PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Facultad De Hábitat, Infraestructura Y Creatividad

Carrera De: Sistemas De Información



TRABAJO DE INTEGRACIÓN CURRICULAR

Desarrollo de una aplicación web prototipo para gestión del servicio de transporte privado en el recinto Julio Moreno Espinosa de la provincia de Santo Domingo de los Tsáchilas

AUTOR:

MATEO CEVALLOS CEVALLOS

QUITO DM, JUNIO DE 2025

DEDICATORIA

Este trabajo se lo dedico a mis hermanos quienes han sido mi ejemplo, especialmente a mi hermana Stefany, quien ha sido como una segunda madre y un pilar fundamental junto a mi madre, brindándome su apoyo incondicional y constante motivación. Extiendo también esta dedicatoria con cariño y gratitud a toda mi familia, cuyo respaldo permanente ha sido indispensable en la culminación de esta etapa.

AGRADECIMIENTOS

Expreso mi profundo agradecimiento al Ingeniero Fabián de la Cruz, cuyas enseñanzas y dedicación fueron clave para alcanzar los objetivos propuestos en este proyecto. Asimismo, a la Pontificia Universidad Católica, institución que me ha brindado las herramientas académicas necesarias para mi desarrollo personal y profesional. Finalmente, agradezco a todos los docentes que me han guiado en el transcurso de la carrera, y por impartir su conocimiento con dedicación y paciencia.

RESUMEN

El presente proyecto de titulación se enfoca en el desarrollo de un prototipo de aplicación web para mejorar la gestión del servicio de transporte privado en el recinto Julio Moreno Espinosa, provincia de Santo Domingo de los Tsáchilas. El objetivo central fue automatizar procesos tradicionalmente manuales, como la gestión de conductores, vehículos, rutas y manejo de incidencias, utilizando tecnologías modernas como Next.js y PostgreSQL.

Se aplicó la metodología ágil Crystal Clear, seleccionada por su capacidad para brindar retroalimentación constante, desarrollo incremental y adaptación rápida a cambios solicitados por usuarios expertos. Esta metodología permitió validar cada etapa del proyecto de manera efectiva.

Se concluye que la adopción de sistemas centralizados representa una solución que permite enfrentar los desafíos cotidianos de gestión en empresas del sector transporte privado.
Finalmente, se proponen recomendaciones clave para garantizar el mantenimiento, expansión y aprovechamiento óptimo de la solución desarrollada, gestión dinámica de parámetros configurables, además se recomienda tomar nota de retroalimentación valiosa del usuario final.

TABLA DE CONTENIDO

CAPÍTULO 1: DEFINICIÓN DEL TRABAJO DE TITULACIÓN 1

1.1. TEMA 1

1.2. JUSTIFICACIÓN 1

1.3. PLANTEAMIENTO DEL PROBLEMA 3

1.4. OBJETIVOS 5

1.4.1. Objetivo General 5

1.4.2. Objetivos Específicos: 5

1.5. ALCANCE 5

1.5.1. Excepciones: 6

CAPÍTULO 2. MARCO TEÓRICO 7

2.1. MARCO CONCEPTUAL 7

2.1.1. Aplicación web 7

2.1.2. Base de datos 7

2.1.3. Frontend 7

2.1.4. Transporte privado 8

2.2. Metodologías de desarrollo de software 8

2.2.1. ¿Qué son las metodologías de desarrollo de software? 8

2.2.2. Comparativa entre metodologías tradicionales y ágiles 8

2.2.2.1. CRITERIO 1: ENFOQUE 9

2.2.2.2. CRITERIO 2: Flexibilidad al Cambio 9

2.2.2.3. CRITERIO 3: Participación del cliente 10

2.2.2.4. CRITERIO 4: Documentación 10

2.2.2.5. CRITERIO 5: Entrega de producto 11

2.2.2.6. CRITERIO 6: Gestión de riesgos 12

2.2.2.7. CRITERIO 7: Ejemplos 13

2.2.2.7.1. Ejemplos de metodologías tradicionales 13

Cascada (Pressman & Maxim, 2020): 13

Modelo en V (Sommerville, 2016): 13

Modelo incremental (Pressman & Maxim, 2020): 13

2.2.2.7.2. Ejemplos de metodologías ágiles 14

Scrum:..... 14

Kanban:..... 14

2.2.2.7.3. Extreme Programming (XP) : 14

2.3. Selección de la metodología más adecuada para el proyecto..... 15

2.3.1. Metodología seleccionada..... 15

2.3.1.1. ¿Por qué Crystal Clear es adecuada para este proyecto?..... 15

2.3.2. Ciclo de desarrollo según Crystal Clear..... 16

2.3.2.1. Preparación del proyecto 16

2.3.2.2. Planificación de entregas..... 16

2.3.2.3. Desarrollo e integración..... 16

2.3.2.4. Pruebas y validación..... 16

2.3.2.5. Retroalimentación y mejora continua..... 16

2.3.3. Entregables generados durante el desarrollo..... 17

2.3.3.1. Versiones funcionales del sistema..... 17

2.3.3.2. Documentación técnica esencial..... 17

2.3.3.3. Resultados de pruebas 17

2.3.3.4. Retroalimentación de usuarios..... 17

2.4. Herramientas de desarrollo 17

2.4.1. PostgreSQL 17

2.4.2. Next.js 18

2.4.3. PlantUML 19

2.4.4. Visual Studio Code 19

2.4.5. Justificación comparativa de herramientas seleccionadas 20

CAPÍTULO 3. Análisis de requisitos 23

3.1. Requerimientos Funcionales 23

3.1.1. Listado de Requerimientos Funcionales 23

3.1.2. Caso de uso de nivel general 24

3.1.3. Caso de uso a detalle F1: Control de usuarios 25

3.1.3.1.1. F1.1: Iniciar sesión 25

3.1.4. Caso de uso a detalle F2: Gestión de supervisores 26

3.1.4.1. F2.1: Insertar supervisor 26

3.1.4.2. F2.2: Eliminar supervisor 27

3.1.4.3. F2.3: Modificar supervisor 28

3.1.4.4. F2.4: Consultar supervisor 28

3.1.4.4.1. F2.4.1: Consulta general 28

3.1.4.4.2. F2.4.2: Consulta por parámetros 29

3.1.5. Caso de uso a detalle F3: Gestión de propietarios 29

3.1.5.1. F3.1: Insertar propietario 30

3.1.5.2. F3.2: Eliminar propietario 30

3.1.5.3. F3.3: Modificar propietario 31

3.1.5.4. F3.4: Consultar propietario 31

3.1.5.4.1. F3.4.1: Consulta general 32

3.1.5.4.2. F3.4.2: Consulta por parámetros 32

3.1.6. Caso de uso a detalle F4: Gestión de personal 33

3.1.6.1. F4.1: Insertar personal 33

3.1.6.2. F4.2: Eliminar personal 34

F4.3: Modificar personal 34

F4.4: Consultar personal 35

F4.4.1: Consulta general 35

F4.4.2: Consulta por parámetros 35

3.1.7. Caso de uso a detalle F5: Gestión de vehículos 36

3.1.7.1. F5.1: Insertar vehículo 36

3.1.7.2. F5.2: Eliminar vehículo 37

3.1.7.3. F5.3: Modificar vehículo 37

3.1.7.4. F5.4: Consultar vehículo 38

3.1.7.4.1. F5.4.1: Consulta general 38

3.1.7.4.2. F5.4.2: Consulta por parámetros 38

3.1.8. Caso de uso a detalle F6: Gestión de rutas 39

3.1.8.1. F6.1: Insertar ruta 39

3.1.8.2. F6.2: Eliminar ruta 40

3.1.8.3. F6.3: Modificar ruta 41

3.1.8.4. F6.4: Consultar ruta 41

3.1.8.4.1. F6.4.1: Consulta general 41

3.1.8.4.2. F6.4.2: Consulta por parámetros 42

3.1.9. Caso de uso a detalle F7: Gestión de horarios 42

3.1.9.1. F7.1: Insertar horario 43

3.1.9.2. F7.2: Eliminar horario 43

3.1.9.3. F7.3: Modificar horario 44

3.1.9.4. F7.4: Consultar horario 44

3.1.9.4.1. F7.4.1: Consulta general 45

3.1.9.4.2. F7.4.2: Consulta por parámetros 45

3.1.10. Caso de uso a detalle F8: Gestión de pasajeros 46

3.1.10.1. F8.1: Insertar pasajero 46

3.1.10.2. F8.2: Eliminar pasajero 47
3.1.10.3. F8.3: Modificar pasajero 47
3.1.10.4. F8.4: Consultar pasajero 48
3.1.10.4.1. F8.4.1: Consulta general 48
3.1.10.4.2. F8.4.2: Consulta por parámetros 48
3.1.11. Caso de uso a detalle F9: Gestión de recorridos 49
3.1.11.1. F9.1: Iniciar recorrido 49
3.1.11.2. F9.2: Registrar pasajero 49
3.1.11.3. F9.3: Terminar recorrido 50
3.1.11.4. F9.4: Consultar recorrido 50
3.1.11.4.1. F9.4.1: Consulta general 51
3.1.11.4.2. F9.4.2: Consulta por parámetros 51
3.1.12. Caso de uso a detalle F10: Gestión de incidentes 52
3.1.12.1. F10.1: Reportar incidente 52
3.1.12.2. F10.2: Consultar incidente 53
3.1.12.2.1. F10.2.1: Consulta general 53
3.1.12.2.2. F10.2.2: Consulta por parámetros ..... **¡Error! Marcador no definido.**
3.2. Requerimientos No Funcionales 54
CAPÍTULO 4. Diseño y desarrollo de la aplicación web 55

4.1. Introducción 55

4.2. Modelado de base de datos 55

4.2.1. Modelo conceptual 55

4.2.2. Modelo lógico 56

4.2.3. Modelo físico 57

4.3. Desarrollo de la aplicación web 58

4.3.1. Cronograma de hitos 58

4.3.2. Notas por hito 58

4.3.2.1. Hito 1 58

4.3.2.2. Hito 2 59

4.3.2.3. Hito 3 59

4.3.2.4. Hito 4 59

4.3.2.5. Hito 5 59

4.3.3. Diseño de la interfaz 60

4.4. Pruebas unitarias del sistema 69

4.4.1. Hito 1 69
4.4.1.1. F1 Control de Usuarios 69
4.4.1.2. Valoración general del hito 71
4.4.2. Hito 2 72
4.4.2.1. F2 Gestión de supervisores 72
4.4.2.2. F3 Gestión de propietarios 74
4.4.2.3. Valoración general del hito 76
4.4.3. Hito 3 77
4.4.3.1. F4 Gestión de personal 77
4.4.3.2. F5 Gestión de vehículos 79
4.4.3.3. Valoración general del hito 81
4.4.4. Hito 4 82
4.4.4.1. F6 Gestión de rutas 82
4.4.4.2. F7 Gestión de horarios 84
4.4.4.3. Valoración general del hito 86
4.4.5. HITO 5 87
4.4.5.1. F8 Gestión de pasajeros 87
4.4.5.2. F9 Gestión de recorridos 89
4.4.5.3. F10 Gestión de incidentes 91

4.4.5.4. Valoración general del hito..... 91

4.5. Aplicación de la metodología seleccionada 92

4.5.1. Software funcional entregable..... 92

4.5.1.1. Hitos definidos:..... 92

4.5.2. Talleres de Reflexión 92

4.5.2.1. Hito 1 92

4.5.2.1.1. ¿Qué salió bien? 92

4.5.2.1.2. ¿Qué podría mejorar? 93

4.5.2.1.3. ¿Qué haré diferente en la siguiente etapa? 93

4.5.2.1.4. Comentarios adicionales: 93

4.5.2.2. Hito 2 93

4.5.2.2.1. ¿Qué salió bien? 93

4.5.2.2.2. ¿Qué podría mejorar? 93

4.5.2.2.3. ¿Qué haré diferente en la siguiente etapa? 93

4.5.2.2.4. Comentarios adicionales: 93

4.5.2.3. Hito 3 94

4.5.2.3.1. ¿Qué salió bien? 94

4.5.2.3.2. ¿Qué podría mejorar? 94

4.5.2.3.3. ¿Qué haré diferente en la siguiente etapa? 94

4.5.2.3.4. Comentarios adicionales: 94

4.5.2.4. Hito 4 94
4.5.2.4.1. ¿Qué salió bien? 94
4.5.2.4.2. ¿Qué podría mejorar? 94
4.5.2.4.3. ¿Qué haré diferente en la siguiente etapa? 95
4.5.2.4.4. Comentarios adicionales: 95
4.5.2.5. Hito 5 95
4.5.2.5.1. ¿Qué salió bien? 95
4.5.2.5.2. ¿Qué podría mejorar? 95
4.5.2.5.3. ¿Qué haré diferente en la siguiente etapa? 95
4.5.2.5.4. Comentarios adicionales: 95
4.5.3. Planificación liviana 96
4.5.4. Lista de riesgos conocidos 97
4.5.5. Roles asumidos durante el proyecto 97
4.5.6. Propiedades esenciales aplicadas 98
4.5.7. Validación con el usuario experto 98
4.5.7.1. ACTA DE REUNIÓN LEVANTAMIENTO DE REQUISITOS 99
4.5.7.1.1. PARTICIPANTES: 99
4.5.7.1.2. OBJETIVO DE LA REUNIÓN 99

4.5.7.1.3. TEMAS TRATADOS 99
4.5.7.1.4. ACUERDOS Y COMPROMISOS 99
4.5.7.1.5. PUNTOS PENDIENTES 99
4.5.7.2. ACTA DE REUNIÓN HITO 1 100
4.5.7.2.1. OBJETIVO DE LA REUNIÓN 100
4.5.7.2.2. TEMAS TRATADOS: 100
4.5.7.2.3. ACUERDOS: 100
4.5.7.3. ACTA DE REUNIÓN HITO 2 100
4.5.7.4. ACTA DE REUNIÓN HITO 3 101
4.5.7.4.1. TEMAS TRATADOS: 101
4.5.7.4.2. ACUERDOS: 101
4.5.7.5. ACTA DE REUNIÓN HITO 4 101
4.5.7.5.1. TEMAS TRATADOS: 101
4.5.7.5.2. ACUERDOS: 101
4.5.7.6. ACTA DE REUNIÓN HITO 5 102
4.5.7.6.1. TEMAS TRATADOS: 102
4.5.7.6.2. ACUERDOS: 102
4.5.8. Acta de aceptación del prototipo. 103
CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES 104

5.1. CONCLUSIONES 104

5.2. RECOMENDACIONES 104

BIBLIOGRAFÍA 106

CAPÍTULO 1: DEFINICIÓN DEL TRABAJO DE TITULACIÓN

1.1. TEMA

Desarrollo de una aplicación web prototipo para gestión del servicio de transporte privado en el recinto Julio Moreno de la provincia de Santo Domingo

1.2. JUSTIFICACIÓN

Las personas particulares e instituciones privadas requieren de movilización segura a diario. Entre los diversos grupos que requieren estos servicios, destacan por ejemplo los estudiantes, quienes representan una gran proporción de usuarios se trasladan diariamente a sus destinos. En la provincia Santo Domingo de los Tsáchilas se coordinan a diario el traslado de 139 632 estudiantes distribuidos en más de 2 500 instituciones educativas esta cifra que sitúa a la Zona 4 entre las de mayor presión operativa del país, lo que vuelve crítica la administración eficiente de conductores, vehículos y rutas (Estadística Educativa – Volumen 4, 2023; Normativa de transporte escolar para las instituciones educativas del Sistema Nacional de Educación, 2022).

Existen una gran cantidad de empresas que ofrecen el servicio de transporte privado realizando la administración de esta actividad de forma manual. La Empresa Pública Municipal de Transporte (EPMT-SD) mantiene todavía un esquema 100% presencial. El representante legal de cada operadora debe llevar más de veinte documentos físicos y hojas Excel para renovar sus permisos de operación, trámite que en 2022 implicó 17 renovaciones, entre ellas la de la Compañía de Transporte Escolar e Institucional Bonanza S.A., evidenciando demoras y sobrecarga administrativa derivadas de procesos manuales (Rendición de cuentas 2022, 2023; Resolución No. EPMT-SD-GG-HFS-RPO-2022-090: Renovación del permiso de operación de la Compañía de Transporte Escolar e Institucional Bonanza S.A., 2022).

Las empresas que ofrecen servicio de transporte a través de procesos manuales a menudo tienen problemas para administrar el recurso humano y vehículos disponibles; retrasos injustificados e incidentes que son parte del día a día de los santodomingueños.

Diversos trabajos resaltan la necesidad de incorporar tecnología para modernizar la gestión de flotas de transporte. Por ejemplo, Gallego desarrolló un "Sistema de Control y Gestión de Transporte Privado" dirigido a pequeñas y medianas empresas, permitiendo una comunicación entre conductores y coordinadores mediante dispositivos móviles (Gallego Rendón, 2016). Asimismo, una tesis realizada en Perú evidenció que un sistema de gestión de transporte reduce errores humanos y mejora la asignación de recursos, disminuyendo así costos por correcciones innecesarias (Cornejo Nava, 2023).

Si bien Quito, como capital del Ecuador, cuenta con una infraestructura tecnológica y de transporte más desarrollada, tiene mayores alternativas para la gestión del transporte, tanto público como privado, incluyendo aplicaciones móviles consolidadas como Uber, Cabify o plataformas de movilidad institucional.

En cambio, Santo Domingo de los Tsáchilas, y en particular el recinto Julio Moreno, enfrenta mayores desafíos en cuanto a conectividad, acceso a servicios de transporte organizados y tecnología aplicada a la movilidad. Esta realidad hace que Santo Domingo represente un entorno con una necesidad más urgente y concreta de soluciones tecnológicas que puedan mejorar la calidad del servicio de transporte privado para la comunidad educativa o institucional.

Por lo tanto, el presente trabajo evidencia que la gestión del transporte privado enfrenta grandes desafíos cuando se realiza de forma manual y tiene como objetivo superar las limitaciones del uso de métodos manuales tradicionales mediante la integración de módulos funcionales que permitan: centralizar la gestión de conductores, vehículos y rutas; facilitar la supervisión y la generación de reportes, además de mejorar la comunicación entre conductor y supervisor, dando la posibilidad de reportar incidentes en ruta. Para lograrlo, se toma como caso de estudio a la "Empresa de Gestión de Transporte Privado Julio Moreno".

Ahora bien, para mejorar los procesos manuales de gestión de transporte privado, se desarrollará una aplicación web que promueve el uso de la tecnología en este tipo de procesos y que además busca mejorar el proceso de gestión interna del servicio en la empresa Julio Moreno, con el fin de brindar un servicio seguro y ágil con los controles adecuados de incidencias que podrían suceder en la prestación del servicio de transporte privado.

1.3. PLANTEAMIENTO DEL PROBLEMA

En América Latina, la gestión del transporte privado es deficiente principalmente debido a que la gestión de este servicio se continúa realizando de forma manual, aunque en años recientes se han realizado iniciativas para modernizarla, estudios regionales indican que el sector de transporte es uno de los más atrasados en transformación digital: según el BID (Banco Interamericano de Desarrollo), el transporte terrestre es el sector con menos adopción tecnológica de la región. Un índice de digitalización de transportistas en Argentina (2024) mostró un promedio de apenas 2.42/5, r lo que significa que la gran mayoría de las empresas aún trabaja con métodos manuales como hojas de cálculo o registros en papel, y sólo un pequeño porcentaje (alrededor del 4%) utiliza sistemas especializados para automatizar procesos (Calatayud et al., 2021).

En el Ecuador, también existen problemas de administración de recursos, falta de comunicación entre conductores y supervisores, y la necesidad de realizar reportes rápidos de incidentes en ruta (Yaulema Rojas et al., 2024). La implementación de sistemas ha demostrado ser una solución efectiva para brindar servicios de transporte de calidad a los usuarios.

El recinto Julio Moreno Espinosa, ubicado a aproximadamente 20 minutos del centro urbano de Santo Domingo de los Tsáchilas, es una comunidad rural que, según el censo realizado en 2022, cuenta con una población de 816 habitantes, observándose un aumento considerable de población en años recientes (Gobierno Autónomo Descentralizado Municipal del cantón Santo Domingo, 2024).

La comunidad enfrenta desafíos significativos en cuanto a movilidad y acceso a servicios educativos. La Unidad Educativa "PIERA GRAZIA BRUCCOLERI" ofrece educación desde primer grado hasta décimo año de educación básica; sin embargo, su capacidad es limitada, contando con 310 estudiantes matriculados según datos oficiales proporcionados por la comunidad Maristas Norandina. Debido a que existen pocas instituciones educativas cercanas, muchos estudiantes deben trasladarse diariamente a instituciones ubicadas en el centro urbano.

El principal medio de transporte disponible es un servicio de autobús operado por la empresa San Jacinto, que cuenta con pocas unidades y una frecuencia de paso aproximada de una hora. Este servicio presenta problemas de puntualidad y confiabilidad, afectando la asistencia y puntualidad de usuarios en general. Además, el costo del pasaje es de 0,35 USD para estudiantes y 0,70 USD para adultos, representando una carga económica adicional para las familias, especialmente porque muchos requieren tomar al menos dos buses para llegar a sus destinos.

La inseguridad en las rutas es otra preocupación relevante. Según datos recientes proporcionados por la Policía Nacional, Santo Domingo ha experimentado un incremento significativo en los índices de delincuencia en los últimos años, generando temor en los padres respecto a la seguridad de sus hijos durante los desplazamientos diarios (La Hora, 2025).

A pesar de estos desafíos, Julio Moreno Espinosa cuenta con un importante potencial turístico y se encuentra en un proceso de urbanización. La comunidad dispone de espacios recreativos que atraen a los ciudadanos y visitantes los fines de semana, existe un creciente interés en mejorar la infraestructura y los servicios locales. Según la Municipalidad de Santo Domingo de los Tsáchilas, recientemente se ha entregado el plano de regulación urbana de este centro poblado, materializando así una expectativa muy esperada por todos sus habitantes (Julio Moreno Espinosa ya cuenta con plano de regulación urbana, 2025).

El servicio de transporte privado enfrenta desafíos en la supervisión y gestión efectiva de los recorridos debido a la falta de automatización de los procesos. Tal es el caso de la empresa de transporte privado Julio Moreno, que actualmente opera en la zona y enfrenta estos mismos desafíos, dependencia de métodos de comunicación informales, como WhatsApp, que dificultan la intervención rápida y organizada por parte de los supervisores.

Además, la falta de un sistema estructurado limita la capacidad de los supervisores para responder de manera ágil ante incidentes en ruta, tales como retrasos, desvíos o altercados menores, afectando la calidad del servicio. Y las empresas no suelen contar con un sistema centralizado para resolución de estas incidencias que no solo afectan a la eficiencia del

servicio, sino que también genera conflictos dentro de la empresa que luego escalan a problemas más graves, como irresponsabilidad por parte de los conductores, informes con datos erróneos e información que conlleva a la toma de malas decisiones por parte del gerente que perturban la operatividad de la empresa y posteriormente a la calidad del servicio que reciben los usuarios.

En este contexto, este trabajo presenta una solución a las dificultades que actualmente tiene este sector debido a los múltiples riesgos y desafíos que enfrenta el servicio de transporte privado.

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar de una aplicación web prototipo para gestión del servicio de transporte privado en el recinto Julio Moreno de la provincia de Santo Domingo de los Tsáchilas

1.4.2. Objetivos Específicos:

1. Identificar una metodología de desarrollo apropiada según las necesidades y características del proyecto.
2. Analizar los requisitos funcionales de la gestión del servicio de transporte privado, aplicando la metodología de desarrollo seleccionada.
3. Diseñar la interfaz de usuario y base de datos acorde a los requerimientos funcionales.
4. Desarrollar el prototipo y por ende realizar las pruebas de la aplicación web.

1.5. ALCANCE

El trabajo de titulación consiste en el desarrollo de un prototipo de aplicación web para la gestión del servicio de transporte privado , que incluye:

- Gestión de supervisores
- Gestión de conductores
- Gestión de vehículos
- Gestión de rutas
- Gestión de horarios

- Gestión de recorridos
- Gestión de incidentes

1.5.1. Excepciones:

En este prototipo, no se implementan funciones de geolocalización ni rastreo GPS en tiempo real, debido a su complejidad y costos asociados.

CAPÍTULO 2. MARCO TEÓRICO

2.1. MARCO CONCEPTUAL

2.1.1. Aplicación web

Una aplicación web es un software al que se accede a través de un navegador y que reside en un servidor, lo que permite la interacción remota sin instalación local. En el contexto del proyecto de gestión de transporte privado, utilizar aplicaciones web facilita la centralización del control de usuarios, vehículos, rutas e incidencias, simplificando tareas de actualización, acceso remoto y mantenimiento del sistema (Sommerville, 2016).

2.1.2. Base de datos

La base de datos es un sistema que almacena y administra datos estructurados para permitir consultas y actualizaciones eficientes. Este proyecto utiliza una base de datos relacional gestionada con PostgreSQL, que organiza información crítica como conductores, vehículos, rutas e incidencias, facilitando operaciones seguras, consistentes y simultáneas que mejoran la eficiencia operativa del transporte privado (*What is a database?*, 2024).

2.1.3. Frontend

El frontend representa la interfaz gráfica visible para el usuario, mediante la cual interactúa con el sistema utilizando elementos como botones, formularios y gráficos. En este proyecto, el uso de React y Next.js permite crear componentes modulares reutilizables que mejoran el rendimiento del sistema gracias al DOM virtual, optimizando así la experiencia del usuario y simplificando el mantenimiento del código.

Los marcos modernos simplifican la construcción de interfaces. React fragmenta la pantalla en componentes reutilizables y actualiza solo las partes que cambian gracias a su DOM virtual, lo que hace que la experiencia sea más rápida y el código más fácil de mantener(*React: Documentación oficial*, 2025). Por ejemplo, un menú o una tarjeta de producto se crean una sola vez y se reusan en varias páginas sin duplicar código.

2.1.4. Transporte privado

El transporte privado consiste en servicios específicos de movilidad contratados por organizaciones o instituciones para satisfacer necesidades particulares de sus usuarios, como estudiantes o empleados. En este proyecto, el transporte privado gestionado está orientado al recinto Julio Moreno, proporcionando soluciones personalizadas a la demanda específica del transporte escolar y optimizando el uso eficiente de recursos disponibles para garantizar calidad y puntualidad en el servicio prestado (López Gómez, 2022).

2.2. Metodologías de desarrollo de software

2.2.1. ¿Qué son las metodologías de desarrollo de software?

Las metodologías de desarrollo de software son un conjunto de principios, prácticas, técnicas y procedimientos que se aplican para para planificar, diseñar, desarrollar y mantener sistemas de software. Estas metodologías buscan optimizar los procesos de desarrollo, garantizar la calidad del producto final y cumplir con los plazos establecidos (Pressman & Maxim, 2020).

