

UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIEROS EN SOFTWARE

FORMA: PROYECTO TECNOLÓGICO

TEMA:

SISTEMA WEB PARA LA GESTIÓN OPERATIVA Y CONTROL AUTOMATIZADO DEL ARRIBO Y DESPACHO DE UNIDADES MEDIANTE LA GEOLOCALIZACIÓN EN PUNTOS ESTABLECIDOS EN LA COMPAÑÍA DE TRANSPORTE ANTOÑO POLO S.A.

AUTORES:

RENAN MARCIAL SIGCHA PUNINA
STEVEN ALEXANDER AGUALONGO MOROCHO

DIRECTOR:

ING. DARWIN CARRIÓN

GUARANDA – ECUADOR

2026

TEMA DEL PROYECTO TECNOLÓGICO

SISTEMA WEB PARA LA GESTIÓN OPERATIVA Y CONTROL AUTOMATIZADO DEL ARRIBO Y DESPACHO DE UNIDADES MEDIANTE LA GEOLOCALIZACIÓN EN PUNTOS ESTABLECIDOS EN LA COMPAÑÍA DE TRANSPORTE ANTOÑO POLO S.A.

DEDICATORIA

Con amor y cariño quiero dedicar este trabajo de tesis a Dios por darme Salud, vida y brindar sabiduría en el transcurso de mi etapa académica.

También a mis padres, por siempre confiar desde el primer momento que ingresé a la Universidad, por ser un ejemplo de resiliencia, responsabilidad y esfuerzo. Inculcando siempre en valores dentro y fuera de casa, estos resultados de finalización de mis estudios son por ustedes, nunca han dejado de apoyarme y enseñar que las cosas se hacen con paciencia.

A mi hermana, que es y será una guía en mis metas y las decisiones que he tomado en mi vida, dentro de mi proceso de estudio ha sido un pilar relevante aportando positivamente a mis conocimientos, motivando moralmente en momentos que estaba a punto de tirar la toalla.

A mi gran amiga, Verito Poaquiza, por su confianza incondicional y por enseñarme que nada es imposible. Tu motivación y tus palabras de aliento fueron el pilar que necesité para mantener la fortaleza moral, dándome la certeza de que rendirse nunca fue una opción.

Y a mis amigos cercanos que me acompañaron en este proceso académico, por guiarme, compartir nuevas experiencias, habilidades y aportar con muchas alegrías cuando me encontraba bajoneado. Esto permitió ser más sociable reír más, amar más y estar satisfecho que las cosas suceden por algo y no pasa por alto mi vida académica, donde las motivaciones y abrazos fueron recibidos por ustedes y en consecuencia han aportado positivamente en mi vida

Renan Marcial Sigcha Punina

Dedico este trabajo de titulación primeramente a Dios, por darme fortalezas, iluminar mi camino, brindarme salud y permitirme llegar a la culminación de esta etapa importante de mi vida profesional.

A mis padres, que han sido pilares fundamentales en el transcurso de mi vida. Gracias por cada sacrificio, por los valores y enseñanzas que me inculcaron, por la confianza depositada en mí y por el apoyo inmenso que recibí en cada paso que daba. Su esfuerzo que han empleado me ha enseñado a no rendirme jamás. Este logro también les pertenece a ustedes.

A mis hermanos, por su cariño, sus consejos y compañía. Gracias por formar parte de mi vida y por motivarme a crecer como persona y como profesional.

A mi amada pareja, Sirey Toapanta, por llegar a mi vida en el momento más indicado y convertirte en la mayor bendición que Dios ha puesto en mi camino. Gracias por tu amor, comprensión, paciencia infinita y por creer en mí cuando los obstáculos se ponían difíciles. Tus palabras de ánimo me ayudaron a enfrentar cualquier circunstancia de la mejor manera y a descubrir una versión mejor de mí mismo. Gracias por inspirarme cada día, por darme motivos para cumplir mis sueños y mantener siempre la esperanza. Este logro también es tuyo y tiene una parte de ti, porque eres una inspiración que nunca dejará de crecer. Gracias por ser parte de este sueño y ayudarme a hacerlo realidad, mi quiteña bonita.

