation: A review and a research agenda. *Journal of Strategic Information Systems*, 28(2), 118–144.
<https://doi.org/10.1016/j.jsis.2019.01.003>

Westerman, G., Bonnet, D., & McAfee, A. (2014). *Leading digital: Turning technology into business transformation*. Harvard Business Review Press.

Vite Team. (2023). *Vite: Next generation frontend tooling*. <https://vitejs.dev>

Westerman, G., Bonnet, D., & McAfee, A. (2014). *Leading digital: Turning technology into business transformation*. Harvard Business Review Press.