En el contexto del transporte privado, es crucial tener definidas rutas, horarios y una gestión eficiente de conductores y vehículos, la metodología elegida definirá cómo se abordan los desafíos de coordinación entre operadores y conductores por medio de la aplicación web. Una buena estrategia metodológica permite adaptar rápidamente el prototipo a necesidades específicas del sector, integrar sugerencias de los responsables del recinto y minimizar el tiempo de inactividad del servicio.

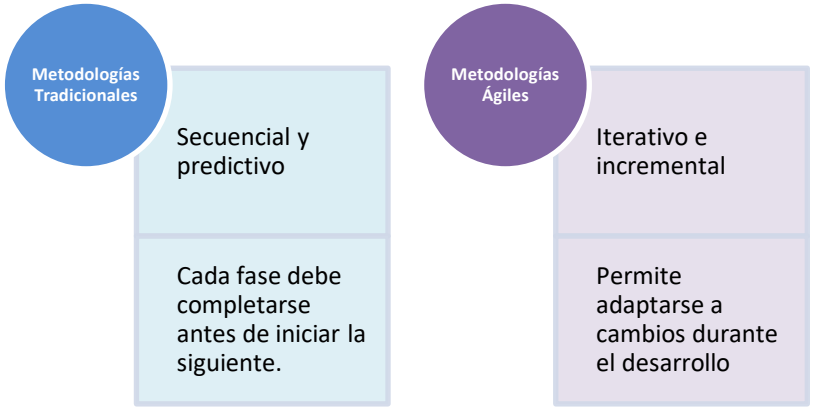
2.2.2. Comparativa entre metodologías tradicionales y ágiles

En las siguientes Figuras se muestran las diferencias fundamentales entre las metodologías tradicionales y ágiles en el desarrollo de software.

Según Pressman y Maxim, las metodologías tradicionales son apropiadas cuando los requisitos del proyecto están claramente definidos desde el inicio y es poco probable que cambien(Pressman & Maxim, 2020). En contraste, Schwaber y Sutherland enfatizan que Scrum, una metodología ágil, permite adaptarse rápidamente a los cambios y mejora la colaboración entre los equipos de desarrollo y los clientes(Schwaber & Sutherland, 2020).

2.2.2.1. CRITERIO 1: ENFOQUE

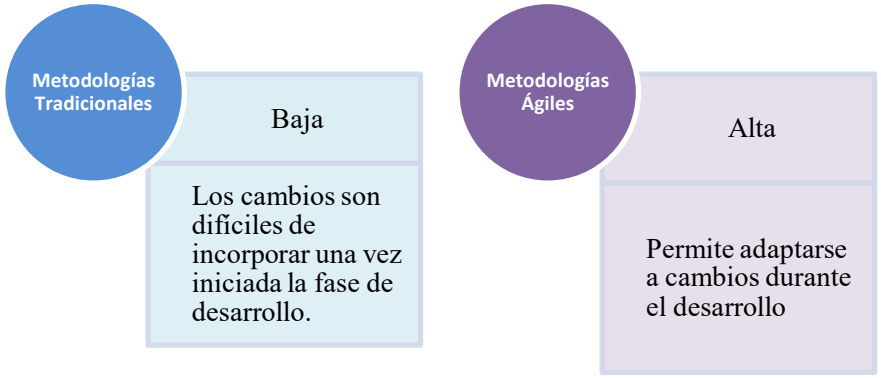
Figura 1.
Comparativa del enfoque de metodologías tradicionales vs metodologías ágiles



En la figura se destaca los dos enfoques de metodologías, para las metodologías tradicionales la predictibilidad y la documentación exhaustiva, mientras que los ágiles priorizan la flexibilidad y la entrega temprana de valor. Para este proyecto se optó por las metodologías agiles, precisamente porque permite ajustarse a los cambios que pueden surgir en el transcurso del desarrollo.

2.2.2.2. CRITERIO 2: Flexibilidad al Cambio

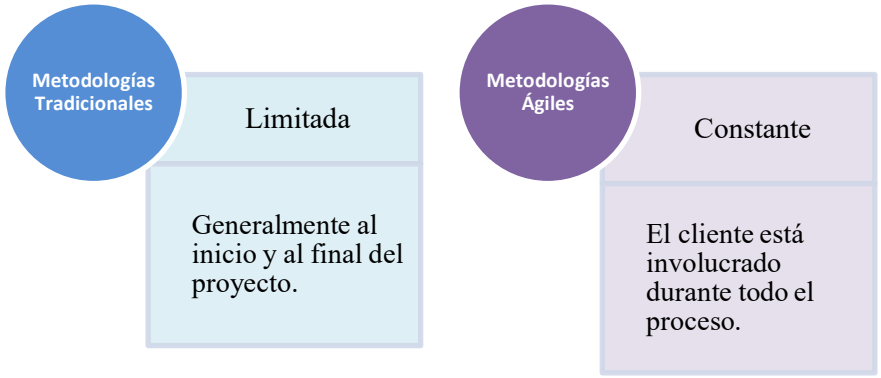
Figura 2.
Comparativa de flexibilidad entre metodologías tradicionales y ágiles



La figura evidencia que los métodos tradicionales favorecen la estabilidad del plan inicial, mientras que los ágiles favorecen la flexibilidad necesaria en proyectos con requisitos volátiles, para ejemplo tenemos el presente proyecto en el cual estamos (Pressman & Maxim, 2020; Sommerville, 2016).

2.2.2.3. CRITERIO 3: Participación del cliente

Figura 3.
Comparativa de participación del cliente entre metodologías tradicionales y ágiles

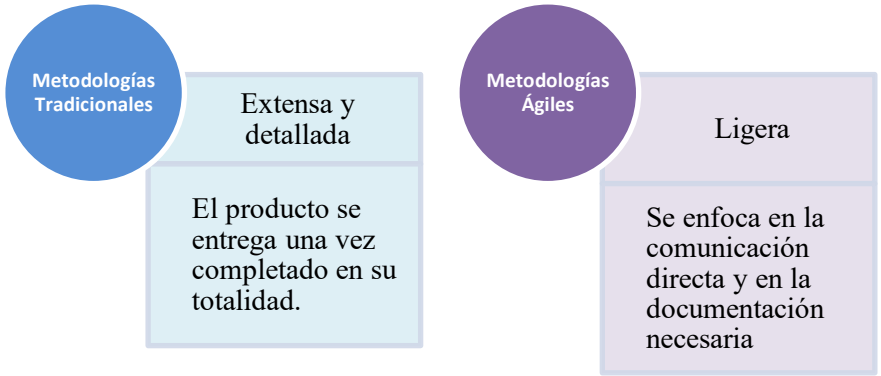


En los métodos tradicionales, como Cascada o el Modelo en V, la interacción con el cliente se concentra en la fase inicial de levantamiento de requisitos y en la aprobación final del producto; durante el desarrollo intermedio, el equipo trabaja con un plan cerrado, lo que reduce la retroalimentación y eleva el riesgo de entregar una solución que ya no se ajuste a las necesidades reales (Pressman & Maxim, 2020).

Por el contrario, las metodologías ágiles promueven la participación continua del cliente (por ejemplo, mediante el rol de *Product Owner* en Scrum), de modo que cada iteración incorpora su retroalimentación y permite ajustar el rumbo del proyecto antes de que los cambios resulten costosos, convirtiendo la colaboración constante en un pilar para maximizar el valor entregado (Sommerville, 2016).

2.2.2.4. CRITERIO 4: Documentación

Figura 4.
Comparativa de documentación entre metodologías tradicionales y ágiles

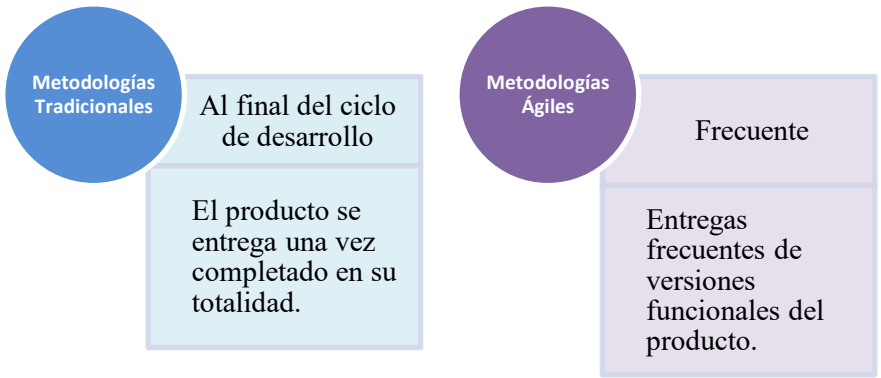


En los modelos tradicionales (p. ej., Cascada o Modelo en V) la documentación es extensa y detallada: se elaboran especificaciones completas de requisitos, diseño, pruebas y manuales antes de escribir la mayor parte del código y, de nuevo, se actualizan al cierre del proyecto. Este enfoque busca asegurar la trazabilidad y servir como contrato formal entre cliente y equipo, pero también incrementa el tiempo invertido en redacción y dificulta adaptar los documentos cuando surgen cambios (Pressman & Maxim, 2020; Sommerville, 2016).

En contraste, los marcos ágiles producen una documentación ligera **y "**justo a tiempo**"**: los artefactos se limitan a lo que realmente ayuda a comprender, compartir o mantener el software (historias de usuario, criterios de aceptación, *Definition of Done*). De acuerdo con el *Agile Manifesto*, se "valora el software funcionando más que la documentación exhaustiva", privilegiando la comunicación cara a cara y la generación de conocimientos en iteraciones cortas. Esta práctica reduce la carga administrativa y facilita la actualización continua, manteniendo la información relevante sin frenar la entrega de valor (Beck et al., 2001; Pressman & Maxim, 2020).

2.2.2.5. CRITERIO 5: Entrega de producto

Figura 5.
Comparativa de la Entrega de producto entre metodologías tradicionales y ágiles



En los enfoques tradicionales, el producto se entrega una sola vez, al término del ciclo completo de desarrollo; es decir, tras finalizar el análisis, el diseño, la codificación y las pruebas de todo el sistema. Este modo "todo-al-final" garantiza que el software llegue al

cliente en estado estable y completamente documentado, pero retrasa la obtención de valor y acumula los riesgos: cualquier falla grave o cambio de requisitos se descubre cuando su corrección resulta más costosa (Pressman & Maxim, 2020; Sommerville, 2016).

En las metodologías ágiles, la entrega es frecuente e incremental: cada iteración produce una versión funcional que el cliente puede usar, evaluar y retroalimentar de inmediato. Este flujo continuo de pequeños lanzamientos reduce el tiempo que tarda el usuario en beneficiarse del producto, permite identificar errores o ajustes tempranos y, en consecuencia, disminuye el impacto de los cambios sobre tiempo y presupuestos.

2.2.2.6. CRITERIO 6: Gestión de riesgos

Figura 6.
Comparativa de entrega de producto entre metodologías tradicionales y ágiles

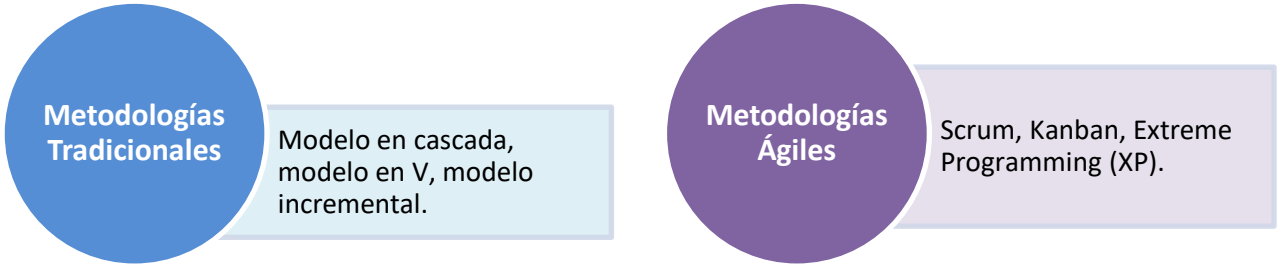


En los modelos tradicionales, la gestión de riesgos se concentra en las etapas iniciales: se identifican, priorizan y documentan posibles amenazas al comienzo del proyecto, y luego se establecen planes de contingencia que rara vez se revisan salvo que ocurra un evento crítico. Este enfoque ofrece un panorama temprano y ordenado de los peligros, pero puede dejar sin tratar riesgos emergentes que aparecen durante el desarrollo, especialmente en proyectos largos (Pressman & Maxim, 2020).

Las metodologías ágiles, en cambio, aplican una gestión de riesgos continua: cada iteración reevalúa la lista de riesgos; los problemas detectados se abordan de inmediato mediante pruebas tempranas, entregas incrementales y retrospectivas periódicas. Esto permite adaptar estrategias conforme avanza el proyecto y mantiene bajo el impacto de amenazas nuevas o cambiantes (Sommerville, 2016).

2.2.2.7. CRITERIO 7: Ejemplos

Figura 7.
Ejemplos entre metodologías tradicionales y ágiles



2.2.2.7.1. Ejemplos de metodologías tradicionales

Cascada (Pressman & Maxim, 2020):

- Es una de las metodologías más antiguas y conocidas.
- Divide el desarrollo en etapas como análisis, diseño, implementación, pruebas, despliegue y mantenimiento (Pressman & Maxim, 2020).
- Las fases son completadas antes de iniciar la siguiente, y los cambios suelen ser difíciles de realizar una vez que se avanza.
- Usado en proyectos con requerimientos claros y poco propensos a cambiar.

Modelo en V (Sommerville, 2016):

- Es una variación del modelo en cascada, pero con un enfoque en la validación y verificación.
- Cada fase de desarrollo tiene una fase de prueba correspondiente, asegurando que el producto cumpla con los requisitos en cada etapa.
- Usado en sistemas críticos como software médico o de avión.

Modelo incremental (Pressman & Maxim, 2020):

- Divide el proyecto en pequeñas iteraciones o versiones funcionales que se desarrollan y entregan de manera incremental.
- Permite incorporar retroalimentación en cada entrega y realizar ajustes.
- Usado en proyectos donde se necesitan entregas parciales rápidas.

2.2.2.7.2. Ejemplos de metodologías ágiles

Scrum:

Scrum es una metodología ágil que divide el trabajo en períodos cortos llamados sprints, que suelen durar de 2 a 4 semanas. En Scrum hay tres roles importantes: el Scrum Master, el Product Owner y el equipo de desarrollo. Cada día se realiza una reunión breve (Daily Scrum) para revisar el progreso y planificar las tareas del día. Es ideal para equipos pequeños y colaborativos que necesitan adaptarse y comunicarse continuamente (Schwaber & Sutherland, 2020; Wikipedia contributors, 2025).

Kanban:

Kanban es una metodología de gestión de flujo de trabajo que muestra las tareas mediante tableros con columnas como “Pendiente”, “En proceso” y “Terminado”(Atlassian, 2025). Fomenta la mejora continua y la flexibilidad al permitir ajustar el flujo de trabajo según la capacidad del equipo y detectar cuellos de botella rápidamente. Es ideal para equipos que buscan optimizar procesos sin dividir el trabajo en sprints, tal como recomienda Kniberg en su obra sobre la combinación de Kanban y Scrum (Kniberg & Skarin, 2020).

2.2.2.7.3. Extreme Programming (XP) :

Extreme Programming (XP) es una metodología ágil que se basa en desarrollar software en ciclos muy cortos. Lo más importante de XP es entregar pequeñas partes funcionales del sistema muy seguido, para que los usuarios las prueben rápidamente. Esta metodología tiene prácticas clave como:

- Dos personas trabajan juntas en una computadora para evitar errores y aprender mutuamente.
- Se escribe primero una prueba sencilla y luego se desarrolla el código necesario para superarla, asegurando así que siempre funcione correctamente.
- Todo el código nuevo se junta y prueba cada día, facilitando encontrar problemas enseguida.
- El código se mejora regularmente para hacerlo más sencillo y fácil de mantener, preparándolo para cambios futuros (Pressman & Maxim, 2020).

2.3. Selección de la metodología más adecuada para el proyecto

2.3.1. Metodología seleccionada

Para el desarrollo de este proyecto, se optó por la metodología Crystal Clear, propuesta por Alistair Cockburn, debido a su enfoque ágil y su adaptabilidad a equipos pequeños y proyectos de baja criticidad. Dado que este proyecto fue desarrollado por una sola persona y se enfocó en la creación de una aplicación web para la gestión del transporte privado, Crystal Clear proporcionó un marco de trabajo adecuado y flexible para abordar los desafíos específicos del proyecto (Cockburn, 2004).

2.3.1.1. ¿Por qué Crystal Clear es adecuada para este proyecto?

Crystal Clear, descrita por Cockburn (2004) como una metodología «human-powered» para equipos de hasta ocho integrantes, reduce la carga documental y privilegia la comunicación directa, lo que resulta idóneo con recursos y tiempo limitados. Sus prácticas, permiten validar cada avance sin demoras administrativas.

Además, el proyecto se ejecuta en un entorno de alta volatilidad normativa y de requisitos: Crystal Clear enfrenta esta incertidumbre mediante hitos cortos y la retroalimentación constante, transformando los cambios en oportunidades de mejora y manteniendo bajos los costos de re-trabajo. Así, la metodología ofrece la ligereza organizativa necesaria para un proyecto individual y para entregar un sistema robusto y pertinente en plazos ajustados(Cockburn, 2004).

Según Cockburn, en su libro *Crystal Clear: A Human-Powered Methodology for Small Teams*, los proyectos exitosos comparten ciertas propiedades clave, entre las cuales destacan:

- **Entrega frecuente de software funcional**: Permite obtener retroalimentación temprana y ajustar el desarrollo según las necesidades reales de los usuarios.
- **Mejora continua mediante reflexión**: Después de cada entrega, se analiza lo que funcionó y lo que no, para implementar mejoras en las siguientes iteraciones.
- **Comunicación fluida y constante**: Aunque el equipo sea pequeño, mantener una comunicación efectiva es crucial para el éxito del proyecto.
- **Acceso fácil a usuarios expertos**: Involucrar a los usuarios finales en el proceso de desarrollo ayuda a que el resultado final responda a lo que el usuario necesita.

2.3.2. Ciclo de desarrollo según Crystal Clear

El desarrollo del sistema se estructuró siguiendo el ciclo propuesto por la metodología Crystal Clear, adaptado a las características específicas de este proyecto. A continuación, se describen las fases implementadas:

2.3.2.1. Preparación del proyecto

Se realizó un análisis integral del contexto del transporte privado escolar, identificando las necesidades de los usuarios y definiendo los objetivos del sistema. Esta fase incluyó la selección de herramientas tecnológicas adecuadas y la planificación inicial del proyecto.

2.3.2.2. Planificación de entregas

Se establecieron hitos de desarrollo cortos, con objetivos específicos y funcionalidades claramente definidas. La planificación se ajustó de manera flexible, permitiendo adaptaciones según la retroalimentación obtenida en cada iteración.

2.3.2.3. Desarrollo e integración

Durante cada hito, se desarrollaron las funcionalidades priorizadas, asegurando su integración con el sistema existente. Se utilizó un entorno técnico que facilitó la coherencia y estabilidad del sistema.

2.3.2.4. Pruebas y validación

Al cierre del desarrollo, se llevaron a cabo pruebas unitarias sobre las funcionalidades críticas, garantizando su operatividad sin alargar el calendario. La validación con usuarios clave se limitó a demostraciones breves enfocadas en los aspectos esenciales, lo que permitió confirmar el cumplimiento de requisitos sin generar trabajo extra.

2.3.2.5. Retroalimentación y mejora continua

Se reflexionó e identificó áreas de mejora y ajustando el proceso de desarrollo en consecuencia. Esta práctica fomentó la mejora continua y la adaptación del sistema a las necesidades cambiantes de los usuarios.

2.3.3. Entregables generados durante el desarrollo

A lo largo del desarrollo del sistema, se generaron los siguientes entregables:

2.3.3.1. Versiones funcionales del sistema

Cada hito de desarrollo culminó, incorporando las funcionalidades desarrolladas y listas para su validación por parte de los usuarios.

2.3.3.2. Documentación técnica esencial

Se elabora documentación técnica concisa y relevante, incluyendo diagramas de casos de uso, y especificaciones de módulos, según las funcionalidades implementadas.

2.3.3.3. Resultados de pruebas

Se documentan los resultados de las pruebas realizadas, registrando los casos de prueba ejecutados.

2.3.3.4. Retroalimentación de usuarios

Se recopila y analiza la retroalimentación proporcionada por los usuarios tras la validación de cada versión del sistema, utilizando esta información para ajustar y mejorar las funcionalidades en los ciclos posteriores.

2.4. Herramientas de desarrollo

Son herramientas que al ser utilizadas permiten a los ingenieros construir sistemas de forma más eficiente, garantizando calidad, mantenibilidad y escalabilidad (Pressman & Maxim, 2020). Para el desarrollo de la presente aplicación web de gestión del servicio de transporte privado en el recinto Julio Moreno, se han seleccionado PostgreSQL, Next.js y PlantUML.

2.4.1. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto reconocido por su fiabilidad, robustez y conformidad con los estándares SQL. Ofrece características avanzadas como soporte para transacciones, extensibilidad mediante funciones definidas por el usuario y una comunidad activa que contribuye a su desarrollo continuo. Estas cualidades hacen de PostgreSQL una opción adecuada para aplicaciones que requieren una gestión eficiente y segura de los datos (*PostgreSQL: The World's Most Advanced Open Source Relational Database*, 2024).

Para el caso de este proyecto es especialmente conveniente por lo siguiente:

- Es de código abierto y libre de costo, lo que elimina cualquier inversión en licencias.
- Tiene amplia compatibilidad con diversos lenguajes de programación como, por ejemplo, Node.js y Javascript, lo que facilita su integración con el backend de Next.js y las herramientas de desarrollo utilizadas.
- Admite transacciones que garantizan la confiabilidad y consistencia de los datos incluso ante errores o fallos del sistema.
- Cuenta con una comunidad activa que proporciona soporte continuo, una gran cantidad de extensiones y actualizaciones regulares, lo que asegura la mantenibilidad a largo plazo.

2.4.2. Next.js

Next.js es un marco de trabajo de código abierto basado en React que proporciona funcionalidades adicionales como la renderización del lado del servidor y la generación de sitios estáticos. Estas características mejoran el rendimiento y la optimización en motores de búsqueda de las aplicaciones web desarrolladas. Next.js es ampliamente utilizado por empresas de diversos tamaños debido a su capacidad para construir aplicaciones web escalables y de alto rendimiento (*Next.js by Vercel – The React Framework*, 2024).

Es un proyecto open source bajo licencia MIT, lo que garantiza cero costos de licencia y total libertad para modificar y redistribuir el código.

- Ofrece gran compatibilidad con el ecosistema React, herramientas de CSS y servicios de backend, facilita la conexión con PostgreSQL y esto permitió trabajar en un entorno compatible para evitar inconvenientes al momento del desarrollo.
- Incorpora estrategias de renderizado que mejoran el SEO y reducen los tiempos de carga, ofreciendo una experiencia de usuario óptima en diferentes dispositivos

2.4.3. PlantUML

PlantUML es una herramienta libre que facilita la elaboración de diagramas UML utilizando una sintaxis textual simple y fácil de entender, que en este proyecto será utilizada para el modelado y para la creación de los casos de uso. Cuenta con características que facilitan la edición y comprensión de los diagramas, especialmente en entornos colaborativos. Además, PlantUML permite crear diferentes tipos de diagramas, como los de clases, secuencia, casos de uso, entre otros (*PlantUML*, 2024).

- Es software libre distribuido bajo licencias de código abierto, sin costo alguno, lo que se ajusta a las restricciones presupuestarias.
- Se integra fácilmente con IDEs como Visual Studio, además permite exportar diagramas en formatos imagen lo que simplifica el proceso de modelado.
- Tiene una sintaxis fácil de entender y permite representar rápidamente un modelo sin necesidad de mucho esfuerzo.
- Permite generar documentación automáticamente durante el proceso de construcción y manteniendo los diagramas actualizados con cada versión del código.

2.4.4. Visual Studio Code

Visual Studio Code es un editor de código fuente gratuito y multiplataforma desarrollado por Microsoft. Ofrece soporte integrado para depuración, control de versiones y resaltado de sintaxis para múltiples lenguajes de programación. Gracias a su flexibilidad y extensibilidad, se ha convertido en una de las herramientas más utilizadas por desarrolladores en todo el mundo, permite instalar complementos lo que lo hace un editor muy personalizable (*Visual Studio Code documentation*, 2024).

Este editor dispone de todo lo necesario para el proyecto al permitir integrar todas las herramientas en un solo lugar, creando así un entorno de trabajo óptimo para el desarrollo.

- Es un editor de código abierto distribuido bajo licencia MIT, lo cual permite su uso sin restricciones de costo, siendo ideal para proyectos académicos y de bajo presupuesto.
- Incorpora una terminal integrada que permite ejecutar comandos directamente desde el editor, lo cual mejora la productividad al evitar el cambio constante entre ventanas.

- Cuenta con funciones de autocompletado, detección de errores en tiempo real y control de versiones con Git, lo cual favorece la escritura de código limpio y bien organizado.
- Se adapta fácilmente a distintos sistemas operativos (Windows, Linux y macOS), garantizando continuidad del trabajo en distintos entornos.

2.4.5. Justificación comparativa de herramientas seleccionadas

A continuación, se presenta una tabla comparativa clara y concisa de cada herramienta elegida, destacando aspectos clave frente a sus alternativas más comunes:

Tabla 1.
PostgreSQL frente a MySQL

Criterio	PostgreSQL	MySQL
Tipo de licencia	Código abierto y gratuito	Código abierto (Oracle)
Soporte de transacciones	Completo, fiable	Depende del motor (InnoDB fiable)
Extensibilidad	Alta, con funciones definidas por usuario	Moderada
Compatibilidad	Alta integración con múltiples tecnologías	Alta

En la tabla podemos ver la comparación entre dos DBMS, PostgreSQL y MySQL.

Tabla 2.
Next.js frente a Angular y Vue

Criterio	Next.js	Angular
Licencia	MIT, abierta y gratuita	Angular MIT
Tiempo para lanzar	Rápido, configuración mínima	Puede requerir más configuración inicial
Facilidad de aprendizaje	Moderada, más fácil de comprender con conocimientos previos de React.	Alta
Documentación	Guías claras y ejemplos sencillos	Manual oficial extenso y detallado

En la tabla se puede apreciar la comparación entre dos Frameworks, basada en cuatro criterios.