A mis lindos gatitos, Oracio, Osiris y Ruffina, cuya presencia hicieron que pudiera soportar largas horas de estudio, regalándome momentos de tranquilidad, paz y alegría que jamás olvidaré. Su compañía hizo que este proceso se sienta más ligero; siempre ocuparán un lugar especial en mi corazón y en mis logros.

Finalmente, dedico este éxito a todos mis amigos y a las personas que, con su amistad y buenos deseos, formaron parte de este capítulo que estoy por completar.

Gracias de todo corazón a todos.

Steven Alexander Agualongo Morocho.

AGRADECIMIENTOS

Al finalizar esta etapa, en primer lugar, agradecemos a Dios por ser el sustento e iluminación para culminar esta carrera y lograr un escalón más. De igual manera expresamos un total agradecimiento a nuestras familias, por los buenos ejemplos, las motivaciones y por ser la base primordial para la finalización de este proyecto.

También expresamos nuestros agradecimientos a la Universidad Estatal de Bolívar por la formación académica recibida durante el transcurso de estos años, base sobre el cual se sustenta el presente trabajo.

Al Ing. Darwin Carrión, director de tesis, le agradecemos por brindar la orientación constante y el criterio técnico con que acompañó cada etapa del desarrollo. Su guía permitió encaminar las decisiones de arquitectura y sostener el rigor metodológico del proyecto.

A los pares académicos, Ingenieros Danilo Barreno y Galuth García, agradecemos las observaciones y revisiones realizadas; sus correcciones ayudaron a depurar el diseño y a mejorar la calidad del resultado de nuestra tesis.

A la Compañía de Transporte Antoño Polo S.A., y de manera particular a su Gerente General, Sr. Leonardo Guamán, expresamos nuestro profundo agradecimiento por facilitar la información de las operaciones, los procesos que realizan internamente y permitir realizar el desarrollo del sistema. Sin esto no hubiera sido factible llevar a cabo nuestro proyecto tecnológico. Gracias a todos.

Renan Sigcha y Alexander Agualongo

CERTIFICADO DE VALIDACIÓN

UEB
UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CERTIFICADO DE VALIDACIÓN

Ing. Darwin Carrión, Ing. Galuth García e Ing. Danilo Barreno, en su orden Director y Pares Académicos del Trabajo de Integración Curricular "SISTEMA WEB PARA LA GESTIÓN OPERATIVA Y CONTROL AUTOMATIZADO DEL ARRIBO Y DESPACHO DE UNIDADES MEDIANTE LA GEOLOCALIZACIÓN EN PUNTOS ESTABLECIDOS EN LA COMPAÑÍA DE TRANSPORTE MIXTO ANTONIO POLO S.A." desarrollado por los señores Renan Marcial Sigcha Punina y Steven Alexander Agualongo Morocho.

CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 02 de Julio del 2026



Ing. Darwin Carrión
Director



Ing. Galuth García
Par Académico



Ing. Danilo Barreno
Par Académico

DERECHOS DE AUTOR

Nosotros **Agualongo Morocho Steven Alexander y Sigcha Punina Renan Marcial** portadores de la Cédula de Identidad No **0105924740 y 0250244530** en calidad de autores y titulares de los derechos morales y patrimoniales del Trabajo de Titulación**: Sistema web para la gestión operativa y control automatizado del arribo y despacho de unidades mediante Geolocalización en puntos establecidos en la compañía de transporte Antoño Polo S.A.**, modalidad **Proyecto Tecnológico**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi/nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo/autorizamos a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Los autores declaran que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.