Tabla 3.
PlantUML frente a Draw.io y Lucidchart

Criterio	PlantUML	Draw.io	Lucid Chart
Licencia	Abierta, gratuita	Gratuita limitada,	Premium pago
Facilidad de uso	Sintaxis textual sencilla	Interfaz gráfica	Interfaz gráfica
Integración IDE	Excelente	Limitada o mediante plugins	Limitada
Personalización	Alta(mediante código)	Baja	media

En la tabla se puede apreciar la comparación entre tres herramientas de modelado, basada en cuatro criterios.

Tabla 4.
Visual Studio Code frente a Sublime Text y Atom

Criterio	**Visual Studio Code**	**Sublime Text**	**Atom**
Licencia	Abierta, gratuita (MIT)	Comercial	Código abierto
Integración herramientas	Alta (Git, depuración, terminal)	Media	Necesita plugins adicionales
Comunidad	Muy activa y actualizaciones constantes	Moderada	En declive
Personalización	Alta	Alta	media

En la tabla se puede apreciar la comparación entre tres herramientas de modelado, basada en cuatro criterios.

Estas tablas proporcionan una comparativa visual, en base a los criterios tomados para la elección de las herramientas utilizadas para el proyecto, para un desarrollo eficaz y sostenible.

CAPÍTULO 3. Análisis de requisitos

En este capítulo nos enfocaremos en el análisis de requerimientos, y de esta manera tener una visión clara de lo que se va a construir en base a las necesidades de los usuarios del sistema, fundamental para el comienzo del desarrollo del sistema, garantizando que el desarrollo responda con precisión a las necesidades reales de sus futuros usuarios y establezca una base sólida para las fases de diseño e implementación.

3.1. Requerimientos Funcionales

3.1.1. Listado de Requerimientos Funcionales

A continuación, el listado formal de requerimientos funcionales para el sistema:

Tabla 5.
Listado de Requerimientos Funcionales con su descripción

REQUERIMIENTO	DESCRIPCIÓN DEL REQUISITO
RF1	El sistema tiene que ofrecer al usuario la posibilidad de iniciar y cerrar sesión con credenciales personales.
RF2	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar supervisores.
RF3	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar propietarios.
RF4	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar personal.
RF5	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar vehículos.
RF6	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar rutas.
RF7	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar horarios.
RF8	El sistema tiene que ofrecer al usuario la posibilidad de insertar, modificar, eliminar y consultar pasajeros.

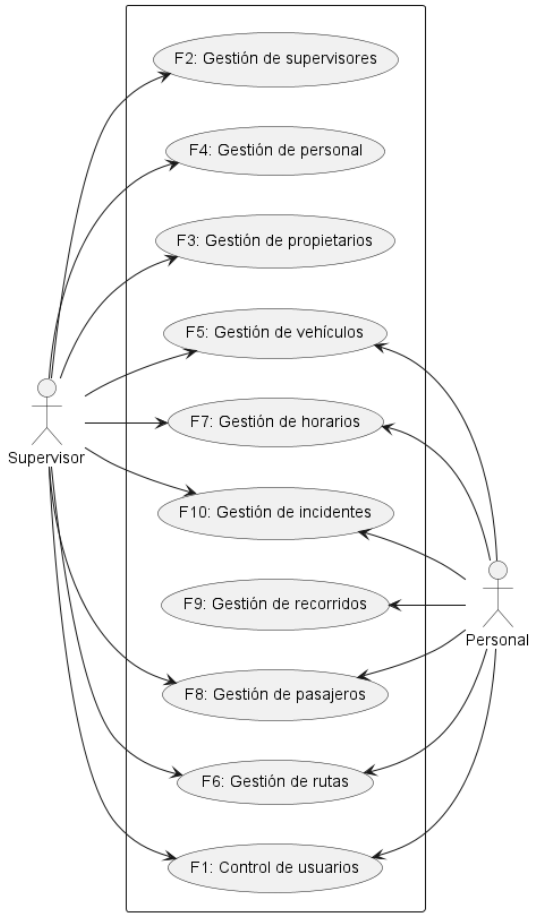
REQUERIMIENTO	DESCRIPCIÓN DEL REQUISITO
RF9	El sistema tiene que ofrecer al usuario la posibilidad de iniciar recorridos, registrar pasajeros y finalizar recorridos.
RF10	El sistema tiene que ofrecer al usuario la posibilidad de registrar y consultar incidentes.
RF11	El sistema tiene que ofrecer al usuario la posibilidad de consultar registros con filtros específicos (consultas por parámetros).

En la tabla se puede apreciar los requisitos funcionales obtenidos del usuario en reuniones.

3.1.2. Caso de uso de nivel general

Nos permiten tener una visión clara de los requerimientos funcionales, actores que intervienen con las funcionalidades del sistema, y las interrelaciones que se establecen entre ellos, facilitando así la planificación y el desarrollo posterior del proyecto.

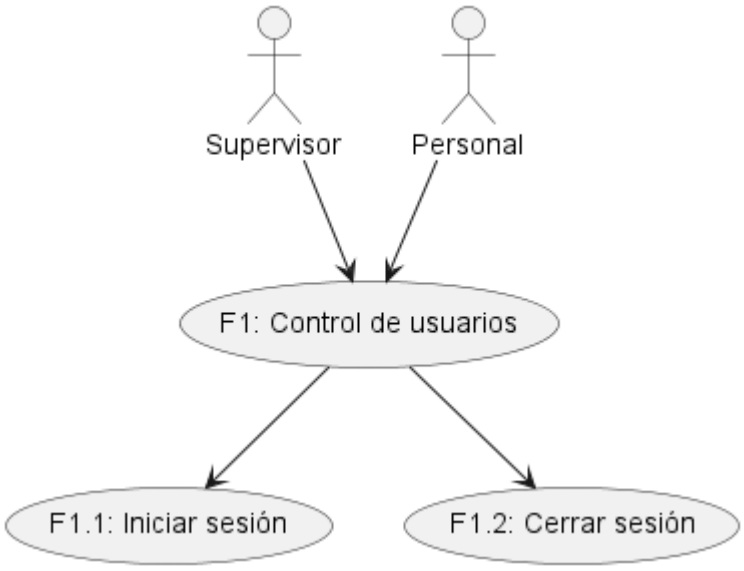
Figura 8.
Diagrama de casos de uso nivel general F0



3.1.3. Caso de uso a detalle F1: Control de usuarios

Este caso de uso se engloba las operaciones de inicio y cierre de sesión, asegurando que sólo los supervisores o el personal autorizado accedan al sistema.

Figura 9.
Diagrama de caso de uso a nivel a detalle F1



Actores: Supervisor, Personal

Descripción: El caso de uso Control de usuarios permite al actor autenticarse (iniciar sesión) y cerrar su sesión, garantizando un acceso mediante credenciales a las funcionalidades del prototipo de gestión de transporte privado.

3.1.3.1.1. F1.1: Iniciar sesión

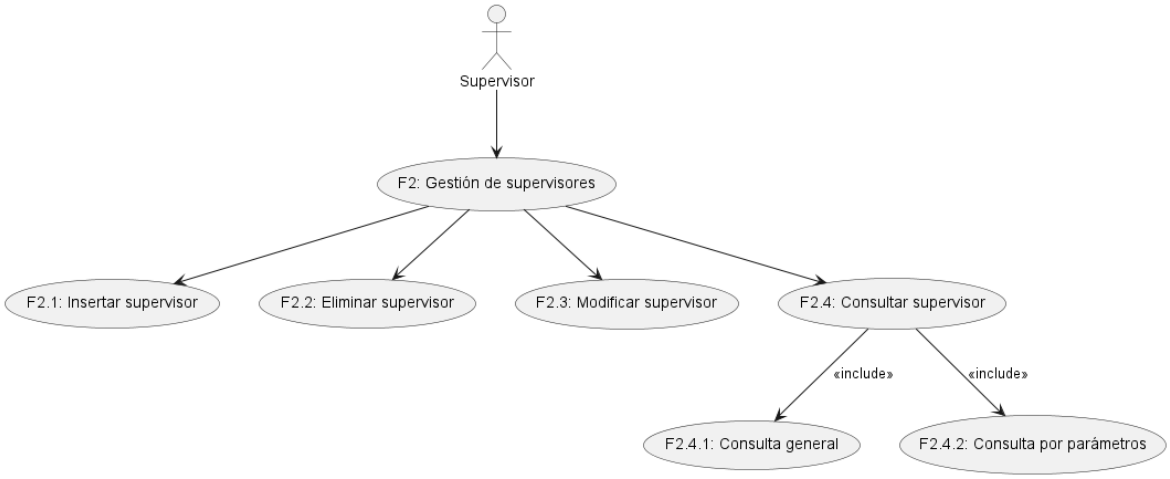
- **Actores:** Supervisor, Personal
- **Descripción:** Permite al Usuario presentar sus credenciales para acceder al sistema.
- **Flujo principal:**
    1. El actor accede a la pantalla de “Iniciar sesión”.
    2. El sistema solicita identificación y contraseña.
    3. El actor introduce sus datos y pulsa “Entrar”.
    4. El sistema valida las credenciales.
    5. El sistema redirige al actor al panel correspondiente.

F1.2: Cerrar sesión

- **Actores:** Supervisor, Personal
- **Descripción:** Permite al Usuario salir de la sesión activa y proteger su cuenta.
- **Flujo principal:**
    1. El actor hace clic en la opción “Cerrar sesión” de las opciones para el usuario en la barra de navegación en la página principal.
    2. El sistema solicita confirmación de cierre de sesión.
    3. El Usuario confirma la operación.
    4. El sistema invalida la sesión y limpia datos temporales.
    5. El sistema redirige a la pantalla de “Iniciar sesión” y muestra un mensaje de confirmación.

3.1.4. Caso de uso a detalle F2: Gestión de supervisores

Figura 10.
Diagrama de caso de uso a detalle F2



Actor: Supervisor

- Descripción: Permite al actor administrar los registros de usuarios, realizando operaciones de inserción, eliminación, modificación y consulta.

3.1.4.1. F2.1: Insertar supervisor

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de supervisor al sistema.

- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor presiona la opción supervisores.
    3. El actor selecciona el botón añadir en supervisores.
    4. El sistema redirige a un formulario con los campos:
        - Cédula del supervisor
        - Contraseña del supervisor
        - Nombre del supervisor
        - Contacto del supervisor
        - Estado del supervisor
    5. El actor llena los campos con los datos requeridos y pulsa el botón guardar.
    6. El sistema valida el ID del supervisor.
    7. El sistema crea el nuevo registro en la tabla supervisores y muestra un mensaje de confirmación.

3.1.4.2. F2.2: Eliminar supervisor

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de supervisor existente.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción supervisores.
    3. El sistema redirige al actor a la página de supervisores, recupera todos los registros de la tabla supervisores y los muestra.
    4. El actor selecciona el registro a eliminar de la tabla y da clic en la opción detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor da clic en eliminar.
    7. El sistema despliega una ventana que advierte de la eliminación del registro.
    8. El actor procede con la eliminación y la confirma.
    9. El sistema borra el registro.
    10. El sistema entrega al actor un aviso que indica que la operación se completó con éxito.

3.1.4.3. F2.3: Modificar supervisor

- **Actor:** Supervisor
- **Descripción:** Actualizar los datos de un supervisor existente.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción supervisores.
    3. El sistema redirige al actor a la página de supervisores, recupera todos los registros de la tabla supervisores y los muestra.
    4. El actor selecciona el registro que desea modificar de la tabla y da clic en detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor modifica lo que necesite y presiona guardar.
    7. El sistema valida los datos.
    8. El sistema efectúa las modificaciones y confirma la actualización.

3.1.4.4. F2.4: Consultar supervisor

- **Actor:** Supervisor
- **Descripción:** Visualizar información de uno o varios supervisores.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción supervisores.
    3. El sistema redirige al actor a la página de supervisores, recupera todos los registros de la tabla supervisores y los muestra.
    4. El actor ingresa el dato con el que desea filtrar la búsqueda del registro en cuestión.

3.1.4.4.1. F2.4.1: Consulta general

- **Actor:** Supervisor
- **Descripción:** Obtener el listado completo de supervisores registrados.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción supervisores.

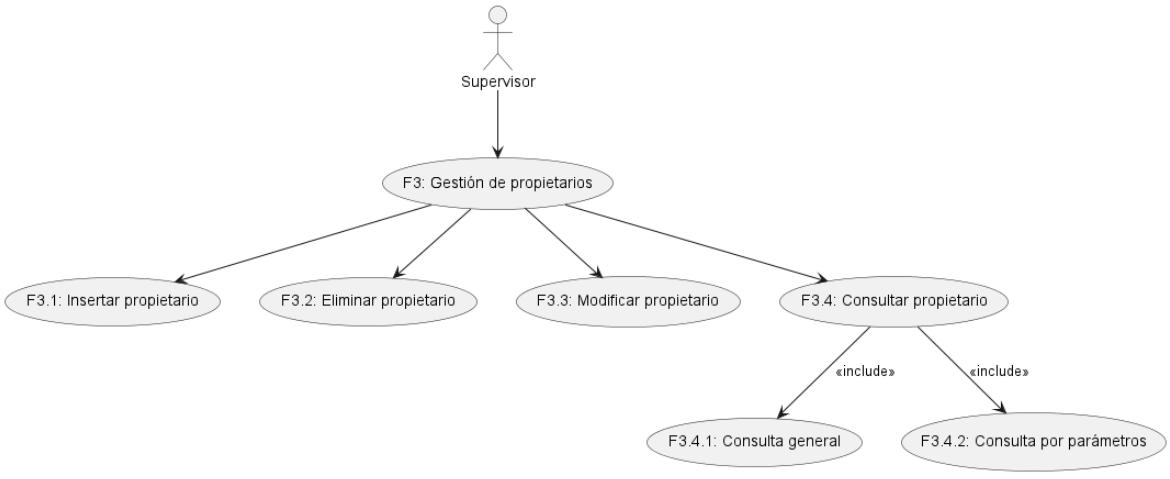
3. El sistema redirige al actor a la página de supervisores, recupera todos los registros de la tabla supervisores y los muestra.

3.1.4.4.2. F2.4.2: Consulta por parámetros

- **Actor:** Supervisor
- **Descripción:** Filtrar supervisores según criterios específicos (p. ej., ID, nombre o estado).
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción supervisores.
    3. El sistema redirige al actor a la página de supervisores, recupera todos los registros de la tabla supervisores y los muestra.
    4. El actor ingresa el dato con el que desea filtrar la búsqueda del registro en cuestión.
    5. El sistema trae los registros en función de lo que llena el usuario.

3.1.5. Caso de uso a detalle F3: Gestión de propietarios

Figura 11.
Diagrama de caso de uso a detalle F3



- **Actor:** Supervisor
- **Descripción:** Permite al actor administrar los registros de propietarios, realizando operaciones de inserción, eliminación, modificación y consulta.

3.1.5.1. F3.1: Insertar propietario

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de propietario al sistema.
- Flujo principal:
    1. El actor da clic en la opción flota vehicular de la barra de navegación.
    2. El actor elige la opción propietarios.
    3. El actor selecciona el botón añadir en propietarios.
    4. El sistema redirige a la página de detalle y muestra un formulario con los campos:
        - Cédula del propietario
        - Nombre del propietario
        - Contacto del propietario
        - Estado del propietario
    5. El actor llena los campos con los datos requeridos y da clic en guardar.
    6. El sistema valida el ID del propietario.
    7. El sistema crea el nuevo registro en la tabla propietarios y muestra un mensaje de confirmación.

3.1.5.2. F3.2: Eliminar propietario

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de propietario existente.
- **Flujo principal:**
    1. El actor da clic en la opción flota vehicular de la barra de navegación.
    2. El actor elige la opción propietarios.
    3. El sistema redirige al actor a la página de propietarios, recupera todos los registros de la tabla propietarios y los muestra.
    4. El actor selecciona el registro que desea eliminar y da clic en detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor da clic en eliminar.
    7. El sistema advierte al actor que al continuar se eliminará el registro.
    8. El actor procede con la eliminación.
    9. El sistema elimina el registro.

10. El sistema indica al actor que la eliminación se completó correctamente.

3.1.5.3. F3.3: Modificar propietario

- **Actor:** Supervisor
- **Descripción:** Actualizar los datos de un propietario existente.
- **Flujo principal:**
    1. El actor da clic en la opción flota vehicular de la barra de navegación.
    2. El actor elige la opción propietarios.
    3. El sistema redirige al actor a la página de propietarios, recupera todos los registros de la tabla propietarios y los muestra.
    4. El actor elige que registro va a modificar y da clic en la opción detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor edita los campos necesarios y presiona el botón guardar.
    7. El sistema valida los datos.
    8. El sistema guarda las modificaciones y confirma la actualización.

3.1.5.4. F3.4: Consultar propietario

- **Actor:** Supervisor
- **Descripción:** Visualizar información de uno o varios propietarios.
- **Flujo principal:**
    1. El actor da clic en la opción flota vehicular de la barra de navegación.
    2. El actor elige la opción propietarios.
    3. El sistema redirige al actor a la página de propietarios, recupera todos los registros de la tabla propietarios y los muestra.
    4. El actor llena el o los campos a filtrar en el formulario ubicado en la parte superior.

3.1.5.4.1. F3.4.1: Consulta general

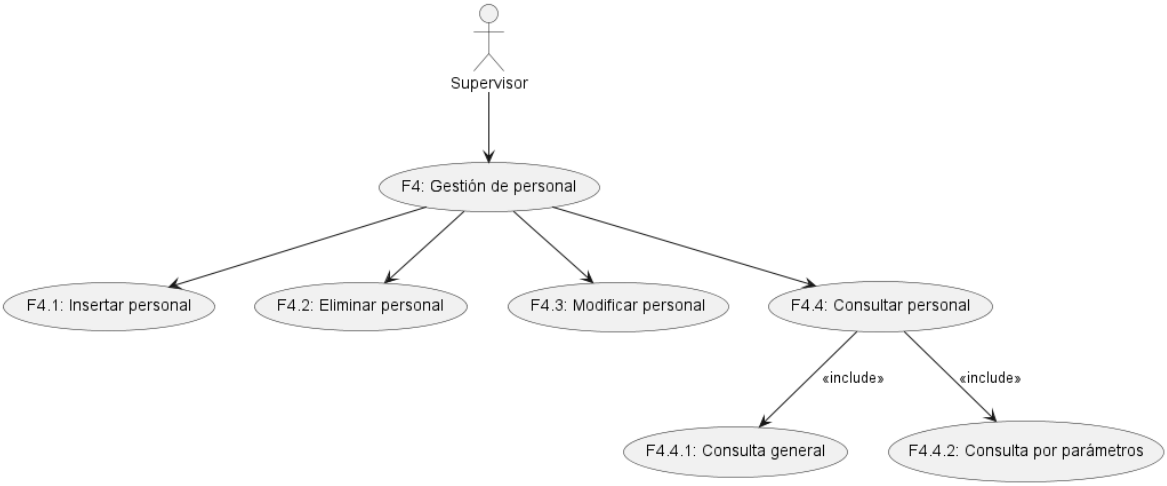
- **Actor:** Supervisor
- **Descripción:** Obtener el listado completo de propietarios registrados.
- **Flujo principal:**
    1. El actor da clic en la opción flota vehicular de la barra de navegación.
    2. El actor elige la opción propietarios.
    3. El sistema redirige al actor a la página de propietarios, recupera todos los registros de la tabla propietarios y los muestra.

3.1.5.4.2. F3.4.2: Consulta por parámetros

- **Actor:** Supervisor
- **Descripción:** Filtrar propietarios según criterios específicos (p. ej., ID, nombre o estado).
- **Flujo principal:**
    1. El actor da clic en la opción flota vehicular de la barra de navegación.
    2. El actor elige la opción propietarios.
    3. El sistema redirige al actor a la página de propietarios, recupera todos los registros de la tabla propietarios y los muestra.
    4. El actor llena el o los campos a filtrar en los campos de búsqueda.
    5. El sistema trae los registros en función de lo que llena el usuario.

3.1.6. Caso de uso a detalle F4: Gestión de personal

Figura 12.
Diagrama de caso de uso a detalle F4



Actor: Supervisor

Descripción: Permite al actor administrar los registros de personal, realizando operaciones de inserción, eliminación, modificación y consulta.

3.1.6.1. F4.1: Insertar personal

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de personal al sistema.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción personal.
    3. El actor selecciona el botón añadir en personal.
    4. El sistema muestra un formulario con los campos:
        - Cédula del personal
        - Contraseña del personal
        - Nombre del personal
        - Contacto del personal
        - Estado del personal
    5. El actor completa los datos y pulsa el botón guardar.
    6. El sistema valida el ID del personal y la existencia del supervisor asociado.
    7. El sistema crea el nuevo registro en la tabla personales y muestra un mensaje de confirmación.

3.1.6.2. F4.2: Eliminar personal

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de personal existente.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción personal.
    3. El sistema redirige al actor a la página de Personal, recupera todos los registros de la tabla personales y los muestra.
    4. El actor selecciona el registro a eliminar y presiona da clic en la opción detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor da clic en eliminar.
    7. El sistema advierte al actor que al continuar se eliminará el registro.
    8. El actor prosigue con la eliminación al elegir confirmar.
    9. El sistema borra el registro.
    10. El sistema confirma al actor que se completó con éxito la acción.

F4.3: Modificar personal

- **Actor:** Supervisor
- **Descripción:** Actualizar los datos de un personal existente.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción personal.
    3. El sistema redirige al actor a la página de Personal, recupera todos los registros de la tabla personales y los muestra.
    4. El actor elige el personal que desea cambiar y da clic en detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor modifica lo que necesite y presiona guardar.
    7. El sistema valida los campos.
    8. El sistema efectúa las modificaciones y confirma la actualización.

F4.4: Consultar personal

- **Actor:** Supervisor
- **Descripción:** Visualizar información de uno o varios personal.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción personal.
    3. El sistema redirige al actor a la página de supervisores, recupera todos los registros de la tabla supervisores y los muestra.
    4. El actor utiliza el o los campos para filtrar su búsqueda.

F4.4.1: Consulta general

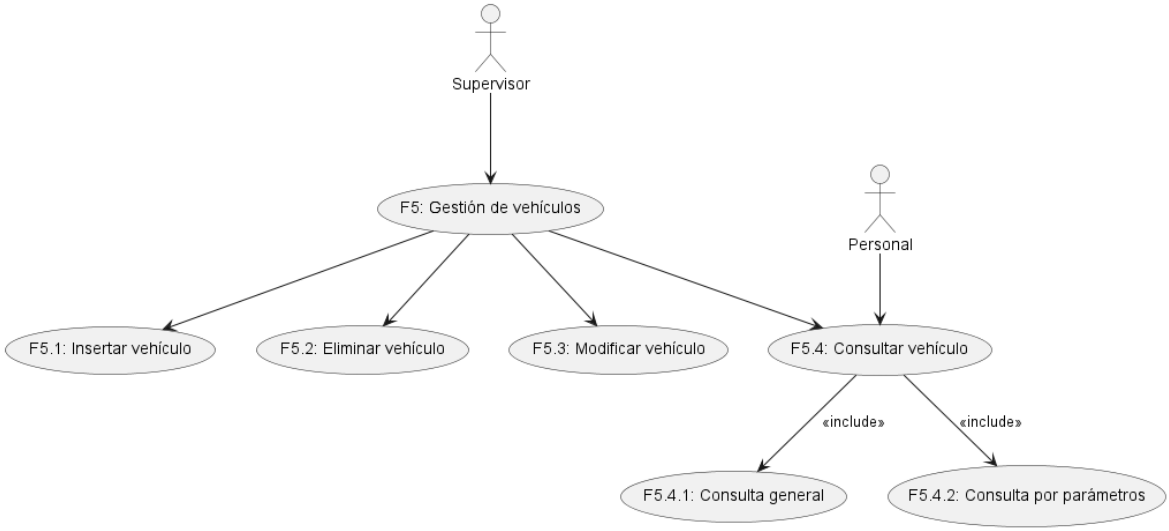
- **Actor:** Supervisor
- **Descripción:** Obtener el listado completo de personal registrado.
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción personal.
    3. El sistema redirige al actor a la página de Personal, recupera todos los registros de la tabla personales y los muestra.

F4.4.2: Consulta por parámetros

- **Actor:** Supervisor
- **Descripción:** Filtrar personal según criterios específicos (p. ej., ID, nombre o estado).
- **Flujo principal:**
    1. El actor da clic en la opción sistema de la barra de navegación.
    2. El actor elige la opción personal.
    3. El sistema redirige al actor a la página de Personal, recupera todos los registros de la tabla personales y los muestra.
    4. El actor llena el o los campos a filtrar en los campos de búsqueda.
    5. El sistema trae los registros en función de lo que llena el usuario.

3.1.7. Caso de uso a detalle F5: Gestión de vehículos

Figura 13.
Diagrama de caso de uso a detalle F5



- **Actor:** Supervisor
- **Descripción:** Permite al actor administrar los registros de vehículos, realizando operaciones de inserción, eliminación, modificación y consulta.

3.1.7.1. F5.1: Insertar vehículo

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de vehículo al sistema.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor da clic en vehículos.
    3. El actor selecciona el botón añadir en vehículos.
    4. El sistema redirige a la página con los campos a insertar:
        - Placa del vehículo
        - Modelo del vehículo
        - Capacidad del vehículo
        - Estado del vehículo
        - Propietario del vehículo
        - Conductor del vehículo

    5. El actor completa los datos y da clic en guardar.

6. El sistema comprueba que los datos sean correctos.
7. El sistema crea el nuevo registro en la tabla vehículos y muestra un mensaje de confirmación.

3.1.7.2. F5.2: Eliminar vehículo

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de vehículo existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor da clic en vehículos.
    3. El sistema redirige al actor a la página de Vehículos, recupera todos los registros de la tabla vehículos y los muestra.
    4. El actor selecciona el registro que desea eliminar y presiona en el botón detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor da clic en la opción eliminar.
    7. El sistema despliega una advertencia que indica que se eliminará el registro si continua.
    8. El actor da clic en confirmar.
    9. El sistema elimina el registro.
    10. El sistema confirma al actor que la acción se realizó exitosamente.

3.1.7.3. F5.3: Modificar vehículo

- **Actor:** Supervisor
- **Descripción:** Actualizar los datos de un vehículo existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor da clic en vehículos.
    3. El sistema redirige al actor a la página de Vehículos, recupera todos los registros de la tabla vehículos y los muestra.
    4. El actor da clic en el botón detalles del registro que desea modificar en vehículos.

5. El sistema muestra la página con el detalle del registro y allí los campos que el actor puede modificar.
6. El actor modifica lo necesario y da clic en guardar.
7. El sistema verifica que los datos sean correctos.
8. El sistema efectúa las modificaciones y confirma la actualización.

3.1.7.4. F5.4: Consultar vehículo

- **Actor:** Supervisor
- **Descripción:** Visualizar información de uno o varios vehículos.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor presiona el botón vehículos.
    3. El sistema redirige al actor a la página de vehículos, recupera todos los registros de la tabla vehículos y los muestra.
    4. El actor utiliza el o los campos para filtrar su búsqueda.

3.1.7.4.1. F5.4.1: Consulta general

- **Actor:** Supervisor
- **Descripción:** Obtener el listado completo de vehículos registrados.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor da clic en la opción vehículos.
    3. El sistema redirige al actor a la página de Vehículos, recupera todos los registros de la tabla vehículos y los muestra.

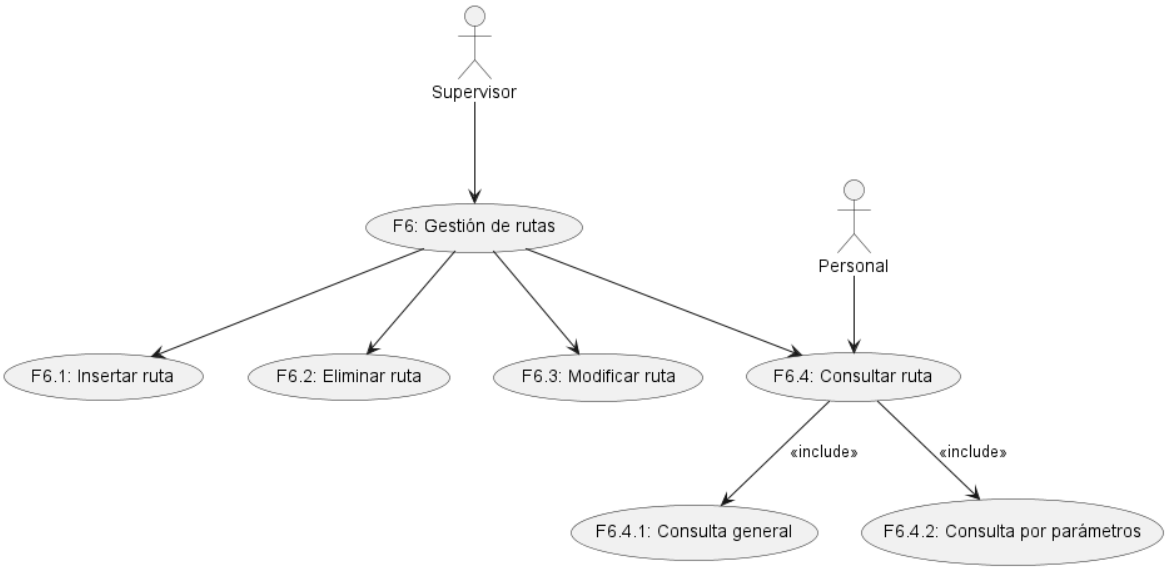
3.1.7.4.2. F5.4.2: Consulta por parámetros

- **Actor:** Supervisor
- **Descripción:** Filtrar vehículos según criterios específicos (p. ej., ID, modelo, estado).
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor da clic en la opción vehículos.

3. El sistema redirige al actor a la página de Vehículos, recupera todos los registros de la tabla vehículos y los muestra.
4. El actor utiliza el o los campos para filtrar su búsqueda.
5. El sistema trae los registros en función de lo que llena el usuario.

3.1.8. Caso de uso a detalle F6: Gestión de rutas

Figura 14.
Diagrama de caso de uso a detalle F2



- **Actor:** Supervisor, Personal
- **Descripción:** Permite al Supervisor administrar los registros de rutas y al Personal consultar la información de sus rutas asignadas.

3.1.8.1. F6.1: Insertar ruta

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de ruta al sistema.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción rutas.
    3. El actor selecciona el botón añadir en rutas.

4. El sistema redirige a una página que contiene los distintos campos necesarios:
    - Id de la ruta
    - Nombre de la ruta
    - Inicio de la ruta
    - Fin de la ruta
    - Estado de la ruta
    - Vehículo asignado a la ruta
5. El actor completa los datos y pulsa el botón guardar.
6. El sistema valida el ID de la ruta y la existencia del vehiculo_id asociado.
7. El sistema crea el nuevo registro en la tabla rutas y muestra un mensaje de confirmación.

3.1.8.2. F6.2: Eliminar ruta

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de ruta existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción rutas.
    3. El sistema redirige al actor a la página de Rutas, recupera todos los registros de la tabla rutas y los muestra.
    4. El actor elige el registro a eliminar desde la tabla y hace clic en el botón de detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor hace clic en el botón de eliminar.
    7. El sistema muestra una ventana para confirmar la eliminación del registro.
    8. El actor confirma la eliminación.
    9. El sistema elimina el registro.
    10. El sistema notifica al Supervisor que la acción se realizó correctamente.

3.1.8.3. F6.3: Modificar ruta

• **Actor:** Supervisor

• **Descripción:** Actualizar los datos de una ruta existente.

• **Flujo principal:**

1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
2. El actor elige la opción rutas.
3. El sistema redirige al actor a la página de Rutas, recupera todos los registros de la tabla rutas y los muestra.
4. El actor elige el registro que desea modificar desde la tabla y hace clic en el botón de detalles.
5. El sistema redirige al actor hacia el formulario con el detalle del registro.
6. El actor modifica los campos requeridos y luego hace clic en el botón guardar.
7. El sistema verifica los datos.
8. El sistema efectúa las modificaciones y confirma la actualización.

3.1.8.4. F6.4: Consultar ruta

- **Actor:** Supervisor, Personal
- **Descripción:** Visualizar información de las rutas asignadas.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción rutas.
    3. El sistema redirige al actor a la página de rutas, recupera todos los registros de la tabla rutas y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario superior.

3.1.8.4.1. F6.4.1: Consulta general

- **Actor:** Supervisor, Personal
- **Descripción:** Obtener el listado completo de rutas disponibles.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción rutas.

3. El sistema redirige al actor a la página de Rutas, recupera todos los registros de la tabla rutas y los muestra.

3.1.8.4.2. F6.4.2: Consulta por parámetros

• **Actor:** Supervisor, Personal

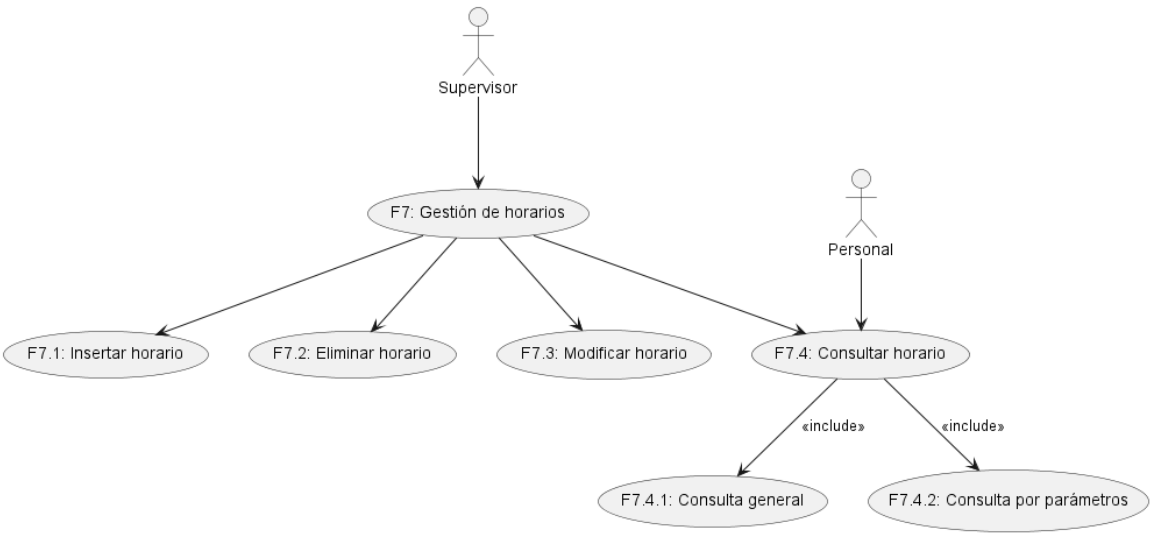
• **Descripción:** Filtrar rutas según criterios específicos (p. ej., nombre de ruta, origen o destino).

• **Flujo principal:**

1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
2. El actor elige la opción rutas.
3. El sistema redirige al actor a la página de Rutas, recupera todos los registros de la tabla rutas y los muestra.
4. El actor ingresa los criterios de búsqueda en el formulario superior.
5. El sistema trae los registros en función de los criterios ingresados.

3.1.9. Caso de uso a detalle F7: Gestión de horarios

Figura 15.
Diagrama de caso de uso a detalle F7



- **Actor:** Supervisor, Personal
- **Descripción:** Permite al actor administrar los registros de horarios, realizando operaciones de inserción, eliminación, modificación y consulta.

3.1.9.1. F7.1: Insertar horario

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de horario al sistema.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción horarios.
    3. El actor selecciona el botón añadir en horarios.
    4. El sistema redirige a una página que contiene los distintos campos necesarios:
        - Nombre del horario
        - Inicio del horario
        - Fin del horario
        - Estado del horario
        - Ruta
    5. El actor completa los datos y pulsa el botón guardar.
    6. El sistema valida la coherencia de horas (inicio < fin).
    7. El sistema crea el nuevo registro en la tabla horarios y muestra un mensaje de confirmación.

3.1.9.2. F7.2: Eliminar horario

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de horario existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción horarios.
    3. El sistema redirige al actor a la página de Horarios, recupera todos los registros de la tabla horarios y los muestra.
    4. El actor selecciona el registro a eliminar de la tabla y da clic en la opción detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor selecciona eliminar.
    7. El sistema muestra un mensaje que pide la confirmación para la eliminación del registro.

8. El actor acepta la eliminación del registro.
9. El sistema borra el registro.
10. El sistema le comunica al actor que la operación se realizó con éxito.

3.1.9.3. F7.3: Modificar horario

- **Actor:** Supervisor
- **Descripción:** Actualizar los datos de un horario existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción horarios.
    3. El sistema redirige al actor a la página de Horarios, recupera todos los registros de la tabla horarios y los muestra.
    4. El actor escoge el registro que desea modificar desde la tabla y da clic en detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor actualiza los campos requeridos y da clic en guardar.
    7. El sistema valida la coherencia de las horas y la existencia de ruta_id.
    8. El sistema efectúa los cambios en la tabla horarios y confirma la actualización.

3.1.9.4. F7.4: Consultar horario

- **Actor:** Supervisor, personal
- **Descripción:** Visualizar información de uno o varios horarios.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción horarios.
    3. El sistema redirige al actor a la página de horarios, recupera todos los registros de la tabla horarios y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario que se encuentra en la parte de arriba.

3.1.9.4.1. F7.4.1: Consulta general

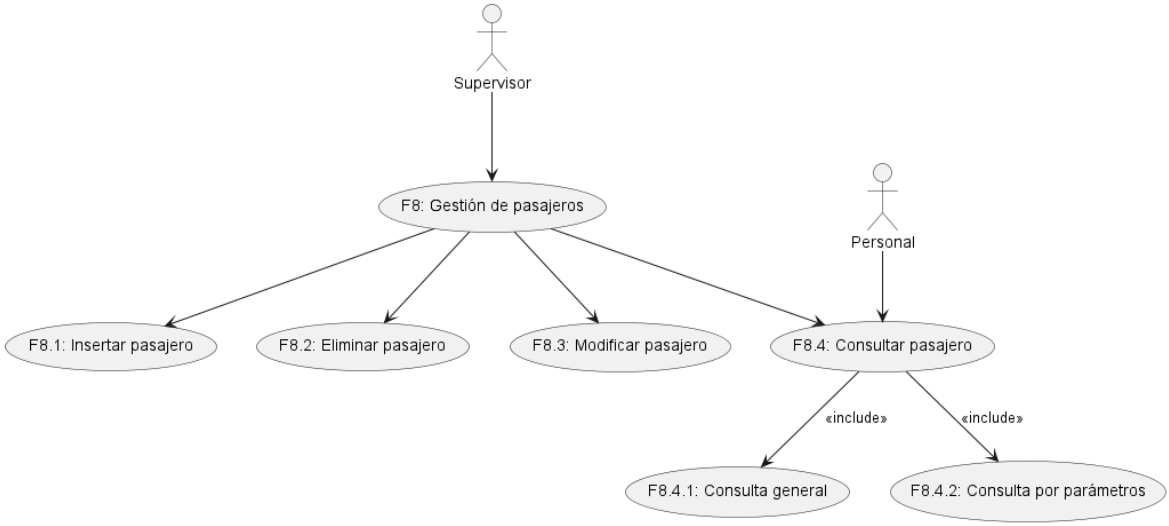
- **Actor:** Supervisor
- **Descripción:** Obtener el listado completo de horarios registrados.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción horarios.
    3. El sistema redirige al actor a la página de Horarios, recupera todos los registros de la tabla horarios y los muestra.

3.1.9.4.2. F7.4.2: Consulta por parámetros

- **Actor:** Supervisor
- **Descripción:** Filtrar horarios según criterios específicos (p. ej., nombre, rango de hora o ruta).
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige la opción horarios.
    3. El sistema redirige al actor a la página de Horarios, recupera todos los registros de la tabla horarios y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario que se encuentra en la parte de arriba.
    5. El sistema trae los registros en función de los filtros ingresados.

3.1.10.Caso de uso a detalle F8: Gestión de pasajeros

Figura 16.
Diagrama de caso de uso a detalle F8



- **Actor:** Supervisor, Personal
- **Descripción:** Permite al Supervisor administrar los registros de pasajeros y al Personal consultar la información de pasajeros.

3.1.10.1. F8.1: Insertar pasajero

- **Actor:** Supervisor
- **Descripción:** Agregar un nuevo registro de pasajero al sistema.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor elige la opción pasajeros.
    3. El actor selecciona el botón añadir en pasajeros.
    4. El sistema redirige a una página que contiene los distintos campos necesarios:
        - Cédula del pasajero
        - Nombre del pasajero
        - Contacto del pasajero
        - Estado del pasajero
    5. El actor ingresa esta información y da clic en guardar.
    6. El sistema valida el ID del pasajero.
    7. El sistema crea el nuevo registro en la tabla pasajeros y muestra un mensaje de confirmación.

3.1.10.2. F8.2: Eliminar pasajero

- **Actor:** Supervisor
- **Descripción:** Eliminar un registro de pasajero existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor elige la opción pasajeros.
    3. El sistema redirige al actor a la página de pasajeros, recupera todos los registros de la tabla pasajeros y los muestra.
    4. El actor selecciona el registro a eliminar de la tabla y da clic en detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor presiona la opción eliminar.
    7. El sistema muestra un mensaje que pide la confirmación para la eliminación del registro.
    8. El actor procede confirmando la eliminación.
    9. El sistema borra el registro.
    10. El sistema le indica al actor que se ha completado la acción con éxito.

3.1.10.3. F8.3: Modificar pasajero

- **Actor:** Supervisor
- **Descripción:** Actualizar los datos de un pasajero existente.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor elige la opción pasajeros.
    3. El sistema redirige al actor a la página de pasajeros, recupera todos los registros de la tabla pasajeros y los muestra.
    4. El actor selecciona el registro que desea alterar en la tabla y da clic en detalles.
    5. El sistema redirige al actor hacia el formulario con el detalle del registro.
    6. El actor altera los campos necesarios y hace clic en el botón guardar.
    7. El sistema verifica que los datos sean correctos.
    8. El sistema efectúa las modificaciones y confirma la actualización.

3.1.10.4. F8.4: Consultar pasajero

- **Actor:** Personal
- **Descripción:** Visualizar información de uno o varios pasajeros.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor elige la opción pasajeros.
    3. El sistema redirige al actor a la página de vehículos, recupera todos los registros de la tabla vehículos y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario que se encuentra en la parte de arriba.

3.1.10.4.1. F8.4.1: Consulta general

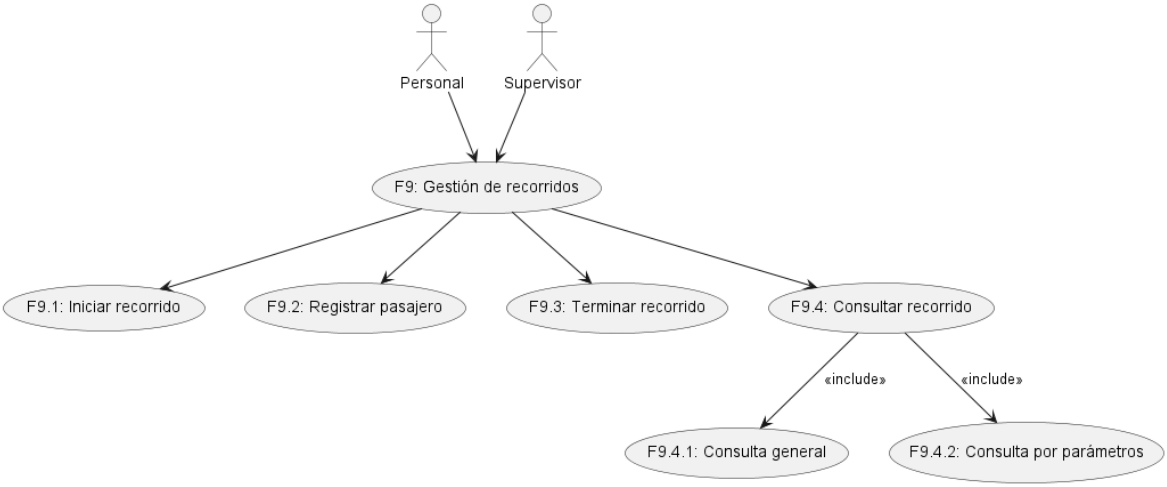
- **Actor:** Personal
- **Descripción:** Obtener el listado completo de pasajeros registrados.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor elige la opción pasajeros.
    3. El sistema redirige al actor a la página de pasajeros, recupera todos los registros de la tabla pasajeros y los muestra.

3.1.10.4.2. F8.4.2: Consulta por parámetros

- **Actor:** Personal
- **Descripción:** Filtrar pasajeros según criterios específicos (p. ej., ID, nombre o estado).
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción flota vehicular.
    2. El actor elige la opción pasajeros.
    3. El sistema redirige al actor a la página de pasajeros, recupera todos los registros de la tabla pasajeros y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario que se encuentra en la parte de arriba.
    5. El sistema trae los registros en función de lo que llena el usuario.

3.1.11.Caso de uso a detalle F9: Gestión de recorridos

Figura 17.
Diagrama de caso de uso a detalle F9



- **Actor:** Personal, Supervisor
- **Descripción:** Permite al Personal iniciar recorridos, registrar pasajeros y terminarlos, y al Supervisor consultar la información de los recorridos.

3.1.11.1. F9.1: Iniciar recorrido

- **Actor:** Personal, Supervisor
- **Descripción:** Permite al Personal registrar el inicio de un nuevo recorrido.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor selecciona recorridos.
    3. El actor hace clic en el botón añadir.
    4. El sistema crea un registro con la hora de inicio
    5. El sistema crea el nuevo registro en la tabla recorridos y muestra un mensaje de confirmación.

3.1.11.2. F9.2: Registrar pasajero

- **Actor:** Personal, Supervisor
- **Descripción:** Permite al Personal añadir pasajeros al recorrido en curso.

- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor selecciona recorridos.
    3. El sistema muestra el recorrido activo y ofrece el botón Registrar pasajero.
    4. El actor pulsa ese botón y el sistema trae la lista de pasajeros al recorrido.
    5. El actor marca con una checkbox a los pasajeros presentes.

3.1.11.3. F9.3: Terminar recorrido

- **Actor:** Personal
- **Descripción:** Permite al Personal dar por finalizado un recorrido registrado.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige recorridos.
    3. El sistema trae la lista de recorridos.
    4. El actor selecciona el recorrido que desea finalizar y en la tabla selecciona detalles.
    5. El sistema redirige al actor a la página con el detalle del recorrido.
    6. El actor ingresa la fecha y hora de finalización del recorrido y da clic en guardar.
    7. El sistema registra la fecha y hora de fin en la tabla recorridos y confirma al actor que la acción fue exitosa.

3.1.11.4. F9.4: Consultar recorrido

- **Actor:** Supervisor, Personal
- **Descripción:** Permite al actor visualizar el histórico de recorridos.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor elige recorridos.
    3. El sistema redirige al actor a la página de recorridos, recupera todos los registros de la tabla recorridos y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario que se encuentra en la parte de arriba.

3.1.11.4.1. F9.4.1: Consulta general

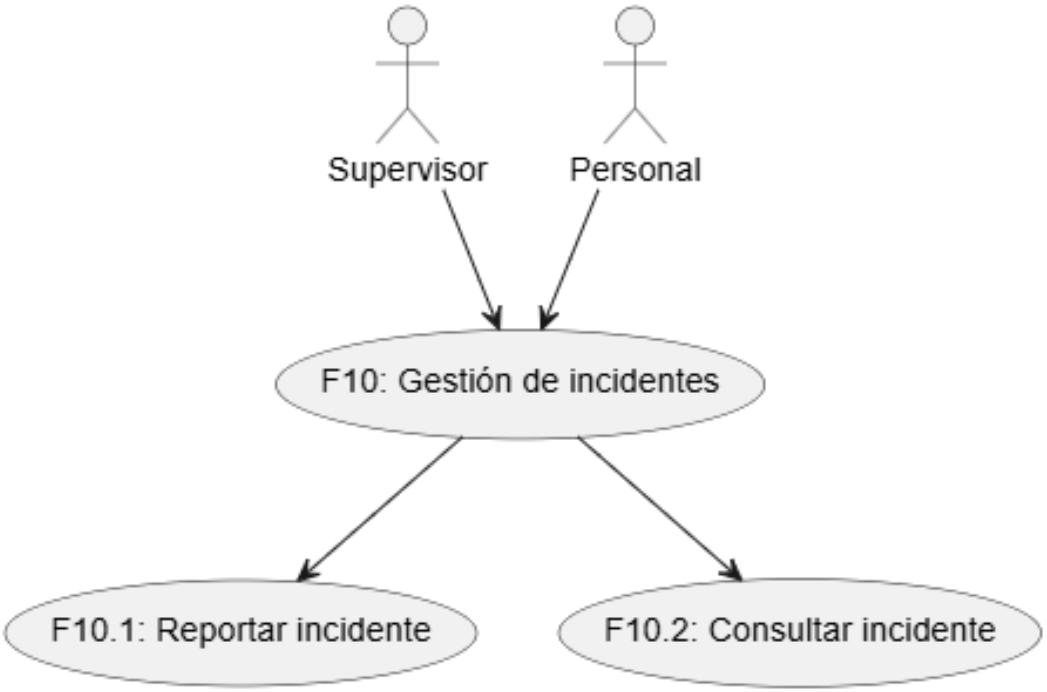
- **Actor:** Supervisor
- **Descripción:** Obtener el listado completo de recorridos.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor presiona el botón recorridos.
    3. El sistema redirige al actor a la página de recorridos, recupera todos los registros de la tabla recorridos y los muestra.

3.1.11.4.2. F9.4.2: Consulta por parámetros

- **Actor:** Supervisor, Personal
- **Descripción:** Filtrar recorridos según criterios (p. ej., fecha, horario o estado).
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor presiona el botón recorridos.
    3. El sistema redirige al actor a la página de recorridos, recupera todos los registros de la tabla recorridos y los muestra.
    4. El actor ingresa los criterios de búsqueda en el formulario que se encuentra en la parte de arriba.
    5. El sistema trae los registros en función de lo que llena el usuario.

3.1.12.Caso de uso a detalle F10: Gestión de incidentes

Figura 18.
Diagrama de caso de uso a detalle F10



- **Actor:** Personal, Supervisor
- **Descripción:** Permite al actor reportar nuevos incidentes durante un recorrido y al Supervisor consultar los incidentes registrados.

3.1.12.1. F10.1: Reportar incidente

- **Actor:** Supervisor, Personal
- **Descripción:** Crear un nuevo registro de incidente asociado a un recorrido.
- **Flujo principal:**
    1. El actor se dirige a la barra de navegación y elige la opción itinerarios.
    2. El actor presiona el botón recorridos.
    3. El actor hace clic en la opción seleccionable incidentes en la parte superior.
    4. El sistema redirige a una página que muestra los registros de incidentes:
        - Fecha del incidente
        - Lugar del incidente
        - Título del incidente
        - Detalles del incidente

5. El actor presiona en el icono de agregar.
6. El sistema muestra los campos para ingresar un nuevo registro
7. El actor llena los campos necesarios.
8. El actor da clic en guardar.
9. El sistema valida que los datos sean correctos.
10. El sistema muestra el incidente creado en la tabla.

3.1.12.2. F10.2: Consultar incidente

- **Actor:** Supervisor, Personal
- **Descripción:** Visualizar la información de uno o varios incidentes registrados.
- **Flujo principal:**
    1. El actor selecciona la opción itinerarios de la barra de navegación.
    2. El actor presiona el botón recorridos.
    3. El sistema redirige al actor a la página de recorridos
    4. El actor hace clic en la opción seleccionable incidentes en la parte superior.
    5. El sistema redirige al actor a la página de incidentes, recupera todos los registros de la tabla incidentes y los muestra.

3.1.12.2.1. F10.2.1: Consulta general

- **Actor:** Supervisor, Personal
- **Descripción:** Obtener el listado completo de incidentes registrados.
- **Flujo principal:**
    1. El actor selecciona la opción itinerarios de la barra de navegación.
    2. El actor presiona el botón recorridos.
    3. El actor hace clic en la opción seleccionable incidentes en la parte superior.
    4. El sistema redirige al actor a la página de incidentes, recupera todos los registros de la tabla incidentes y los muestra.

3.2. Requerimientos No Funcionales

Tabla 6.
Modelo Conceptual

ID	Tipo	Descripción del requerimiento no funcional
NF1	Usabilidad	El sistema debe proporcionar una interfaz simple que permita al usuario navegar por las opciones del menú, formularios claros con indicaciones explícitas.
NF2	Validación	El sistema debe verificar obligatoriamente que todos los campos requeridos de los formularios sean llenados y que los datos ingresados cumplan con el formato definido antes de ser almacenados en la base de datos.
NF3	Seguridad	El sistema debe requerir obligatoriamente la autenticación mediante usuario y contraseña únicos para cada usuario autorizado, previniendo accesos no autorizados al sistema y garantizando la privacidad de los datos gestionados.

En la tabla se puede apreciar los requisitos no funcionales obtenidos del usuario en reuniones.