Steven Alexander Agualongo Morocho
C.I: 0105924740



Renan Marcial Sigcha Punina
C.I: 0250244530

INDICE DE CONTENIDOS

TEMA DEL PROYECTO TECNOLÓGICO i
DEDICATORIA ii
AGRADECIMIENTOS iv
CERTIFICADO DE VALIDACIÓN v
INDICE DE CONTENIDOS vii
INDICE DE TABLAS x
ÍNDICE DE FIGURAS xii
INTRODUCCIÓN 1
RESUMEN 2
ABSTRACT 3
KICHWA 4
CAPÍTULO I 5
FORMULACIÓN GENERAL DEL PROYECTO 5
1.1. Tema 5
1.2 Descripción del Problema 5
1.3 Justificación 6
1.4 Objetivos: General y Específicos 7
Objetivo General 7
Objetivo Especifico 7
CAPÍTULO II 8
MARCO TEÓRICO 8
2.1 Antecedentes 8
2.2 Científico 11
2.2.1 Metodología del desarrollo de software 11
2.2.2 Metodologías Ágiles 11
2.3 Conceptual 13
2.4 Legal 19
2.4.1 Ley de Propiedad Intelectual 19
2.4.2 Ley Orgánica de Protección de Datos Personales 20
CÓDIGO INTERNO COMPAÑÍA DE TRANSPORTE ANTOÑO POLO S.A. 24
CAPÍTULO III 27

METODOLOGÍA 27
3.1 Metodología de Desarrollo de Software 27
Valores y principios de las metodologías 28
3.2 Técnicas e Instrumentos de Recopilación de Datos 32
Entrevista 32
Observación 33
CAPÍTULO IV 34
INGENIERÍA DEL PROYECTO 34
4.1 Planificación 34
4.1.1 Análisis de instrumentos 34
4.1.2 Análisis de observación 39
4.1.3 Historias de usuario 42
4.1.4 Gestión de trabajo mediante Kanban 55
4.2 Diseño del sistema 58
4.2.1 Vista Arquitectónica 58
4.2.2 Vistas arquitectónicas 4+1 60
4.2.2.1 Vista de escenarios 60
4.2.2.2 Vista lógica del sistema 61
4.2.2.3 Vistas de procesos 62
4.2.2.4 Vista de desarrollo 63
4.2.2.5 Vista física 64
4.2.3 Modelo arquitectónico 65
4.2.4 Modelo 66
4.2.5 Vista 67
4.2.6 Controlador 70
4.3 Programación 72
4.3.1 Patrones de Diseño GoF 73
4.4 Pruebas 84
4.4.1 Pruebas unitarias 85
4.4.2 Pruebas de funcionalidad 88
4.2.3 Pruebas de rendimiento 92
4.2.4 Pruebas de seguridad 93
4.2.5 Pruebas de integración 93

CONCLUSIONES 96
RECOMENDACIONES 97
BIBLIOGRAFÍA 98

INDICE DE TABLAS

Tabla 1: Lenguajes de programación 14
Tabla 2: Ubicación Geográfica. 26
Tabla 3: Análisis de la observación. 39
Tabla 4: Restricciones del sistema y su trazabilidad 41
Tabla 5: Historia de Gestión de usuarios del sistema. 42
Tabla 6: Historia de Autenticación al sistema. 43
Tabla 7: Historia de Gestión de unidades. 44
Tabla 8: Historia de Configuración parámetro del sistema 44
Tabla 9: Historia de Gestión de grupos de descansos 45
Tabla 10: Historia de Gestión del sorteo diario de turnos 46
Tabla 11: Historia de Confirmación de presencia con geolocalización. 47
Tabla 12: Historia de Control de cola de conductores. 48
Tabla 13: Historia de Registro de Salida del Conductor. 49
Tabla 14: Historia de Registro de llegada y multa por retraso. 50
Tabla 15: Historia de Gestión de justificativos. 51
Tabla 16: Historia de Monitoreo de GPS. 52
Tabla 17: Historia de Control documental de conductores y unidades. 52
Tabla 18: Historia de Gestión de ciclos anuales de operación. 53
Tabla 19: Historia de Reportes operativos del sistema. 54
Tabla 20: Roles de acceso al sistema 55
Tabla 21: Descripción de las vistas arquitectónicas 4+1 de kruchten 58
Tabla 22: Algoritmo de Gestión de Unidades 70
Tabla 23: Algoritmo de Confirmación de Presencia con Geocerca. 70
Tabla 24: Algoritmo de Confirmación de Bloqueo de Turno 71
Tabla 25: Herramientas de pruebas 84
Tabla 26: Usuarios finales 84
Tabla 27: Tabla de pruebas unitarias. 87
Tabla 28: Caso de pruebas de Modulo Administración del sistema 88
Tabla 29: Caso de prueba de Modulo planificación de turnos 89
Tabla 30: Caso de prueba de Modulo Operación diaria 89
Tabla 31: Caso de prueba de modulo Multa y sanciones 90
Tabla 32: Caso de prueba de Modulo de monitoreo y documentación 91
Tabla 33: Tabla de prueba de rendimiento 92