CAPÍTULO 4. Diseño y desarrollo de la aplicación web

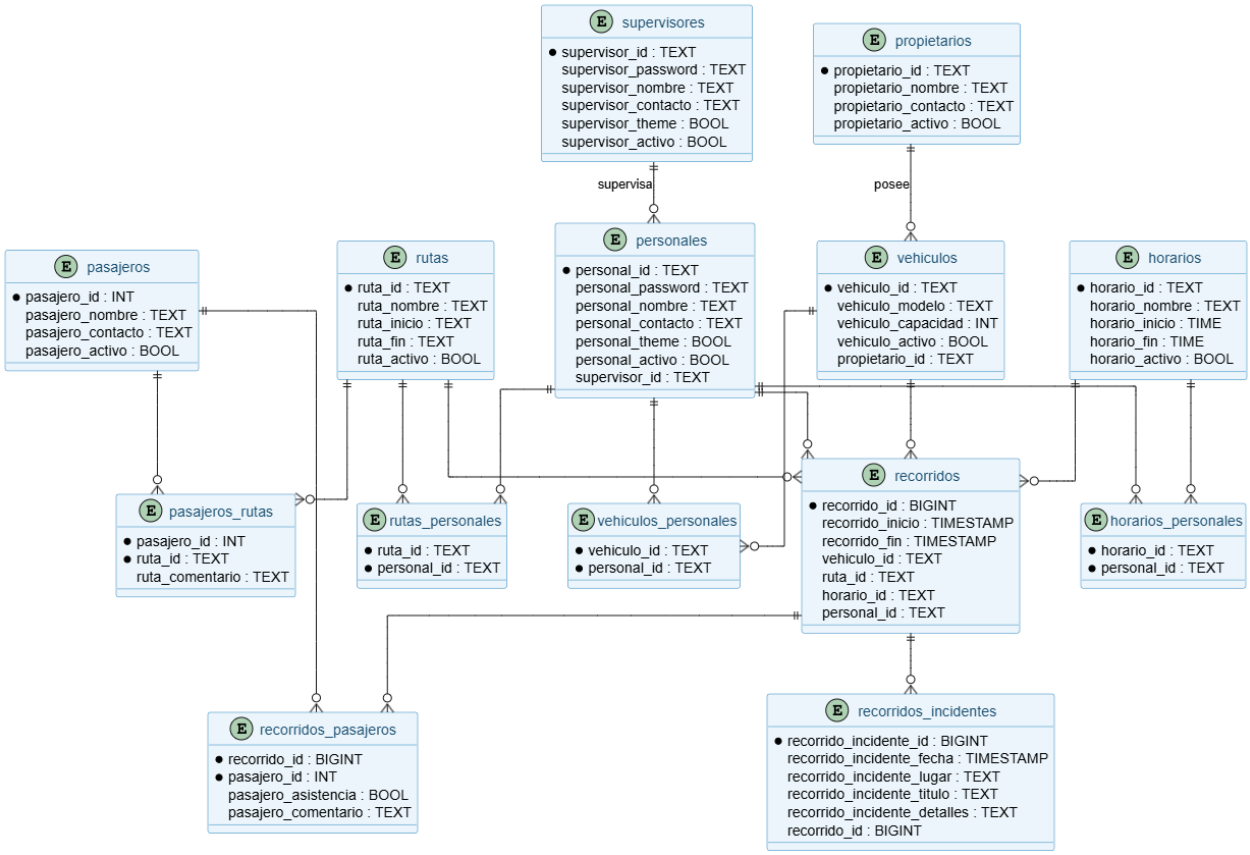
4.1. Introducción

En este capítulo se muestra los modelos de datos en sus niveles conceptual, lógico y físico; así como las pruebas unitarias. De este modo se proporciona una visión clara y coherente de cómo materializar los casos de uso y los requisitos en un sistema.

4.2. Modelado de base de datos

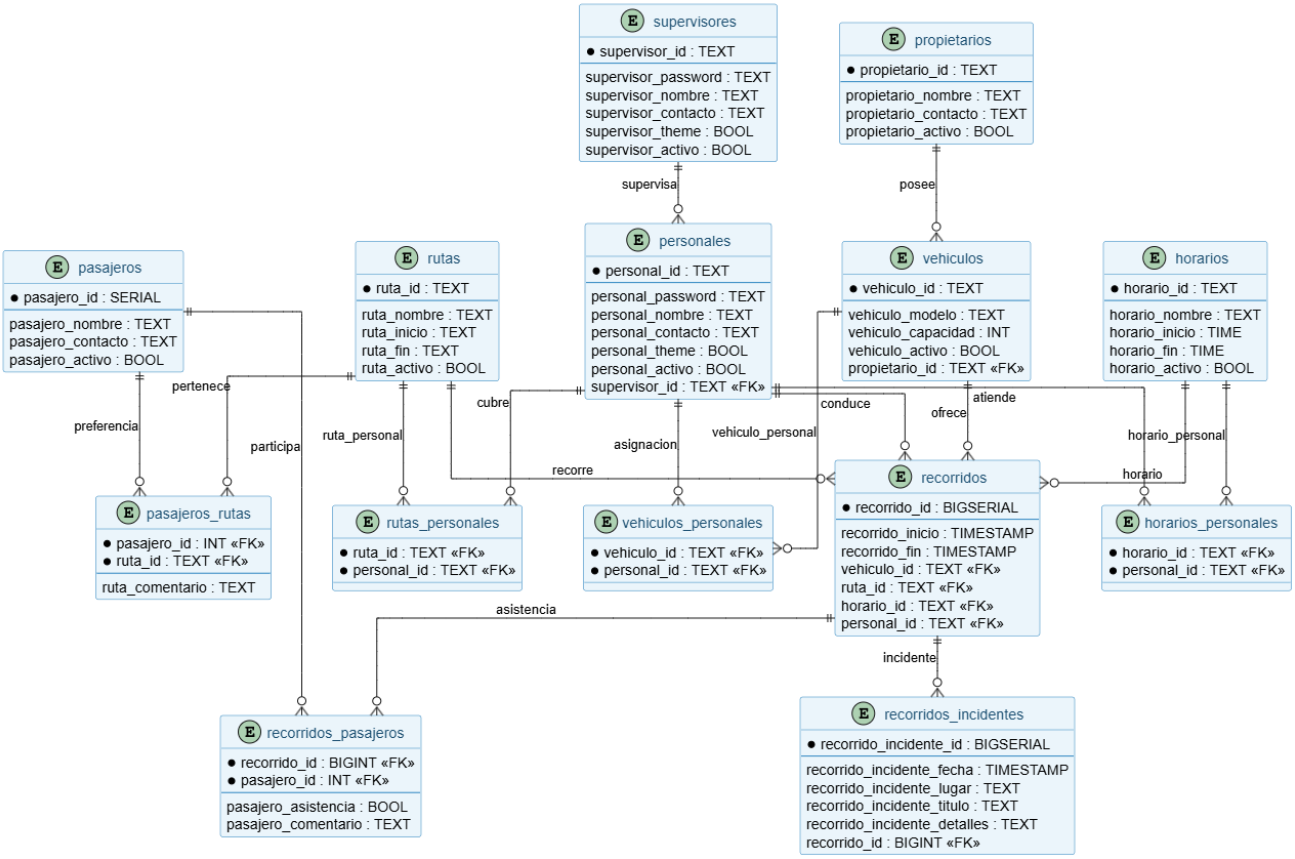
4.2.1. Modelo conceptual

Figura 19.
Modelo Conceptual



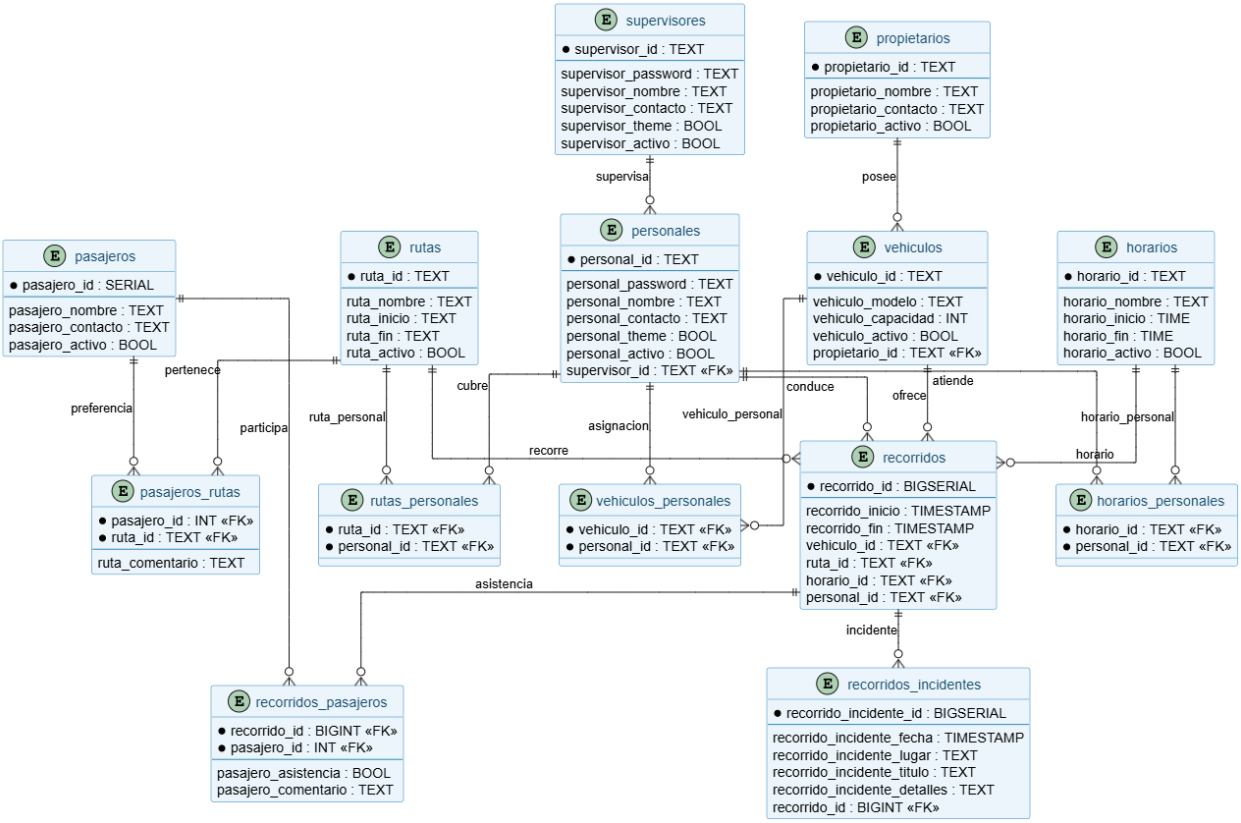
4.2.2. Modelo lógico

Figura 20.
Modelo lógico



4.2.3. Modelo físico

Figura 21.
Modelo Físico



4.3. Desarrollo de la aplicación web

4.3.1. Cronograma de hitos

Tabla 7.
Cronograma de hitos

Hito	Fecha inicio	Fecha fin	Módulos cubiertos	Observación
H1	23-abr-25	30-abr-25	F1 Control de usuarios	inicio y cierre de sesión con validación; sesión en cookie simple
H2	01-may-25	05-may-25	F2 Supervisores y F3 Propietarios	Confirmación antes de borrar, Validación del formulario
H3	06-may-25	14-may-25	F4 Personal y F5 Vehículos	Personal vinculado a Supervisor, Vehículo vinculado a Propietario y Conductor
H4	15-may-25	23-may-25	F6 Rutas y F7 Horarios	Ruta enlazada a Vehículo, Horario valida hora inicio < hora fin, incluir campos requeridos en formulario
H5	26-may-25	06-jun-25	F8 Pasajeros, F9 Recorridos y F10 Incidentes	Flujo iniciar/terminar recorrido y reporte de incidente asociado

En la tabla se puede apreciar el cronograma de hitos del proyecto, incluyendo observaciones.

4.3.2. Notas por hito

4.3.2.1. Hito 1

- **Hecho:** Login y cierre de sesión funcionan con credenciales de supervisores y personal.
- **Próximo:** Ajustar mensaje “Contraseña incorrecta”.
- **Bloqueo:** Ninguno.

4.3.2.2. Hito 2

- **Hecho:** Altas, ediciones y bajas de supervisores y propietarios validadas.
- **Próximo:** Automatizar la inserción del supervisor que crea al propietario en base a la sesión actual, usuario solicita validación de la cédula de propietarios y supervisores.
- **Bloqueo:** Consulta por parámetro por supervisor aún no se aplica.

4.3.2.3. Hito 3

- **Hecho:** CRUD Personal terminado; Vehículos conectados con propietario y personal.
- **Próximo:** Revisar estilos básicos de formularios, usuario requiere más de un personal por vehículo.
- **Bloqueo:** La consulta por parámetro por personal que maneja el vehículo no funciona.

4.3.2.4. Hito 4

- **Hecho:** Rutas enlazado a personal; horarios registrados con verificación de hora, la ruta puede tener más de un personal.
- **Próximo:** usuario pide mejorar distribución de interfaz.
- **Bloqueo:** Validar al guardar horario cuando horas se capturan en formato diferente al del sistema y validar que la hora de inicio sea menor a la hora fin.

4.3.2.5. Hito 5

- **Hecho:** Pasajeros, recorridos e incidentes
- **Próximo:** Registro de incidentes incompleto.
- **Bloqueo:** validación de formato TIMESTAMP para inicio y fin de recorrido, además corregir filtros de búsqueda por formato TIMESTAMP.

4.3.3. Diseño de la interfaz

En esta sección se mostrará evidencias del diseño de la interfaz de cada funcionalidad