Tabla 34: Tabla de Pruebas de seguridad..... 93
Tabla 35: Prueba de integración flujo de multas..... 94
Tabla 36: Prueba de integración flujo de notificaciones. 94
Tabla 37: Prueba de integración flujo de historial de cambios. 95
Tabla 38: Tabla de presupuesto del proyecto..... 104

ÍNDICE DE FIGURAS

Figura 1: Tablero visual de Kanban del proyecto 56
Figura 2: Tablero visual Kanban proceso planificación. 56
Figura 3: Tablero visual Kanban proceso diseño 56
Figura 4: Tablero visual Kanban proceso de programación 57
Figura 5: Tablero final de pruebas 57
Figura 6: Diagrama de casos de uso (escenarios) 60
Figura 7: Diagrama de clases (Vista logica) 61
Figura 8: Diagrama de actividad, Sorteo diario y generación de turnos 62
Figura 9: Diagrama de actividad, control automatizado por Geocercas 63
Figura 10: Vista de desarrollo (diagrama de paquetes) 64
Figura 11: Diagrama de despliegue 65
Figura 12: Modelo-Vista-Controlador 65
Figura 13: Diagrama de Entidad-Relación 66
Figura 14: Vista de la Página de Inicio. 67
Figura 15: Vista de Inicio de Sesión 67
Figura 16: Vista de Conductor 68
Figura 17: Vista de Gerente 68
Figura 18: Vista de Justificativo 69
Figura 19: Vista de Registro - Salida 69
Figura 20: Cronograma de Gantt 102

INTRODUCCIÓN

Hoy en día, las herramientas informáticas han radicalizado el manejo de los procesos en las empresas. En el caso del transporte terrestre, contar con un sistema que ayude a organizar las operaciones diarias ya no es una opción, sino una necesidad, pues permite llevar un mejor control de las unidades, coordinar los turnos de los conductores y tomar decisiones con información más clara y ordenada.

La Compañía de Transporte Antoño Polo S.A. realiza procesos críticos como controlar la llegada y salida de sus unidades (camionetas), asignar turnos y aplicar las normas internas de la empresa. El problema radica en que la mayoría de estos procesos se ejecutan de manera manual. Esto dificulta generar un seguimiento preciso de los datos y asegurarse de que los tiempos de viaje se cumplan. Como resultado, se generan retrasos, errores y una operación que no termina de ser eficiente.

Ante esta problemática, el presente proyecto de titulación plantea desarrollar un sistema web que automatice la gestión operativa de la compañía, usando geolocalización para controlar el arribo y despacho de unidades en los puntos definidos.

Para hacerlo posible, se aplicaron conocimientos de ingeniería de software, desarrollo web, geolocalización (Google Maps) y bases de datos (MySQL). Todo esto aplicado a un problema real que se identificó dentro de la empresa. Adicional a esto se utilizaron herramientas como Laravel (Backend) y Tailwind CSS, Alpine.js y el motor de plantillas Blade para el frontend. Al final, lo que se busca es ofrecer una solución tecnológica práctica que ayude a que la operación sea más eficiente y que la información esté siempre disponible y bien organizada.