Figura 22.
Inicio de sesión

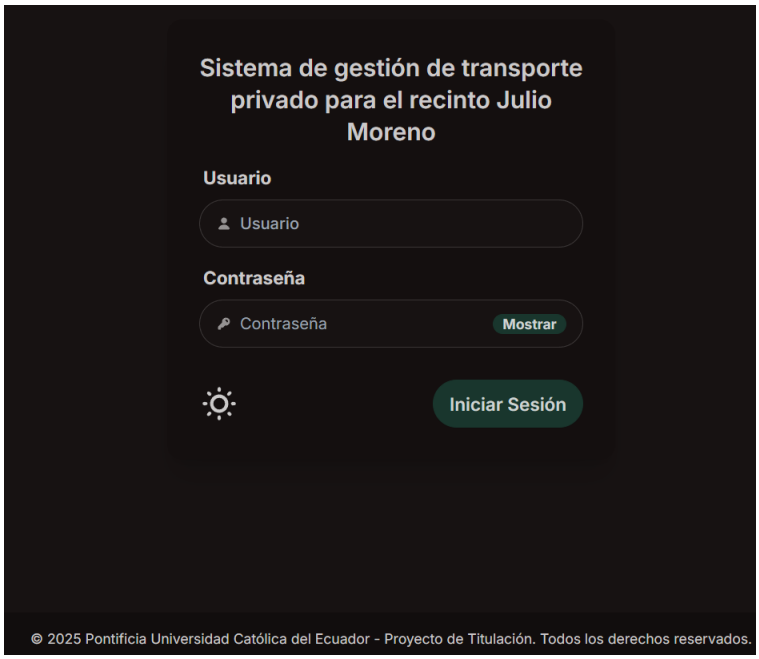


Figura 23.
Página principal

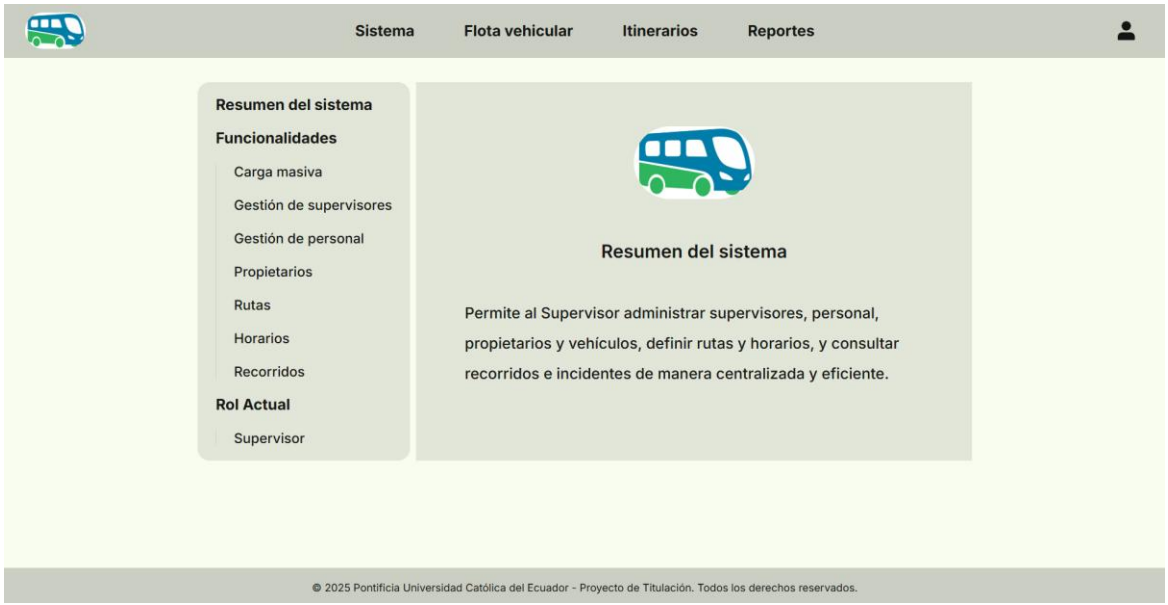


Figura 24.
Supervisores



Figura 25.
Crear supervisor

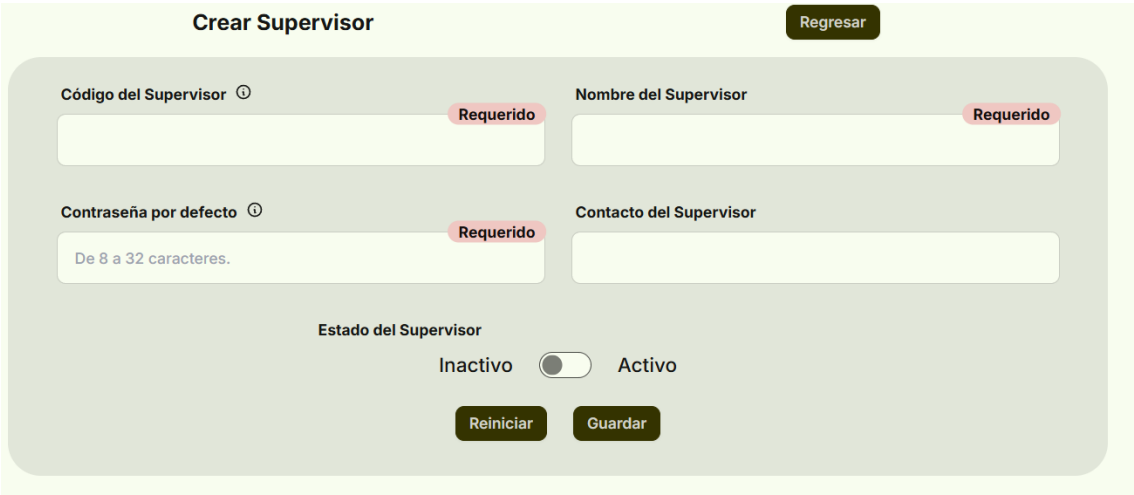


Figura 26.
Modificar supervisor

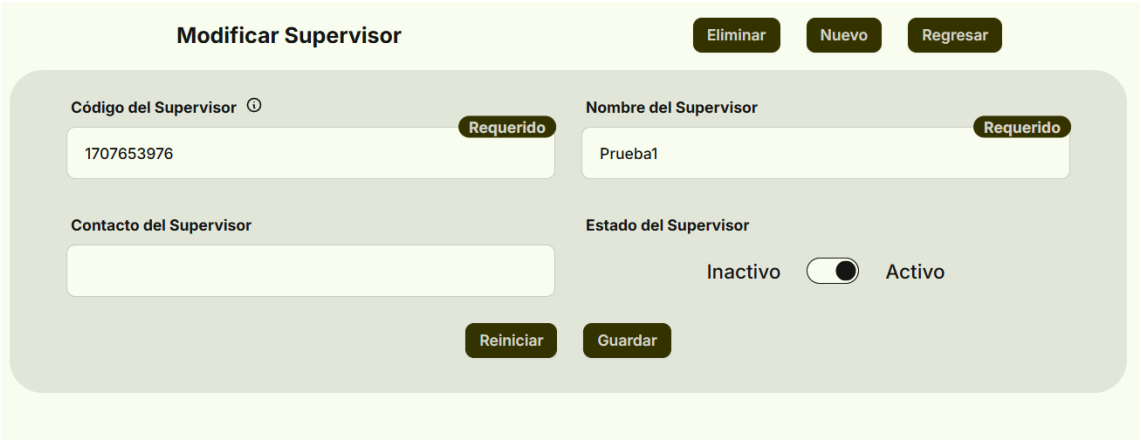


Figura 27.
Propietarios



Figura 28.
Crear propietario

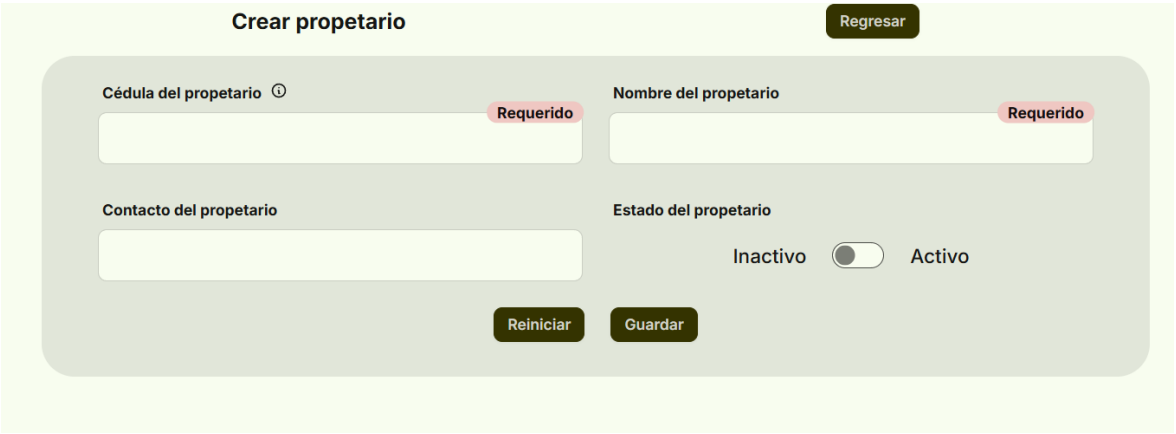


Figura 29.
Modificar propietario

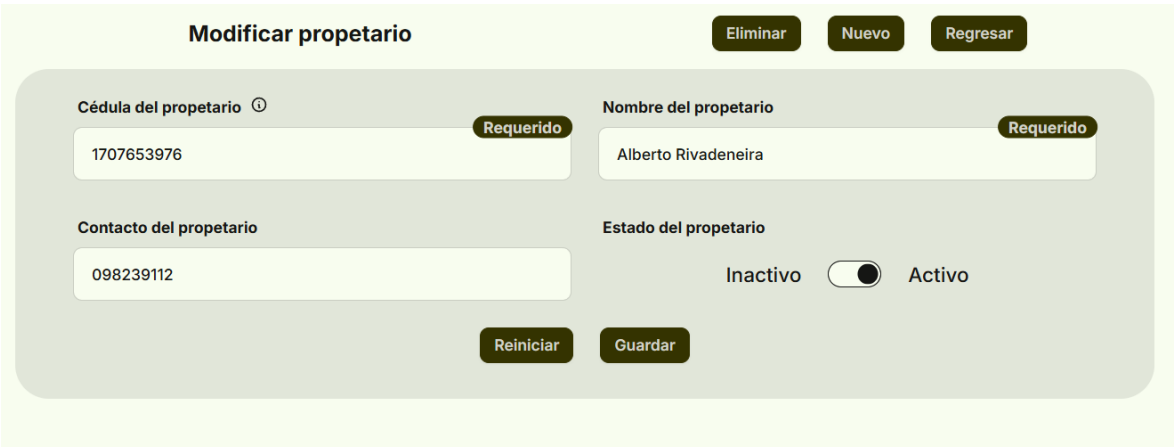


Figura 30.
Personales

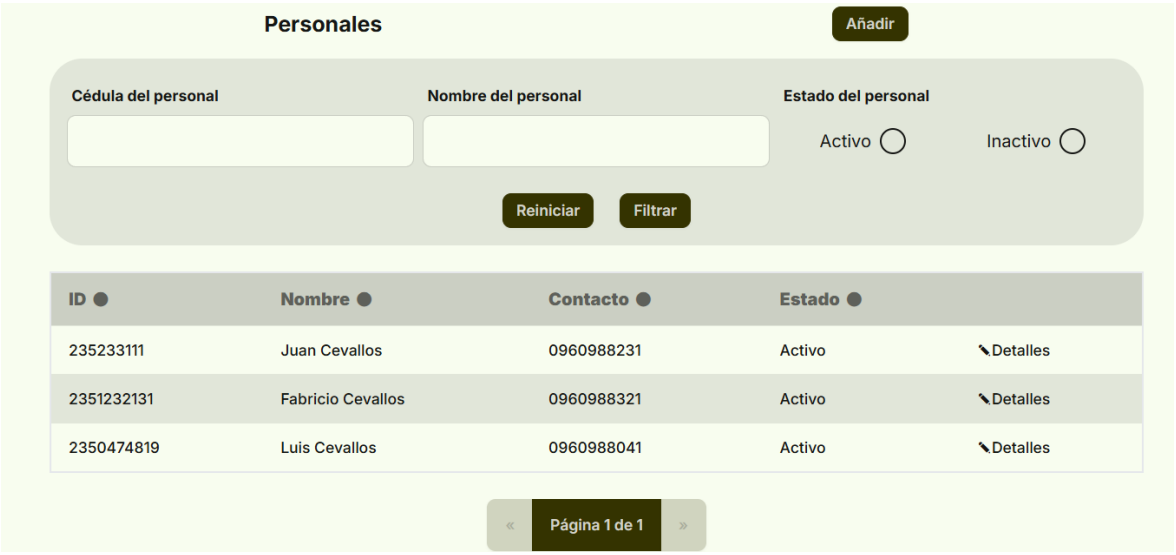


Figura 31.
Crear personal



Figura 32.
Modificar personal

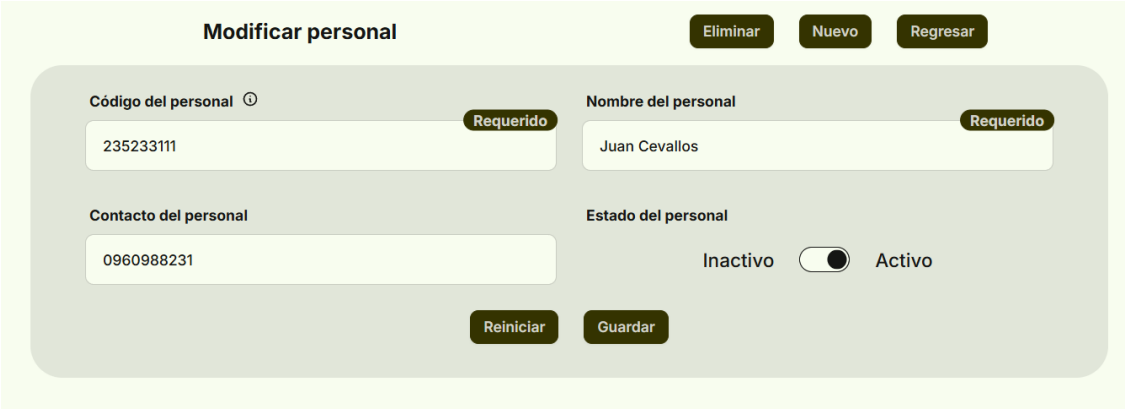


Figura 33.
Vehículos



Figura 34.
Crear vehículo



Figura 35.
Modificar vehículo



Figura 36.
Rutas



Figura 37.
Crear ruta



Figura 38.
Modificar ruta



Figura 39.
Horarios



Figura 40.
Crear horario



Figura 41.
Modificar horario



Figura 42.
Pasajeros



Figura 43.
Crear pasajero

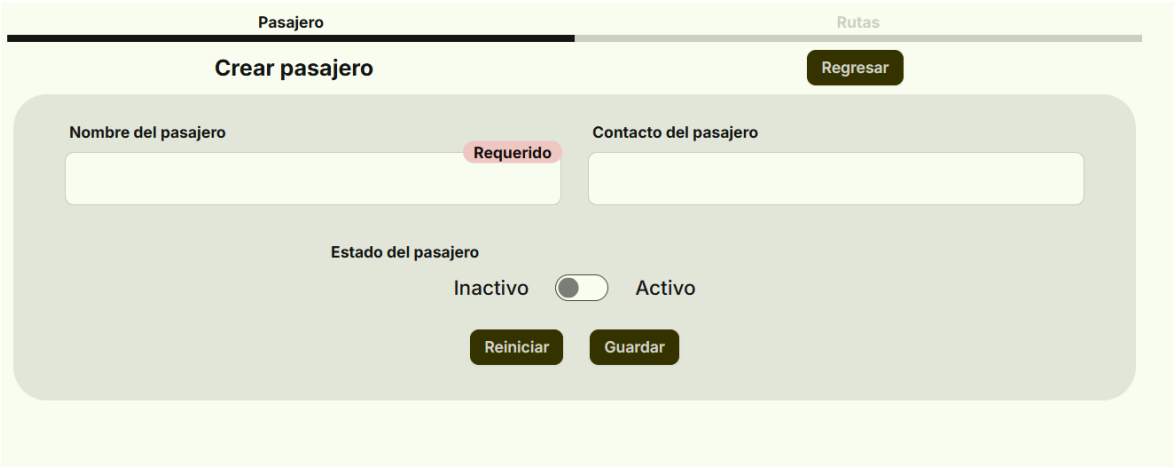


Figura 44.
Modificar pasajero

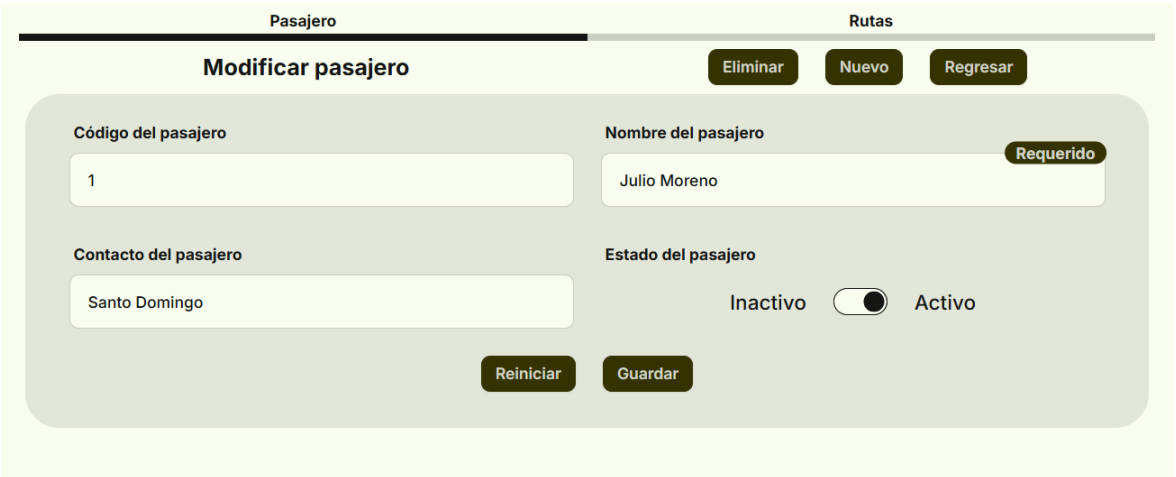
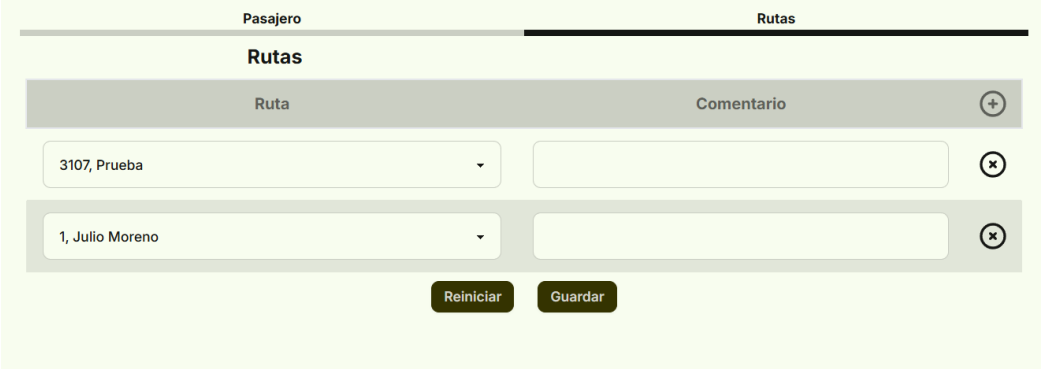


Figura 45.
Asignar rutas a pasajero



4.4. Pruebas unitarias del sistema

4.4.1. Hito 1

Fecha de ejecución: 30 de abril de 2025

Probado por: Mateo Cevallos

4.4.1.1. F1 Control de Usuarios

Tabla 8.
Tabla de pruebas F1

Prueba ID	Funcionalidad	Observaciones
P001	Iniciar sesión	Se inició la sesión correctamente
P002	Cerrar sesión	La sesión se cerró correctamente
P003	Login inválido	Se mostró mensaje de error esperado
P004	Cookie de sesión	Cookie creada y eliminada correctamente

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 46.
Panel de Inicio de Sesión

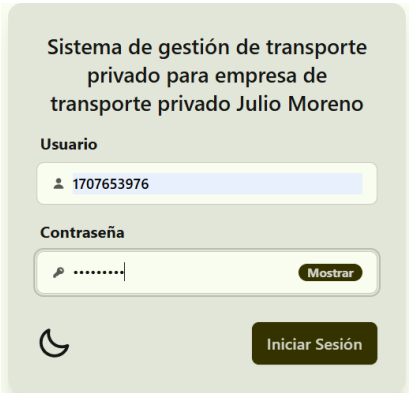


Figura 47.
Página Principal posterior al inicio de sesión



Figura 48.
Página de información general del usuario

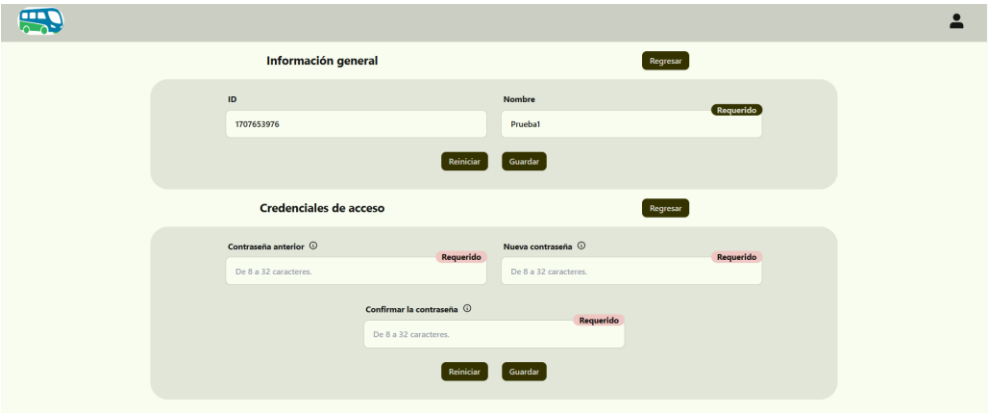


Figura 49.

Panel de inicio de sesión posterior al cierre de sesión



Figura 49.
Mensaje de error posterior al ingreso de credenciales inválidas o inexistentes



Figura 50.
Creación de la cookie posterior al inicio de sesión

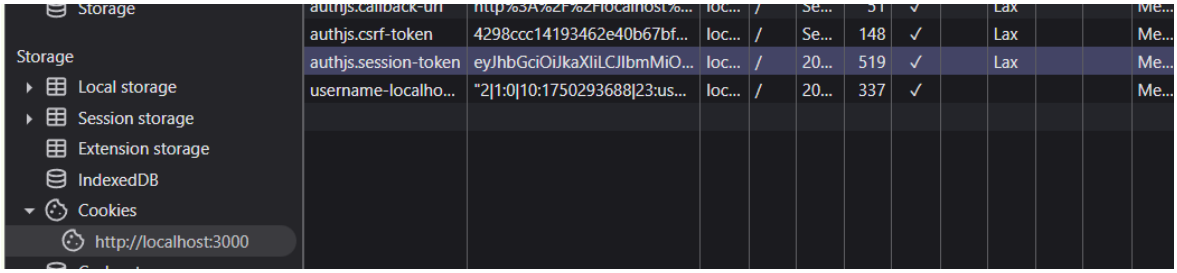
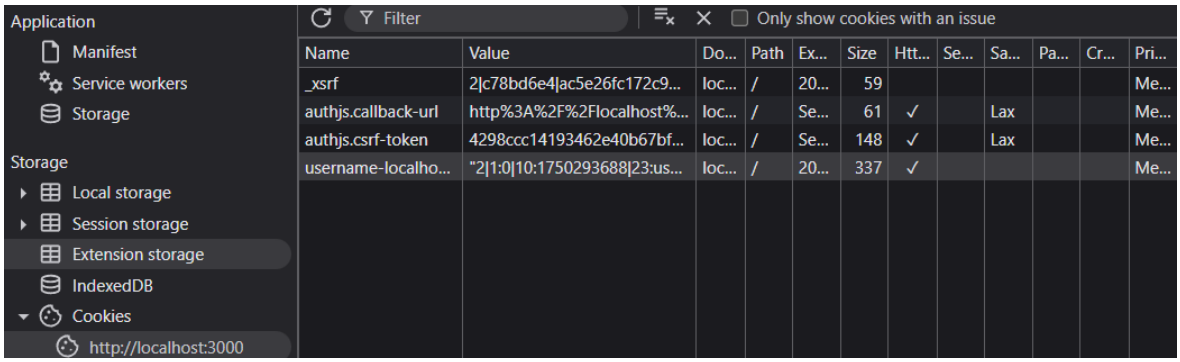


Figura 51.
Eliminación de la cookie posterior al cierre de sesión



4.4.1.2. Valoración general del hito

Durante el Hito 1 se validó el flujo de acceso al sistema, asegurando el inicio como el cierre de sesión de forma segura. Todas las pruebas fueron exitosas y demostraron que el sistema gestiona correctamente la autenticación de usuarios, incluyendo el control de cookies de sesión.

4.4.2. Hito 2

Fecha de ejecución: 05 de mayo de 2025

Probado por: Mateo Cevallos

4.4.2.1. F2 Gestión de supervisores

Tabla 9.
Tabla de pruebas F2

Prueba ID	Funcionalidad	Observaciones
P001	Insertar supervisor	Supervisor registrado correctamente
P002	Consultar supervisor	Resultados mostrados según parámetros de búsqueda
P003	Modificar supervisor	Datos modificados y reflejados en la vista
P004	Eliminar supervisor	Eliminación realizada con confirmación de usuario

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 52.
Creación del supervisor

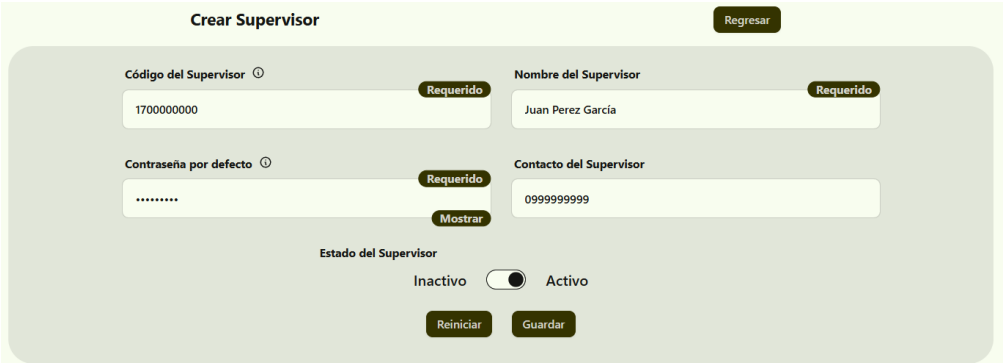


Figura 53.
Consulta que muestra al supervisor creado



Figura 54.
Búsqueda por filtro de supervisores



Figura 55.
Modificación del supervisor

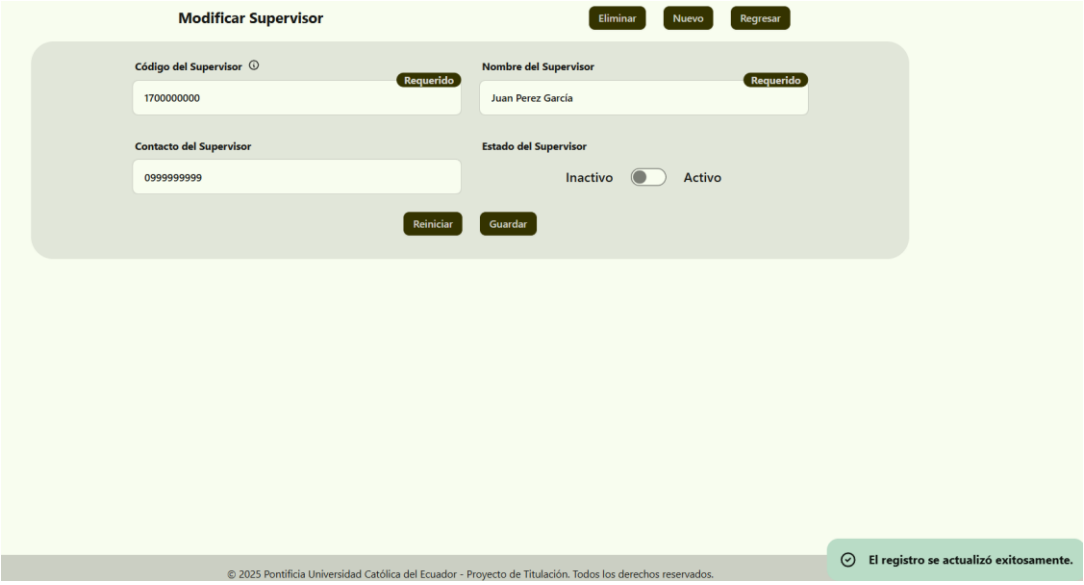


Figura 56.
Eliminación del supervisor



Figura 57.
Consulta que prueba la eliminación del supervisor



4.4.2.2. F3 Gestión de propietarios

Tabla 10.
Tabla de pruebas F3

Prueba ID	Funcionalidad	Observaciones
P005	Insertar propietario	Propietario registrado y listado actualizado
P006	Consultar propietario	Búsqueda efectiva y datos filtrados correctamente
P007	Modificar propietario	Datos actualizados y persistentes en la tabla
P008	Eliminar propietario	Eliminación ejecutada tras confirmación del usuario

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 58.
Creación del propietario

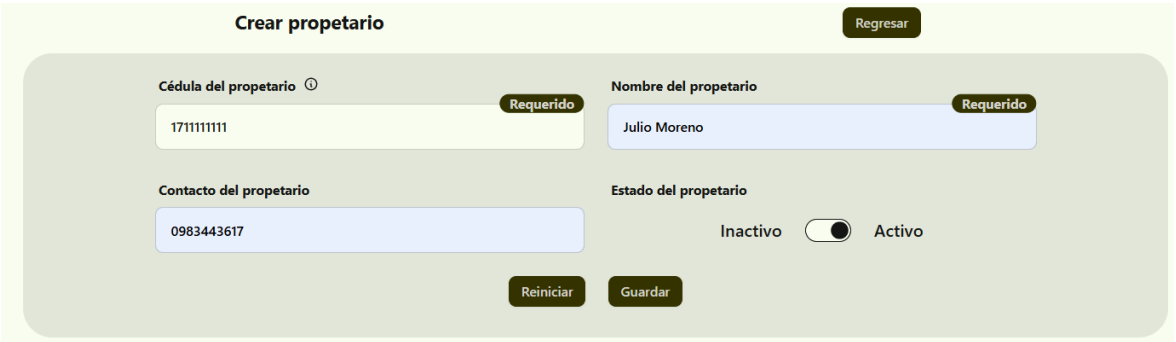


Figura 59.
Consulta que muestra al propietario creado



Figura 60.
Búsqueda por filtros en propietarios



Figura 61.
Modificación del propietario

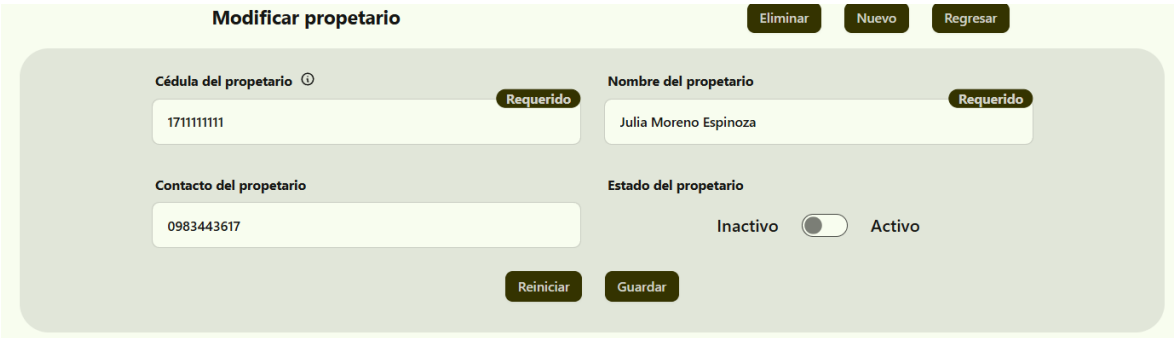


Figura 62.
Consulta que prueba la modificación al propietario

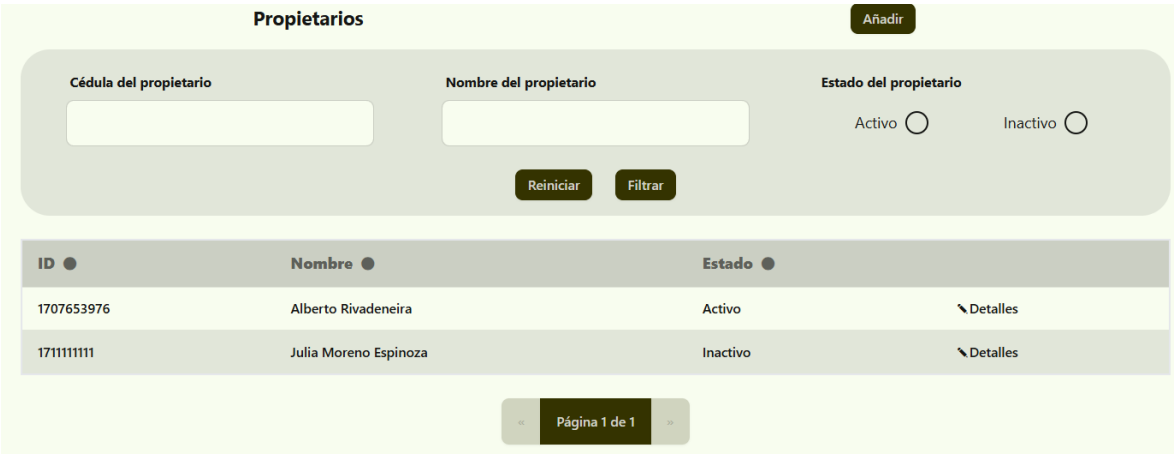


Figura 63.
Eliminación del propietario



4.4.2.3. Valoración general del hito

Durante el Hito 2 se validaron todas las operaciones CRUD sobre los módulos de supervisores y propietarios. Las pruebas demostraron que el sistema gestiona correctamente la inserción, consulta, modificación y eliminación con validaciones activas y confirmaciones cuando es necesario. Todas las funcionalidades evaluadas fueron aprobadas sin errores.

4.4.3. Hito 3

Fecha de ejecución: 14 de mayo de 2025

Probado por: Mateo Cevallos

4.4.3.1. F4 Gestión de personal

Tabla 11.
Tabla de pruebas F4

Prueba ID	Funcionalidad	Observaciones
P009	Insertar personal	Personal registrado correctamente y vinculado a supervisor
P010	Consultar personal	Consulta general y filtrada mostró datos esperados
P011	Modificar personal	Datos actualizados correctamente
P012	Eliminar personal	Personal eliminado tras confirmación del usuario

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 64.
Creación del personal



Figura 65.
Consulta que prueba la creación del personal



Figura 66.
Modificación del personal

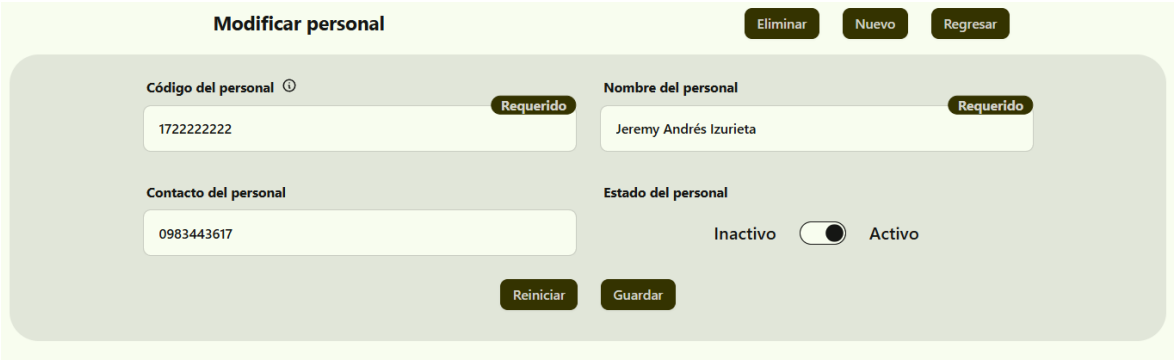


Figura 67.
Consulta que valida la modificación del personal

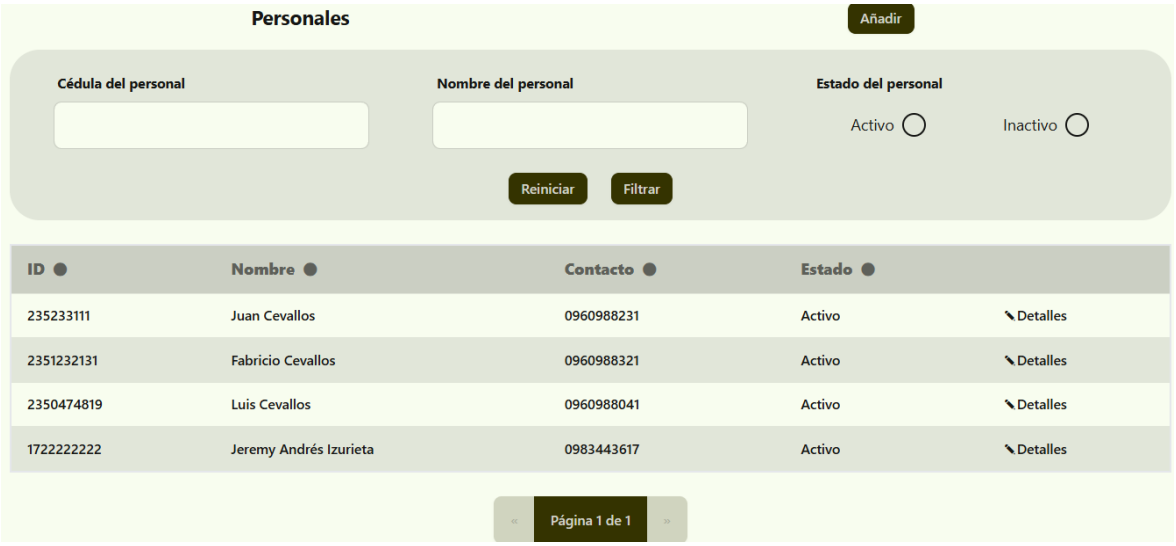


Figura 68.
Búsqueda por filtros en personales

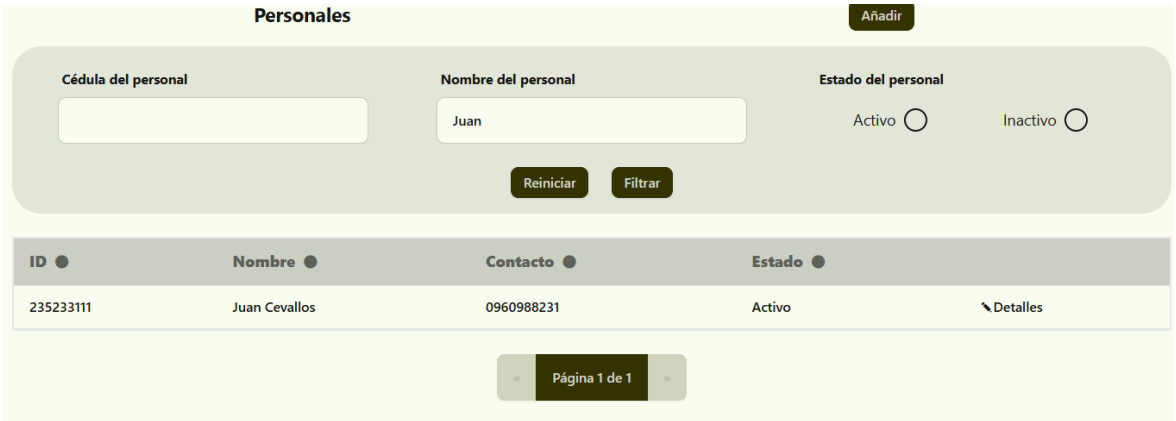
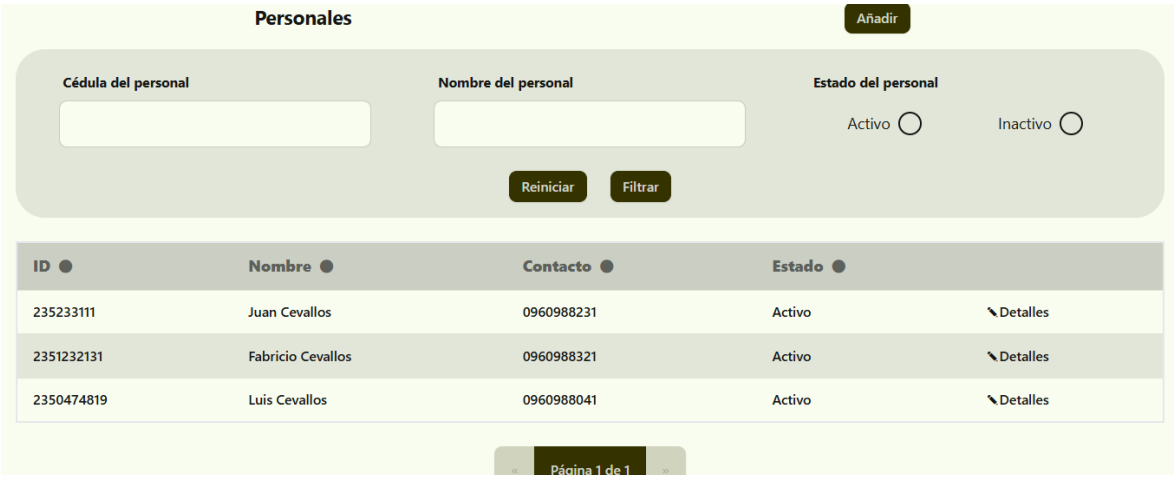


Figura 69.
Eliminación del personal



Figura 70.
Consulta que prueba la eliminación del propietario



4.4.3.2. F5 Gestión de vehículos

Tabla 12.
Tabla de pruebas F5

Prueba ID	Funcionalidad	Observaciones
P013	Insertar vehículo	Vehículo registrado y vinculado a propietario y conductor
P014	Consultar vehículo	Lista mostrada correctamente con filtros aplicados
P015	Modificar vehículo	Información actualizada en la base de datos
P016	Eliminar vehículo	Eliminación realizada correctamente con confirmación

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 71.
Creación del vehículo



Figura 72.
Consulta que prueba la creación del vehículo



Figura 73.
Búsqueda por filtros de vehículos



Figura 74.
Modificación de vehículos



Figura 75.
Asignación de personal al vehículo seleccionado



Figura 76.
Consulta que prueba la modificación del vehículo

ID	Modelo	Capacidad	Estado	Propietario	
PCD-3128	Toyota Hiace 2019	30	Activo	1801234567	Detalles
PCD-4129	Mercedes Sprinter 20	20	Activo	1801234568	Detalles
PDF_81399	HINO DHA 14	10	Activo	juan	Detalles
PCD-5130	Nissan Urvan 2018 SSS	12	Activo	1801234569	Detalles

4.4.3.3. Valoración general del hito

Durante el Hito 3 se evaluaron los módulos de personal y vehículos. Se confirmó que las relaciones con otras entidades (supervisores y propietarios) se gestionaron adecuadamente. Cada acción de inserción, modificación, consulta y eliminación respondió sin errores y con evidencia verificable en la interfaz del sistema.