El documento está dividido en cuatro fases: El primero explica el problema, por qué es importante resolverlo y qué objetivos se persiguen. El segundo reúne los conceptos teóricos que sirven de base para el desarrollo. El tercero describe cómo se llevó a cabo el proyecto, aquí se utilizó una combinación de dos metodologías ágiles (XP + Kanban), esto ayuda a obtener mejor flexibilidad en el desarrollo y obtener el sistema funcional mucho más rápido. El cuarto, por último, presenta la ejecución del proyecto en cuatro fases planificación, diseño, codificación y pruebas.

RESUMEN

La Compañía de Transporte Antoño Polo S.A. operaba bajo procesos obsoletos para la actualidad. La asignación de turnos se realizaba mediante sorteo manuales, el registro de salidas y llegadas se anotaban en cuadernos y la comunicación entre las paradas de Guaranda y Salinas dependía del WhatsApp. Esta falta de sistematización generaba pérdida de información, inconsistencia en los datos y constantes conflictos logísticos entre los conductores y la administración.

Ante esta problemática se planteó el desarrollo de un sistema web orientado a automatizar la gestión operativa, abarcando desde la asignación de turnos hasta el control de arribo y despacho, integrando tecnología de geolocalización (GPS) para el monitoreo de cada unidad. Para construirlo se trabajó con Laravel en el backend, MySQL para los datos y Tailwind CSS + Alpine.js para la interfaz de usuario. La parte del GPS se implementó como una PWA y los cálculos de distancia se procesan en el servidor, garantizando la integridad de la información y previniendo la manipulación de coordenadas desde el celular.

El proyecto se gestionó bajo un enfoque metodológico (XP y Kanban), estableciendo el tiempo de entrega cada dos semanas. Al culminar el desarrollo del sistema se hicieron pruebas de varios tipos: unitarias, funcionales, de integración, de rendimiento con K6 y de seguridad con OWASP ZAP. Los resultados evidenciaron un rendimiento óptimo en módulos críticos, como el sorteo algorítmico y el monitoreo, sin detectar vulnerabilidades que comprometan el funcionamiento del sistema

En la actualidad, la implementación del sistema web logró sustituir los procesos manuales, centralizando la información, automatizando el flujo de trabajo y dotando a la compañía de una herramienta precisa para la toma de decisiones basada en datos verificables.

Palabras clave: sistema web, geolocalización, gestión de transporte y automatización

ABSTRACT

Compañía de Transporte Antoño Polo S.A. used to run its operations without anything close to a proper system. Shifts were drawn on slips of paper, departures were written in notebooks, and the only way the Guaranda and Salinas terminals communicated was through WhatsApp. Anyone who has worked in transportation knows this leads to complaints, lost data, and drivers arguing over their turn. That's where the idea of building something better came from.

The goal was to create a web system that would automate all of that, from shift assignment to arrival and departure control, using the driver's cell phone GPS to know where each bus was in real time. Laravel was used on the backend, MySQL for the database, and Tailwind CSS with Alpine.js for the interface. The GPS part was implemented as a PWA, and all distance calculations run on the server side so that nobody could tamper with the location data from their phone.

The project followed an agile methodology, with deliveries scheduled throughout the process and testing done as progress was made. Once finished, several types of tests were run: unit tests, functional tests, integration tests, performance tests with K6, and security tests with OWASP ZAP. Everything worked well. The main modules, like the shift draw and geofence monitoring, ran without errors, and no serious security issues were found.

Today, the system has replaced the notebooks and WhatsApp. Information is centralized, shifts are assigned automatically, and the company can make decisions based on real data instead of relying on what someone remembers or jotted down on a loose sheet of paper.

Keywords: web system, geolocation, transportation management, GPS tracking, automation

KICHWA

Antoño Polo S.A. aparina Tandanaku, kunan pachapaka, ñami mawkayashka rurana ñankunawan llamkakurka rina mitakunataka uchilla papilkunawanmi akllak karka, llukshinakunataka kwadirnukunapimi killkak karka, Guaranda Salinas terminalkunapuraka WhatsApp-llawanmi rurarik karka. Aparinapi llamkashka maykan runapash yachanmichayka, piñarinakunaman, chinkarishka willaykunaman, paykunapak mitamanta makanakuk chofirkunamanpashmi chayan. Chaymantami ashtawan alli imallatapash rurana yuyay wacharirka.

Chaymantaka, tukuy chayta paylla rurarichun, sistemata wiñachina yuyaymi rikurirka; mitakunata churanamanta callarishpa, antawakunapak chayamunata, llukshinatapash rikunakaman. Selular GPS-wanmi, sapan antawa maypi kashkata kunan pachallapi yachana karka. Kayta rurankapakka: llamkarka, willaykunapakka MySQL, ñawi pankakunapika Tailwind CSS Alpine.js-wanpash. GPS rurayka shuk PWA shinami churarirka; karu tupuykunaka servidor ladopimi rurarin, pipash selularmanta maypi kashkata ama tikrachi tukuchunllami.

Kay llamkayka, ágil nishka utka rurana ñanwanmi apashka karka; mashna pachapi rurashkata rikuchishpa, ñawpakman rishpa kamashpapash. Sistema wiñachiyta tukuchishpaka, achka laya kamaykunatami rurarka: unitaria, funcional, integración, K6-wan rendimiento kamay, OWASP ZAP-wan amachay kamaypash. Tukuymi allita llamkarka; mitata akllana, geocerca-wan rikuna shinallak jatun ruraykunaka mana pantarirkachu, amachaypipash mana jatun llakikuna tuparirkachu.

Kunanka, kay sistema kwadirnukunapa, WhatsApp-pa kuskatapash ña japirka. Tukuy willaykunaka shuk kuskallapimi tantarishka, mitakunaka paykunalla churarin, llamkana wasipash kikin willaykunapi shayarishpami akllaykunata rurai tukun; mana pipash yuyarishkapi, mana shuk sapalla pankapi killkashkapi shayarishpa.

Mutsurik shimikuna: web (llika) sistema, maypi kashka riksina, aparinata pushana, GPS katina, paylla rurarina.

CAPÍTULO I

FORMULACIÓN GENERAL DEL PROYECTO

1.1. Tema

Sistema web para la gestión operativa y control automatizado del arribo y despacho de unidades mediante Geolocalización en puntos establecidos en la compañía de transporte Antoño Polo S.A.

1.2 Descripción del Problema

La compañía de transporte Antoño Polo S.A domiciliada en la parroquia de Salinas realiza sus actividades operativas en dos puntos establecidos: en Salinas y Guaranda, donde realizan el proceso de arribo y despacho de sus unidades es gestionado de manera manual por una tarjetera que se encarga de coordinar el orden de la llegada y salida. El registro de los conductores se lleva en hojas de cálculo, mientras que la asignación de turnos se realiza de forma manual en las paradas generando una desconexión entre la información administrativa y operativa. Esta situación afecta la eficiencia del servicio y el cumplimiento de las normativas internas de la compañía, entre las principales se destacan:

- **Incumplimiento de las reglas de negocio:** El reglamento interno establece un intervalo de 15 minutos para la salida de cada unidad y un tiempo máximo de una hora para el trayecto entre Salinas y Guaranda o viceversa. Sin embargo, al no haber un control bien preciso, provoca errores en el orden de salida y llegada de las unidades, molestias y confusiones de los conductores.
- **Control ineficiente de sanciones:** Las multas por incumplimiento de horarios o arribo tardío se registran de manera manual, lo que dificulta su seguimiento, control y aplicación adecuada, con tendencias a perdida o destrucción de información.
- **Falta de validación de información en tiempo real:** La persona encargada de la asignación de turnos no dispone de acceso inmediato a información, como el estado de pagos de multas por parte de los socios (conductores) o la vigencia de documentos, lo que puede permitir la operación de unidades que no cumplen con los requisitos establecidos.