4.4.4. Hito 4

Fecha de ejecución: 23 de mayo de 2025

Probado por: Mateo Cevallos

4.4.4.1. F6 Gestión de rutas

Tabla 13.
Tabla de pruebas F6

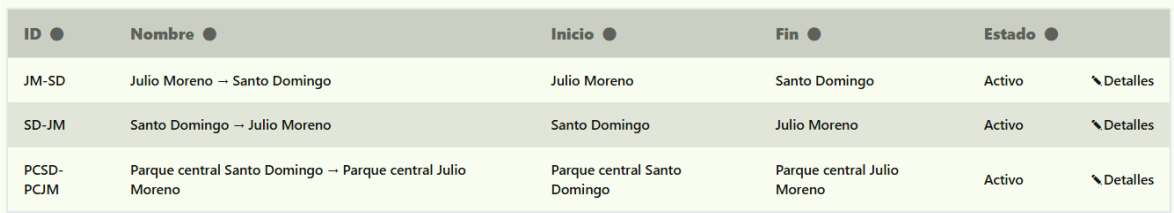
Prueba ID	Funcionalidad	Observaciones
P017	Insertar ruta	Ruta creada exitosamente y vinculada a un vehículo
P018	Consultar ruta	Resultados mostrados correctamente según búsqueda
P019	Modificar ruta	Cambios guardados y reflejados en la interfaz
P020	Eliminar ruta	Eliminación realizada correctamente con confirmación

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 77.
Creación de ruta



Figura 78.
Consulta que prueba la creación de la ruta.



ID	Nombre	Inicio	Fin	Estado	
JM-SD	Julio Moreno → Santo Domingo	Julio Moreno	Santo Domingo	Activo	Detalles
SD-JM	Santo Domingo → Julio Moreno	Santo Domingo	Julio Moreno	Activo	Detalles
PCSD-PCJM	Parque central Santo Domingo → Parque central Julio Moreno	Parque central Santo Domingo	Parque central Julio Moreno	Activo	Detalles

Figura 79.
Modificación de la ruta

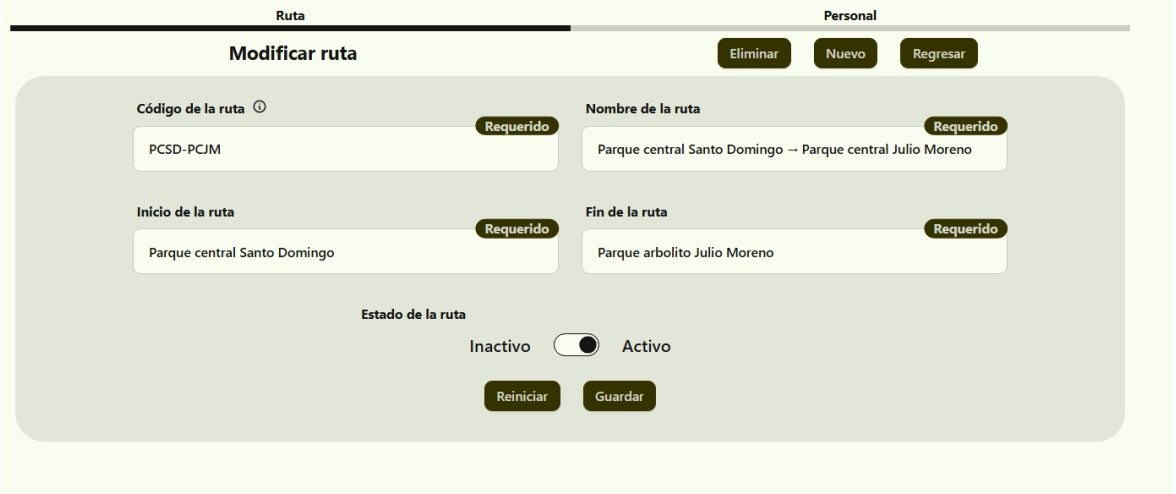


Figura 80.
Consulta que prueba la modificación de la ruta



Figura 81.
Asignación del personal a la ruta

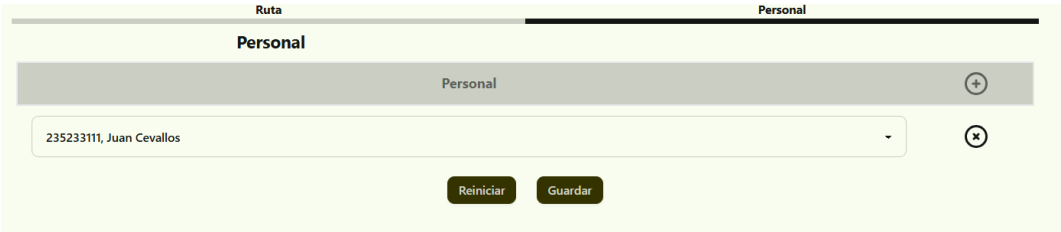
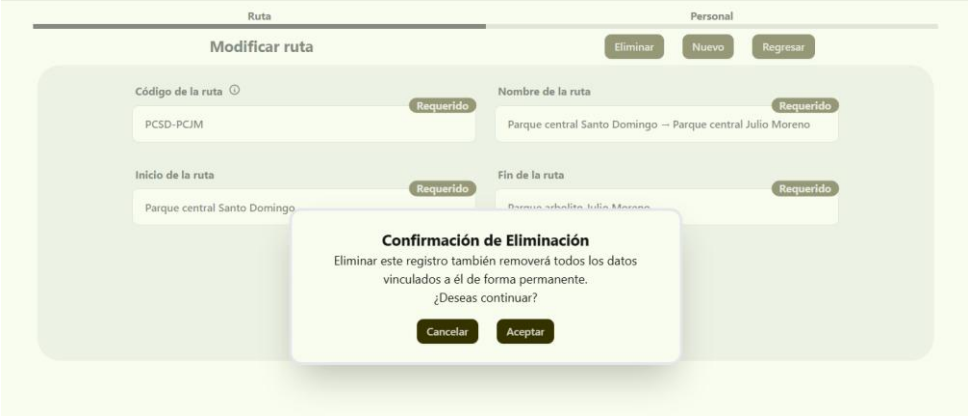


Figura 82.
Eliminación de la ruta



4.4.4.2. F7 Gestión de horarios

Tabla 14.
Tabla de pruebas F7

Prueba ID	Funcionalidad	Observaciones
P021	Insertar horario	Horario creado correctamente asociado a una ruta
P022	Validación horas	Error mostrado si hora de inicio es mayor que hora de fin
P023	Consultar horario	Lista mostrada correctamente con filtros aplicados
P024	Eliminar horario	Eliminación confirmada y datos removidos de forma segura

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo

Figura 83.
Creación del horario



Figura 84.
Consulta que prueba la creación del horario

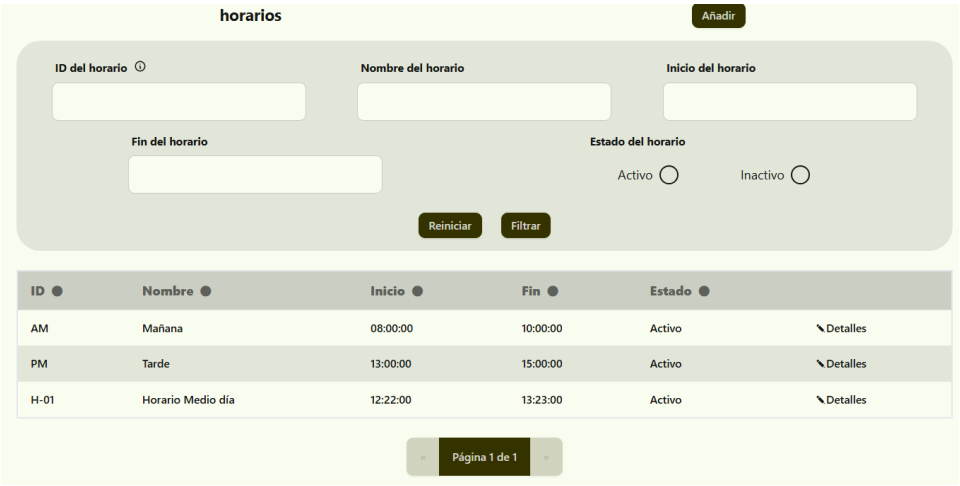


Figura 85.
Modificación de horario



Figura 86.
Asignación de personal con horarios

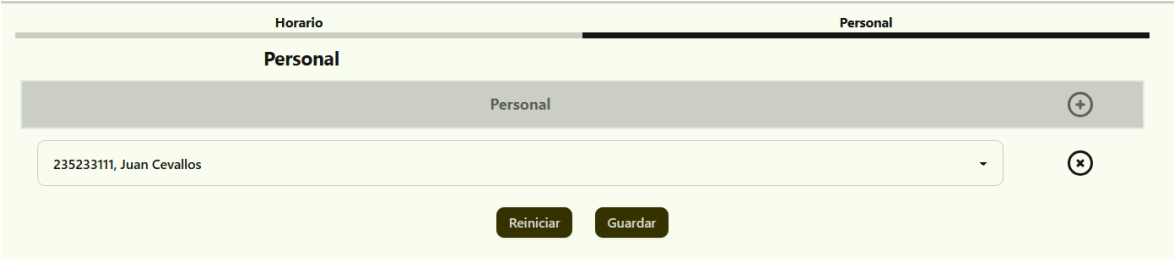


Figura 87.
Eliminación del horario



Figura 88.
Consulta que prueba la eliminación del horario



4.4.4.3. Valoración general del hito

Durante el Hito 4 se verificó el correcto funcionamiento del sistema en la administración de rutas y horarios. Se validaron relaciones con vehículos, reglas lógicas (como validación de horario) y la manipulación de los registros desde la interfaz. Todas las acciones fueron ejecutadas con éxito, cumpliendo los objetivos de la iteración.

4.4.5. HITO 5

Fecha de ejecución: 06 de junio de 2025

Probado por: Mateo Cevallos

4.4.5.1. F8 Gestión de pasajeros

Tabla 15.
Tabla de pruebas F8

Prueba ID	Funcionalidad	Observaciones
P025	Insertar pasajero	Pasajero registrado correctamente con datos completos
P026	Consultar pasajero	Listado generado correctamente con filtros aplicados
P027	Modificar pasajero	Datos actualizados y reflejados en la interfaz
P028	Eliminar pasajero	Eliminación ejecutada tras confirmación del usuario

Figura 89.
Creación del pasajero



Figura 90.
Consulta que prueba la creación del pasajero



Figura 91.
Búsqueda por filtros de pasajeros

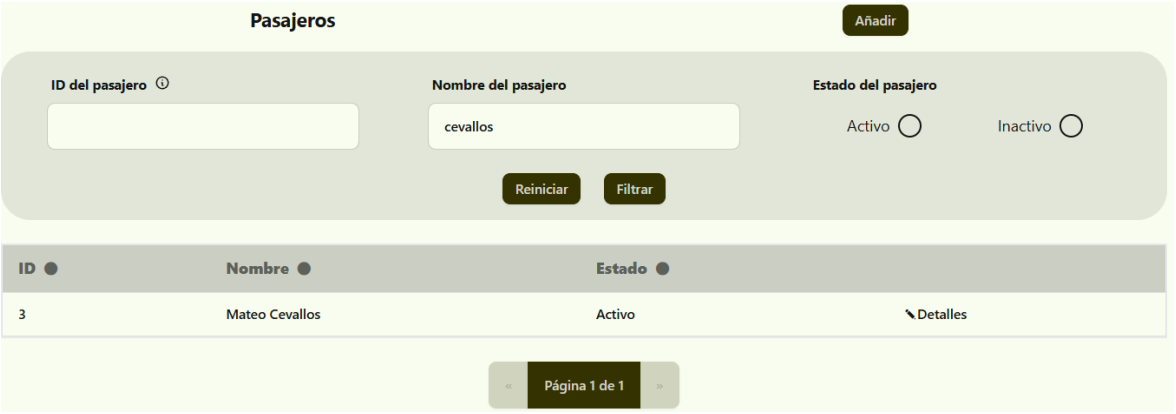


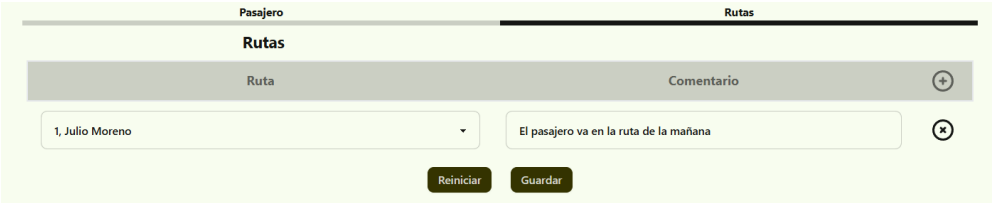
Figura 92.
Modificación del pasajero



Figura 93.
Consulta que prueba la modificación del pasajero



Figura 94.
Asignación de ruta del pasajero



4.4.5.2. F9 Gestión de recorridos

Tabla 16.
Tabla de pruebas F9

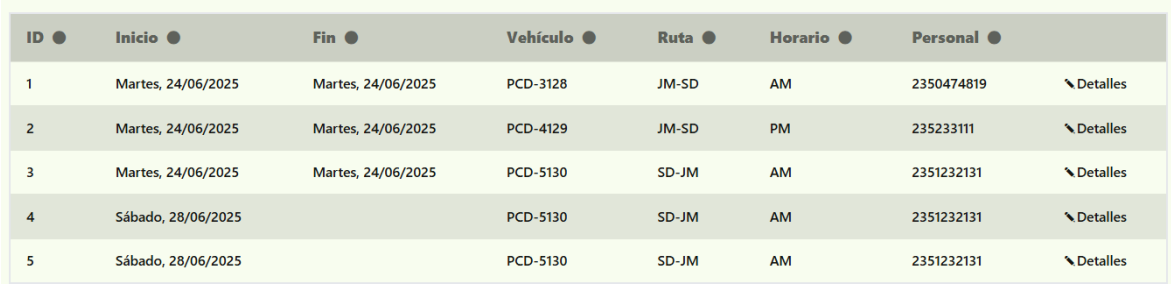
Prueba ID	Funcionalidad	Observaciones
P029	Iniciar recorrido	Recorrido iniciado correctamente desde un horario existente
P030	Finalizar recorrido	Cambio de estado realizado correctamente al finalizar
P031	Consultar recorridos	Lista de recorridos mostrada correctamente
P032	Asociar pasajeros	Pasajeros asignados correctamente al recorrido

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 95.
Creación del recorrido



Figura 96.
Consulta que prueba la creación del recorrido

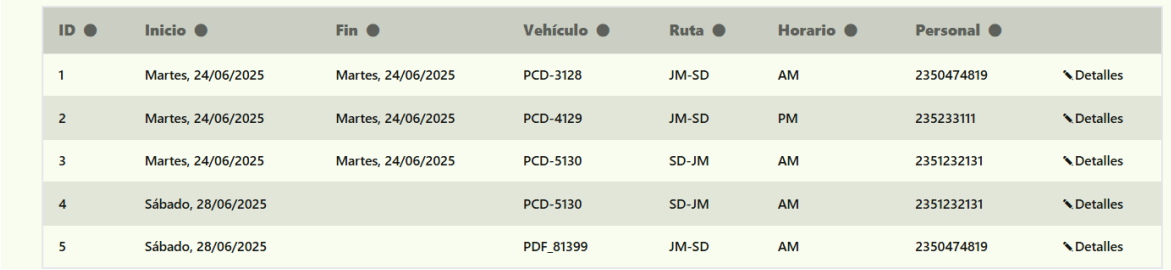


ID	Inicio	Fin	Vehículo	Ruta	Horario	Personal	
1	Martes, 24/06/2025	Martes, 24/06/2025	PCD-3128	JM-SD	AM	2350474819	Detalles
2	Martes, 24/06/2025	Martes, 24/06/2025	PCD-4129	JM-SD	PM	235233111	Detalles
3	Martes, 24/06/2025	Martes, 24/06/2025	PCD-5130	SD-JM	AM	2351232131	Detalles
4	Sábado, 28/06/2025		PCD-5130	SD-JM	AM	2351232131	Detalles
5	Sábado, 28/06/2025		PCD-5130	SD-JM	AM	2351232131	Detalles

Figura 97.
Modificación del recorrido



Figura 98.
Consulta que prueba la modificación del recorrido



ID	Inicio	Fin	Vehículo	Ruta	Horario	Personal	
1	Martes, 24/06/2025	Martes, 24/06/2025	PCD-3128	JM-SD	AM	2350474819	Detalles
2	Martes, 24/06/2025	Martes, 24/06/2025	PCD-4129	JM-SD	PM	235233111	Detalles
3	Martes, 24/06/2025	Martes, 24/06/2025	PCD-5130	SD-JM	AM	2351232131	Detalles
4	Sábado, 28/06/2025		PCD-5130	SD-JM	AM	2351232131	Detalles
5	Sábado, 28/06/2025		PDF_81399	JM-SD	AM	2350474819	Detalles

Figura 99.
Lista de registro de asistencia de pasajeros



4.4.5.3. F10 Gestión de incidentes

Tabla 17.
Tabla de pruebas F10

Prueba ID	Funcionalidad	Observaciones
P033	Reportar incidente	Incidente registrado correctamente y vinculado a recorrido
P034	Consultar incidentes	Resultados mostrados correctamente
P035	Eliminar incidente	Eliminación confirmada desde la vista del detalle

En la tabla se puede apreciar las funcionalidades probadas con su id respectivo.

Figura 100.
CRUD incidentes

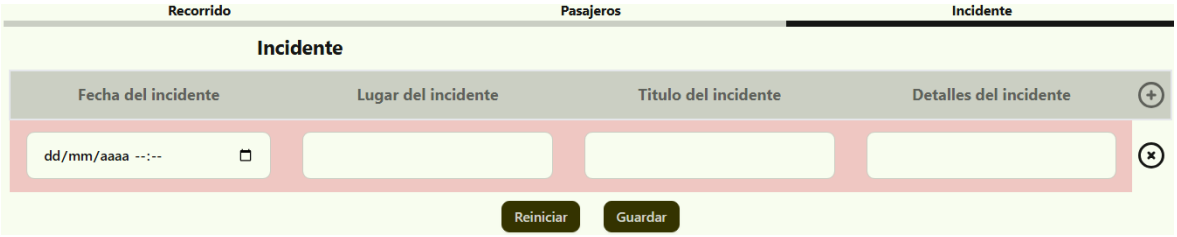
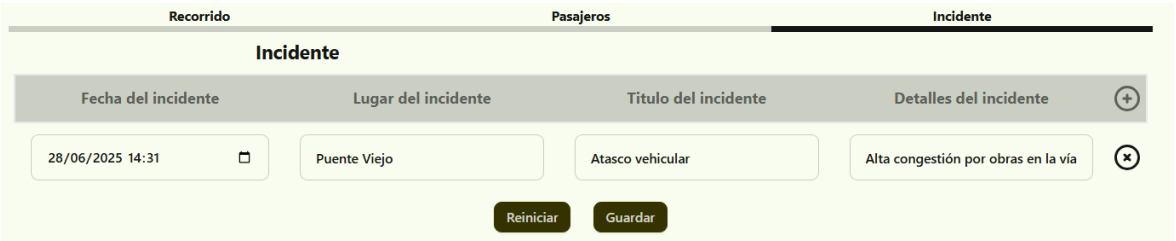


Figura 101.
Creación de incidente



4.4.5.4. Valoración general del hito

Durante el Hito 5 se probaron los procesos finales relacionados con el seguimiento de pasajeros, ejecución de recorridos y manejo de incidentes. Se verificó la lógica completa del flujo desde el inicio de un recorrido hasta el cierre, con soporte para asignaciones y eventos relacionados. Todas las funcionalidades respondieron como se esperaba, cerrando el ciclo funcional del sistema de transporte

4.5. Aplicación de la metodología seleccionada

4.5.1. Software funcional entregable

El desarrollo del sistema se organizó en cinco hitos de entrega, cada uno enfocado en un conjunto de módulos funcionales. Cada hito representó una fase completa y validada del sistema, mediante retroalimentación del cliente.

4.5.1.1. Hitos definidos:

- **Hito 1 (23-abr al 30-abr):** Inicio y cierre de sesión con validación en cookie.
- **Hito 2 (01-may al 05-may):** Módulos de supervisores y propietarios con validación y confirmación de eliminación.
- **Hito 3 (06-may al 14-may):** Módulos de personal (vinculado a supervisor) y vehículos (vinculados a propietario y conductor).
- **Hito 4 (15-may al 23-may):** Módulos de rutas (enlazadas a vehículo) y horarios (validación de horas y campos requeridos).
- **Hito 5 (26-may al 06-jun):** Módulos de pasajeros, recorridos (iniciar/finalizar) e incidentes (registro vinculado a recorrido).

4.5.2. Talleres de Reflexión

Descripción: Al finalizar cada hito, se llevó a cabo una reflexión sobre el proceso y los resultados obtenidos. Esto permitió mejorar continuamente la forma de trabajo.

4.5.2.1. Hito 1

Fecha de inicio: 23-04-2025
Fecha de finalización: 30-04-2025

4.5.2.1.1. ¿Qué salió bien?

- Se logró implementar correctamente el inicio y cierre de sesión.
- La validación de credenciales y el uso de cookies funcionó sin errores.

4.5.2.1.2. ¿Qué podría mejorar?

- El diseño del mensaje de error podría ser más visible.
- Podría añadirse retroalimentación visual al presionar el botón login.

4.5.2.1.3. ¿Qué haré diferente en la siguiente etapa?

- Probar los flujos con más combinaciones de entradas.
- Verificar respuesta del sistema ante campos vacíos y condiciones límite.

4.5.2.1.4. Comentarios adicionales:

- La gestión de sesiones respondió como se esperaba.
- Fue un buen punto de partida para el resto de los módulos.
- La funcionalidad quedó estable y lista para pruebas con usuarios.

4.5.2.2. Hito 2

Fecha de inicio: 01-05-2025
Fecha de finalización: 05-05-2025

4.5.2.2.1. ¿Qué salió bien?

- Se completó el CRUD de supervisores y propietarios.
- Las validaciones de campos y confirmaciones funcionaron correctamente.

4.5.2.2.2. ¿Qué podría mejorar?

- Se identificó que algunos formularios podrían optimizarse visualmente. Las alertas podrían ser más descriptivas.

4.5.2.2.3. ¿Qué haré diferente en la siguiente etapa?

- Incluir alertas de éxito al guardar o eliminar registros.
- Aplicar retroalimentación visual más clara.

4.5.2.2.4. Comentarios adicionales:

- Se cumplió con lo planificado para el hito.

- La estructura del módulo quedó clara y reutilizable para los siguientes.

4.5.2.3. Hito 3

Fecha de inicio: 06-05-2025
Fecha de finalización: 14-05-2025

4.5.2.3.1. ¿Qué salió bien?

- Se establecieron correctamente las relaciones entre personal y supervisores, y entre vehículos, conductores y propietarios.

4.5.2.3.2. ¿Qué podría mejorar?

- La respuesta visual tras registrar datos no era siempre clara.
- No todos los formularios tenían confirmación visible.

4.5.2.3.3. ¿Qué haré diferente en la siguiente etapa?

- Agregar mensajes de éxito y error en todas las acciones CRUD.
- Probar los flujos de inserción y modificación con más detalle.