La integración de información administrativa y operativa depende de procesos manuales y el uso de registros físicos, generando una brecha critica de la integridad de los datos disminuyendo la precisión de salida y llega de las unidades a las paradas, baja transparencia en la aplicación de multas y limitaciones en la confiabilidad de la información para la toma de decisiones.

1.3 Justificación

La automatización de un sistema web en el sector del transporte ha transformado la gestión operativa, donde el uso de métodos tradicionales y manuales en el control de las unidades (camionetas, buses entre otros) resulta cada vez menos efectiva frente a las exigencias actuales que demandan una precisión para optimizar recursos.

La implementación de un sistema web con integración de tecnología GPS genera consecuencias positivas sobre la gestión de flota vehicular, el uso de estas herramientas optimiza de manera explícita un control constante en las operaciones dentro de una organización. Esto permite a la administración poseer herramientas precisas para el seguimiento de sus unidades en tiempo real. Como resultado es indispensable para anticipar desviaciones y obtener una toma de decisiones eficaz (García & Saavedra, 2026)

El uso de plataformas web integrales contribuye a reducir significativamente los errores derivados de procesos manuales, mejorando la trazabilidad y la supervisión en las operaciones de transporte. La pertinencia de este tipo de soluciones se sustenta en estudios y experiencias previas relacionadas con la implementación de tecnologías web y sistemas de geolocalización en el sector del transporte (Palacios, 2025)

En el caso de la compañía Antoño Polo S.A el desarrollo de este sistema contribuirá a mejorar la organización de los turnos, optimizar el control de los tiempos de recorrido y disponer de información confiable para la toma de decisiones, así mismo el sistema facilitará la aplicación de multas de manera transparente al calcular automáticamente las sanciones basándose en las coordenadas de GPS, penalizando a aquellas unidades que incumplan el intervalo de salida o excedan el tiempo máximo de llegada. Toda esta operatividad se gestionará proveyendo al personal administrativo y a la tarjetera de una

interfaz centralizada, controlada mediante roles y usuarios, lo que permitirá administrar el orden establecido con total precisión, también permitirá controlar previamente la validez de los documentos de cada unidad licencia y matricula.

Los beneficiarios directos serán el personal administrativo y la tarjetera quienes contarán con una herramienta tecnológica para gestionar los procesos de manera más eficiente. Asimismo, los conductores se beneficiarán al contar con un sistema más justo, ordenado y transparente en el control de su operatividad.

El enfoque de este proyecto es transformar la gestión de una empresa de transporte local tradicional en un modelo operativo digitalmente avanzado, adaptando las mejores prácticas tecnológicas al contexto específico de Salinas y Guaranda. El alcance y perspectiva del producto comprende un Sistema web dinámico con módulos de gestión de turnos y visualización de geolocalización en mapas interactivos.

El proyecto se encuentra alineado al dominio de las Tecnologías de la Información y Comunicación, dentro de la línea de investigación de Ingeniería de Software, Redes y Telecomunicaciones, específicamente en la sublínea de diseño e implementación de sistemas de información de la carrera de Software de la Universidad Estatal de Bolívar.

1.4 Objetivos: General y Específicos

Objetivo General

Desarrollar un sistema web para la gestión operativa y control automatizado del arribo y despacho de unidades mediante geolocalización en puntos establecidos en la Compañía de Transporte Antoño Polo S.A.