4.5.2.3.4. Comentarios adicionales:

- La complejidad aumentó por la relación entre entidades, pero se resolvió de forma controlada.

4.5.2.4. Hito 4

Fecha de inicio: 15-05-2025
Fecha de finalización: 23-05-2025

4.5.2.4.1. ¿Qué salió bien?

- Se logró vincular rutas con vehículos correctamente.
- La validación de horas en los formularios de horarios respondió correctamente.

4.5.2.4.2. ¿Qué podría mejorar?

- Hubo dificultad inicial para validar condiciones de horario con campos vacíos.

4.5.2.4.3. ¿Qué haré diferente en la siguiente etapa?

- Implementar validaciones preventivas antes de guardar horarios.
- Diseñar formularios con advertencias visibles antes de enviar.

4.5.2.4.4. Comentarios adicionales:

- El módulo fue más técnico que visual, pero se logró implementar con solidez.

4.5.2.5. Hito 5

Fecha de inicio: 26-05-2025

Fecha de finalización: 06-06-2025

4.5.2.5.1. ¿Qué salió bien?

- Se completó el flujo de recorrido con asociación de pasajeros e incidentes.
- Todas las funcionalidades respondieron correctamente al integrar múltiples entidades.

4.5.2.5.2. ¿Qué podría mejorar?

- La vista del recorrido podría mostrar más detalle visual del estado.
- El formulario de incidente podría beneficiarse de un editor de texto enriquecido.

4.5.2.5.3. ¿Qué haré diferente en la siguiente etapa?

- Mejorar los formularios donde se ingresa texto libre.
- Permitir edición de incidentes ya registrados.

4.5.2.5.4. Comentarios adicionales:

- Fue el hito más completo y funcional, cerrando el ciclo esperado por el usuario.
- La validación con el usuario experto confirmó la utilidad práctica del sistema.

4.5.3. Planificación liviana

Se utilizó un esquema tipo tablero para organizar y priorizar tareas de forma visual utilizando Microsoft Planner.

Figura 102.
Tablero de tareas 70%

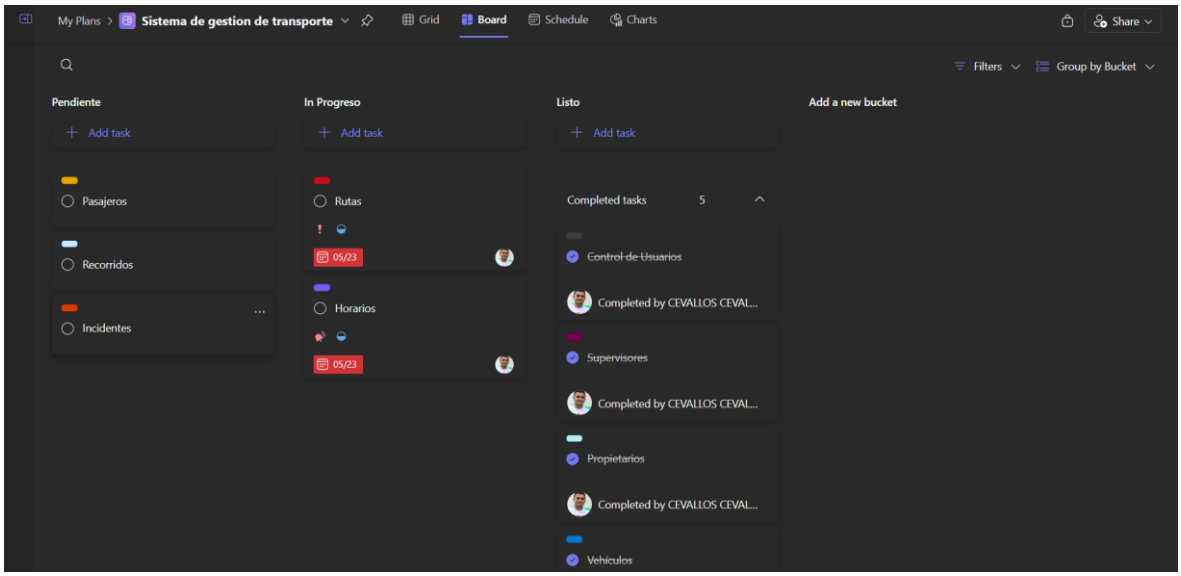
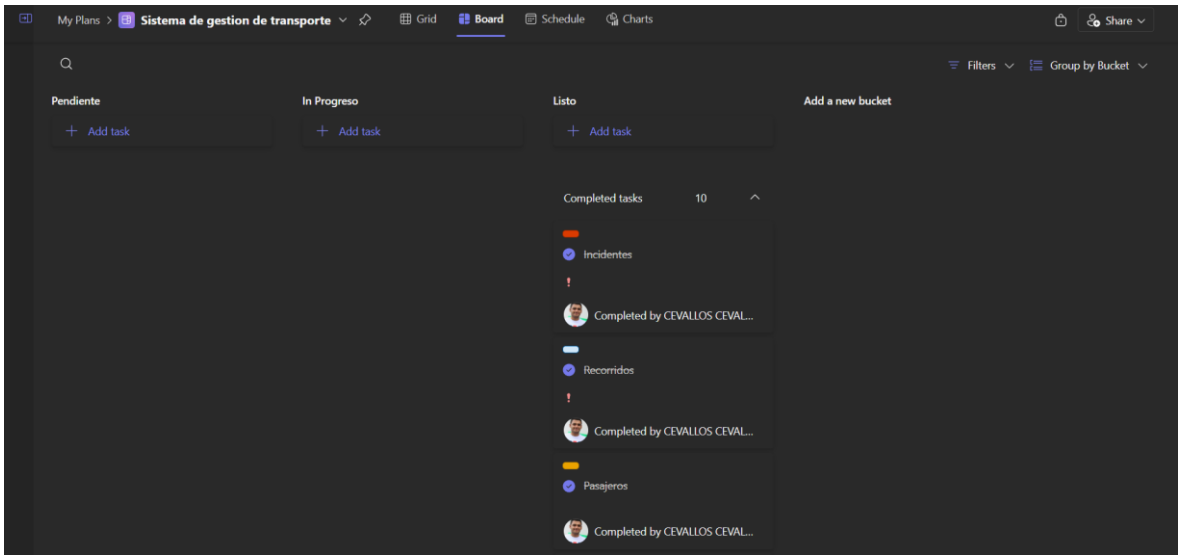


Figura 103.
Tablero de tareas completo



4.5.4. Lista de riesgos conocidos

Se identificaron riesgos potenciales tanto técnicos como de organización. Cada riesgo se evaluó con una medida de prevención o mitigación.

Tabla 18.
Tabla de riesgos conocidos

Riesgo	Severidad	Acción de mitigación
Retraso en desarrollo por carga académica	Alta	Organizar entregas por hitos
Dificultad de pruebas con usuarios reales	Media	Recopilar retroalimentación del usuario experto en cada hito
Pérdida de datos durante desarrollo	Alta	Usar un repositorio para almacenar el proyecto

En la tabla se puede apreciar los riesgos presentes y su acción para mitigarlos el desarrollo.

4.5.5. Roles asumidos durante el proyecto

Descripción: Los roles de la metodología Crystal Clear fueron considerados como guías de responsabilidad.

Tabla 19.
Tabla de roles del proyecto

Rol	Asignado a
Patrocinador	Silvia Suárez
Usuario experto	Silvia Suarez
Diseñador y programador	Mateo Cevallos
Tester y documentador	Mateo Cevallos

En la tabla se puede apreciar los roles del proyecto.

4.5.6. Propiedades esenciales aplicadas

Tabla 20.
Tabla de propiedades de la metodología aplicadas en el proyecto

Propiedad	Aplicación en el proyecto
Entrega frecuente	Cada módulo fue desarrollado y probado por separado.
Mejora reflexiva	Se aplicaron talleres de reflexión tras cada entrega.
Comunicación cercana	Validación directa con supervisor de transporte privado.
Seguridad personal	Se mantuvo un entorno libre para plantear ideas o dudas.
Acceso a usuarios	Validación del sistema con el supervisor real.
Entorno técnico confiable	Pruebas manuales y módulos por separado.
Enfoque sostenido	Se asignaron periodos de tiempo dedicados al desarrollo del sistema.

En la tabla se puede apreciar las propiedades de la metodología que fueron aplicadas durante el proyecto.

Este conjunto de entregables demuestra la implementación de la metodología Crystal Clear durante el desarrollo del sistema de transporte privado, adaptado al contexto académico y técnico del proyecto.

4.5.7. Validación con el usuario experto

Como complemento a las pruebas técnicas ejecutadas por el desarrollador, se realizaron sesiones de revisión funcional con la supervisora de transporte privado escolar (Silvia Suárez), quien actuó como usuaria experta del sistema.

Estas reuniones se llevaron a cabo al finalizar cada hito de desarrollo y permitieron validar si las funcionalidades entregadas cumplían con los requerimientos previamente levantados. La retroalimentación fue utilizada para realizar ajustes menores en los formularios, flujos de navegación y validaciones.

Cada una de estas reuniones fue registrada en un acta firmada y archivada como evidencia del proceso de evaluación funcional.

4.5.7.1. ACTA DE REUNIÓN LEVANTAMIENTO DE REQUISITOS

Fecha: 20/04/2025

4.5.7.1.1. PARTICIPANTES:

- Mateo Cevallos – Desarrollador del sistema
- Silvia Suárez – Supervisora de transporte privado

4.5.7.1.2. OBJETIVO DE LA REUNIÓN

- Validar el listado de requerimientos funcionales y no funcionales, y el caso de uso general previo al desarrollo del sistema.

4.5.7.1.3. TEMAS TRATADOS

- Levantamiento de requerimientos funcionales definición de RF01 al RF11 con el cliente
- Identificación de restricciones del sistema, se acordó no implementar GPS por cuestión de costos

4.5.7.1.4. ACUERDOS Y COMPROMISOS

- Cliente revisará y devolverá retroalimentación de los módulos del sistema
- El desarrollador se compromete a hacer cambios en base a la retroalimentación del cliente.

4.5.7.1.5. PUNTOS PENDIENTES

- Validar interfaz inicial del sistema
- Mostrar interfaz y control de usuarios como primer Hito

4.5.7.2. ACTA DE REUNIÓN HITO 1

Fecha: 26/04/2025

Módulo: F1 Control de usuarios

4.5.7.2.1. OBJETIVO DE LA REUNIÓN

- Validar funcionalidades de inicio y cierre de sesión con credenciales válidas y cookie simple.

4.5.7.2.2. TEMAS TRATADOS:

- Inicio de sesión el sistema permite acceder con credenciales válidas
- Cierre de sesión funcionalidad creada y verificada
- Persistencia de sesión, uso de cookie simple implementado
- Credenciales incorrectas, mensaje ajustado correctamente

4.5.7.2.3. ACUERDOS:

- Validar sesiones por rol personal o supervisor

4.5.7.3. ACTA DE REUNIÓN HITO 2

Fecha: 04/05/2025

Módulos: F2 Supervisores y F3 Propietarios

TEMAS TRATADOS:

- CRUD Supervisores, comprobado y funcional
- CRUD Propietarios, vinculado al supervisor correctamente
- Confirmación antes de borrar, Validado con cuadro emergente
- Pendiente: Filtro por parámetro, en desarrollo

ACUERDOS:

- Terminar filtros por parámetro para poder buscar correctamente supervisores y propietarios.

4.5.7.4. ACTA DE REUNIÓN HITO 3

Fecha: 11/05/2025

Módulos: F4 Personal y F5 Vehículos

4.5.7.4.1. TEMAS TRATADOS:

- CRUD Personal vinculado con supervisor
- CRUD Vehículos, relación con propietario y conductor
- Validación de formularios, detalles estéticos pendientes
- Filtro por personal conductor falta ser revisado por el cliente

4.5.7.4.2. ACUERDOS:

- Mejorar aspecto visual del formulario

4.5.7.5. ACTA DE REUNIÓN HITO 4

Fecha: 18/05/2025

Módulos: F6 Rutas y F7 Horarios

4.5.7.5.1. TEMAS TRATADOS:

- Registro de rutas, enlazadas con vehículos
- Registro de horarios, validación hora inicio < hora fin
- Visualización de vehículo, mostrar nombre en rutas
- Formato de hora Ajustes pendientes

4.5.7.5.2. ACUERDOS:

- Corregir formato de hora para compatibilidad

4.5.7.6. ACTA DE REUNIÓN HITO 5

Fecha: 01/06/2025

Módulos: F8 Pasajeros, F9 Recorridos y F10 Incidentes

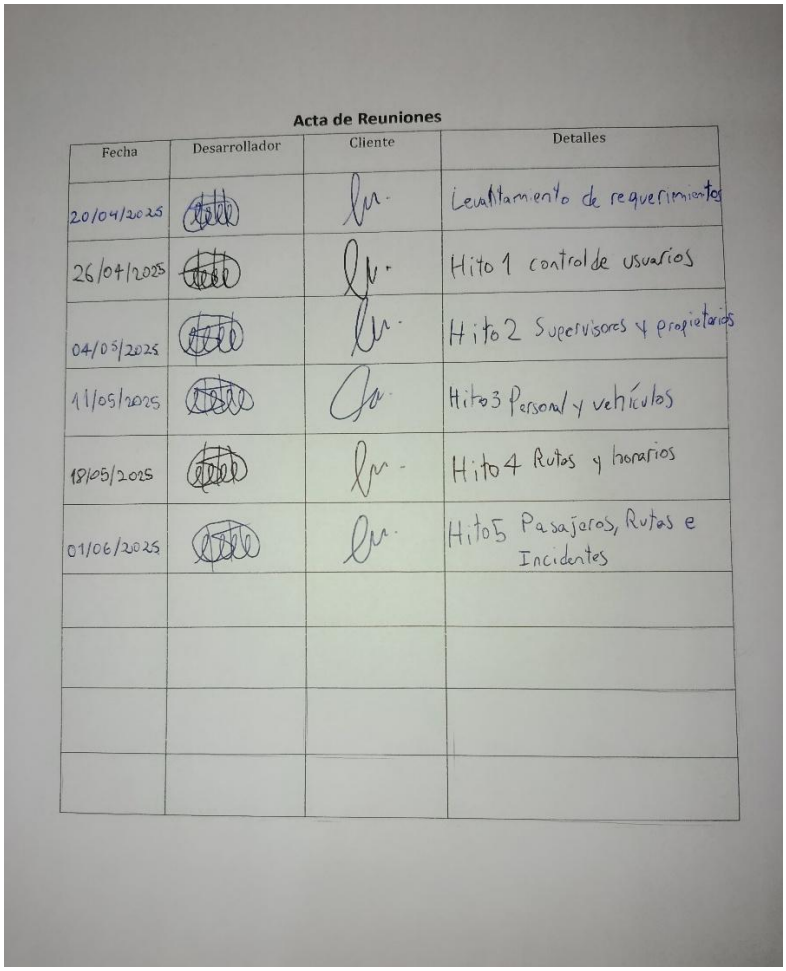
4.5.7.6.1. TEMAS TRATADOS:

- CRUD Pasajeros – Mejorar campos de ingreso – Mateo Cevallos
- Inicio y fin de recorridos – Flujo probado – Mateo Cevallos
- Registro de incidentes – Asociación con recorrido operativa – Mateo Cevallos
- Consulta por parámetros funcional y lista para pruebas

4.5.7.6.2. ACUERDOS:

- Implementar que un pasajero pueda tener varias rutas y una ruta varios pasajeros

Figura 104.
Acta de Reuniones

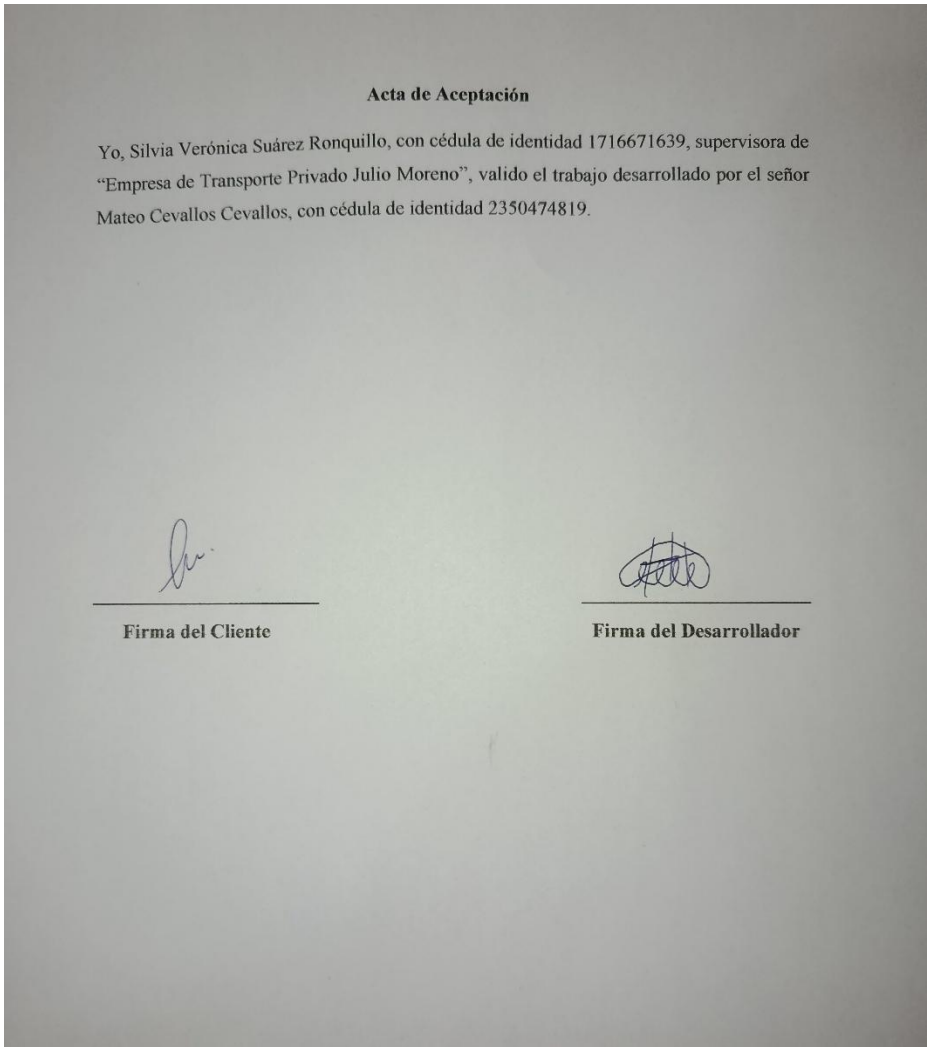


Acta de Reuniones

Fecha	Desarrollador	Cliente	Detalles
20/04/2025			Levantamiento de requerimientos
26/04/2025			Hito 1 control de usuarios
04/05/2025			Hito 2 Supervisores y propietarios
11/05/2025			Hito 3 Personal y vehículos
18/05/2025			Hito 4 Rutas y horarios
01/06/2025			Hito 5 Pasajeros, Rutas e Incidentes

4.5.8. Acta de aceptación del prototipo.

Figura 105.
Acta de Aceptación



Acta de Aceptación

Yo, Silvia Verónica Suárez Ronquillo, con cédula de identidad 1716671639, supervisora de "Empresa de Transporte Privado Julio Moreno", valido el trabajo desarrollado por el señor Mateo Cevallos Cevallos, con cédula de identidad 2350474819.

Firma del Cliente

Firma del Desarrollador

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

1. La implementación de una aplicación web para la gestión del servicio de transporte privado puede contribuir significativamente a la optimización de procesos administrativos y operativos, lo que puede resultar en una la eficiencia, seguridad y calidad del servicio ofrecido.
2. El desarrollo mediante la metodología Crystal Clear permitió obtener retroalimentación constante del usuario experto y facilitó ajustes rápidos a través de hitos, lo que garantizó que la aplicación respondiera efectivamente a las necesidades reales del contexto estudiado.
3. La integración y validación continua con el usuario final en cada etapa del desarrollo permitió identificar necesidades adicionales no previstas inicialmente, destacando la importancia de mantener un enfoque flexible y adaptativo durante el ciclo de vida del proyecto.
4. Los talleres de reflexión ayudan a mejorar continuamente los resultados obtenidos tras la culminación de cada hito, además permite identificar oportunamente aspectos que necesitan ajustes y a asegurar que los aprendizajes obtenidos sean aplicados en etapas posteriores o en futuros proyectos.
5. Es importante identificar los riesgos que pueden surgir durante el desarrollo para poder anticiparse al desastre y mitigarlos a tiempo, de esta forma se puede evitar problemas graves que pongan en riesgo la finalización del proyecto.

5.2. RECOMENDACIONES

1. Se recomienda continuar aplicando una metodología de desarrollo ágil como Crystal Clear, ya que ha demostrado efectividad al facilitar la retroalimentación continua, garantizando así una rápida adaptación a los cambios y mejoras solicitadas por los usuarios.
2. Es esencial generar todos los entregables contemplados por la metodología empleada, enfatizando la importancia de asegurar una gestión eficiente durante el desarrollo y mantenimiento futuro del software.

3. Se sugiere implementar un sistema que permita la gestión dinámica de parámetros configurables, lo que brindará a los usuarios finales mayor autonomía para realizar ajustes en la aplicación sin necesidad constante de intervención técnica externa, aumentando así la eficiencia operativa.
4. Se recomienda mantener un plan estructurado y continuo de pruebas funcionales con participación del usuario experto, realizando pruebas progresivas en cada hito del desarrollo, lo cual contribuirá a asegurar la calidad y cumplimiento efectivo de los requerimientos establecidos.
5. Escoger la metodología adecuada para el proyecto depende del contexto de este; el tipo de negocio definirá si es que los requisitos pueden cambiar o no. Por eso, es clave analizar bien los criterios antes de tomar una decisión. Una comparativa de metodologías puede contribuir con una visión más clara de estas.
6. Se recomienda tomar nota de las sugerencias del usuario, la memoria es frágil y podemos perder retroalimentación valiosa. Por esa razón se debe tener un registro de cada reunión, para adaptar el sistema correctamente.

BIBLIOGRAFÍA

Atlassian. (2025, abril 17). *Kanban: ¿Qué es?* https://www.atlassian.com/es/agile/kanban

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. https://agilemanifesto.org/

Calatayud, A., Katz, R., & Riobó, A. (2021). *Impulsando la transformación digital del transporte en América Latina y el Caribe*. Banco Interamericano de Desarrollo. https://publications.iadb.org/es/impulsando-la-transformacion-digital-del-transporte-en-america-latina-y-el-caribe

Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley Professional.

Cornejo Nava, K. A. (2023). *Sistema de gestión de transporte para incrementar la productividad en el proceso de importación en empresa de transporte* [Universidad San Ignacio de Loyola]. https://repositorio.usil.edu.pe/entities/publication/2604df6e-ba20-4955-aaf9-d86eb7909ba3

Estadística Educativa – Volumen 4. (2023). Ministerio de Educación del Ecuador. https://educacion.gob.ec/wp-content/uploads/downloads/2023/11/Estadistica-Educativa_Volumen-4.pdf

Gallego Rendón, C. M. (2016). *Sistema Inteligente de Control y Gestión de Transporte Privado – SINCONGESTRAP* [Universidad Internacional de La Rioja]. https://reunir.unir.net/bitstream/handle/123456789/4602/GALLEGO%20RENDON,%20CLAUDIA%20MARIA.pdf

Gobierno Autónomo Descentralizado Municipal del cantón Santo Domingo. (2024). *Plan de Desarrollo y Ordenamiento Territorial. Ciudad Región 2050. Versión final*. Gobierno Autónomo Descentralizado Municipal del cantón Santo Domingo. https://www.santodomingo.gob.ec/?page_id=25032

Julio Moreno Espinosa ya cuenta con plano de regulación urbana. (2025, mayo 17). Municipio de Santo Domingo de los Tsáchilas. https://www.santodomingo.gob.ec/?p=24438

Kniberg, H., & Skarin, M. (2020). *Kanban and Scrum: Making the Most of Both*. https://www.agileleanhouse.com/lib/lib/People/HenrikKniberg/KanbanAndScrumInfoQVersionFINAL.pdf

La Hora. (2025). *En Santo Domingo, aumentan muertes violentas y la inseguridad en el primer mes de 2025*. https://www.lahora.com.ec/santodomingo/En-Santo-Domingo-aumentan-muertes-violentas-y-la-inseguridad-en-el-primer-mes-de-2025-20250204-0013.html

López Gómez, A. F. (2022). El servicio privado de transporte: Una propuesta de reinterpretación. *Revista Derecho del Estado*, *51*(enero–abril), 261–311. https://doi.org/10.18601/01229893.n51.09

Next.js by Vercel – The React Framework. (2024). Vercel. https://nextjs.org/

Normativa de transporte escolar para las instituciones educativas del Sistema Nacional de Educación. (2022). Ministerio de Educación del Ecuador. https://educacion.gob.ec/wp-content/plugins/download-monitor/download.php?force=0&id=19755

PlantUML. (2024). https://plantuml.com/

PostgreSQL: The World's Most Advanced Open Source Relational Database. (2024). PostgreSQL Global Development Group. https://www.postgresql.org/

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9a ed.). McGraw-Hill Education.

React: Documentación oficial. (2025). Meta. https://react.dev/

Rendición de cuentas 2022. (2023). Empresa Pública Municipal de Transporte de Santo Domingo. https://www.epmtsd.gob.ec/docs/Rendicion%20de%20Cuentas/2022/RENDICION%20DE%20CUENTAS%20PRESENTACION%202022%20FINAL.pdf

Resolución No. EPMT-SD-GG-HFS-RPO-2022-090: Renovación del permiso de operación de la Compañía de Transporte Escolar e Institucional Bonanza S.A. (2022). Empresa Pública Municipal de Transporte de Santo Domingo. https://www.epmtsd.gob.ec/resoluciones/views/modules/resoluciones/2022/RESOLUCIÓN%20No.%20EPMT-GG-HFS-RPO-2022-090.pdf

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum*. Scrum.org. https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Sommerville, I. (2016). *Software Engineering* (10a ed.). Pearson Education Limited.

Visual Studio Code documentation. (2024). Microsoft. https://code.visualstudio.com/docs

What is a database? (2024). IBM Documentation. https://www.ibm.com/docs/en/i/7.4.0?topic=database

Wikipedia contributors. (2025). *Scrum (software development)*. https://en.wikipedia.org/wiki/Scrum_(software_development)

Yaulema Rojas, E. F., Núñez Silva, L. G., Moposita Lasso, R. M., & Almache Villacís, C. A. (2024). Implementación de un sistema web para la gestión integral del transporte para la Cooperativa Río San Pablo. *Dominio de las Ciencias*, *10*(3), 719–739. https://doi.org/10.23857/dc.v10i3.3950