Objetivo Especifico

- Analizar la situación actual de la gestión operativa de la compañía para identificar sus requerimientos.
- Diseñar una arquitectura de software robusta que permita un sistema adecuado y bajo las normas más alta de calidad.
- Implementar el sistema web mediante un lenguaje de programación y bases de datos adecuados.
- Realizar pruebas del sistema que garanticen el correcto funcionamiento.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

Damiani (2021) presentó la investigación titulada “Sistema de gestión operativa y automatización de procesos utilizando RPA en una compañía de servicio de monitoreo vehicular GPS”. El propósito fue resolver una problemática operativa en una empresa mayorista mediante el diseño y desarrollo de un producto de software enfocado en el monitoreo satelital y la creación de aplicaciones personalizadas. Para la estructuración del sistema se aplicó la metodología ágil Scrum, lo que permitió modular la arquitectura del software a través de historias de usuarios, definiendo roles claves como el Product Owner y el equipo de desarrollo para iterar entregables de valor. Para la recolección y análisis de datos, el estudio empleó el modelado de procesos de negocios evaluando el escenario de antes y después de la implementación, validando la efectividad del sistema mediante indicadores de éxito operativos. Las herramientas de desarrollo del software se programaron utilizando el lenguaje C# bajo el entorno .NET, e integró tecnologías de automatización robótica de procesos (RPA).

Reinoso (2024) desarrolló la investigación titulada "Aplicación web y móvil para el monitoreo de la operatividad de las unidades de transporte de taxi para la empresa de servicio de radio taxi del cantón Ibarra, haciendo uso de geoposicionamiento", cuyo objetivo fue implementar un sistema para el monitoreo en tiempo real de la operatividad de las unidades vehiculares. Para la construcción de este aplicativo, el autor empleó metodologías ágiles de desarrollo, utilizando herramientas como satélites GPS, un servidor API Rest, bases de datos en la nube y la integración con la tecnología de Google Maps para el trazado de geocercas. El resultado principal demostró que la implementación de un sistema ágil y sistemático optimiza la eficacia del control vehicular, superando los sistemas de rastreo tradicionales y permitiendo al administrador monitorear continuamente a las unidades, lo que reduce la inseguridad y agiliza el servicio de despacho.

Yuqui & Motoche (2024) presentaron el estudio titulado "Desarrollo de un sistema web para la automatización de procesos de transporte pesado en la ciudad de Quito". El

propósito de esta investigación fue crear un aplicativo que automatice los procesos operativos más importantes del sector, tales como la planificación de rutas, la gestión de cargas y la asignación de flotas. Para ello, se aplicó un enfoque metodológico cuantitativo, de nivel descriptivo y diseño no experimental transversal, empleando simultáneamente la metodología ágil de Programación Extrema (XP) para el ciclo de desarrollo del producto. La recolección de datos se realizó mediante encuestas aplicadas a una muestra de 67 trabajadores del área de transporte. Las herramientas de desarrollo del software se estructuraron dividiendo claramente la arquitectura: se utilizó el framework Laravel (PHP) para la lógica de negocio en el backend, Next.js (React) para la interfaz de usuario en el frontend, y PostgreSQL como sistema gestor de base de datos relacional. Los resultados demostraron que el sistema implementado logró reducir significativamente las tareas manuales y la carga de trabajo administrativo, permitiendo una asignación eficiente de vehículos basada en la distancia y duración de los viajes.

Cabrera & Briceño (2025) desarrollaron el proyecto técnico titulado “Desarrollo de un programa estadístico para el análisis y gestión de información de geoposición vehicular de la empresa ETAPA EP”. El propósito fue analizar descriptivamente los datos de geoposición generados por la flota vehicular para optimizar el monitoreo y control, reducir procesos manuales y disminuir tiempos improductivos. Para el procesamiento y estructuración del sistema, se empleó una metodología enfocada en la limpieza, análisis descriptivo y visualización geoespacial de datos. La recolección y análisis de información se basó en la importación de bases de datos generadas por los dispositivos de rastreo GPS en formato .CSV correspondientes a recorridos reales. Las herramientas de desarrollo del software de escritorio se programaron íntegramente utilizando el lenguaje Python, empleando bibliotecas como Pandas para la limpieza de datos, Matplotlib para la generación de gráficas, Folium para mapas interactivos y Tkinter para el diseño de la interfaz visual. Los resultados evidenciaron que la plataforma automatizada facilita la identificación de patrones de conducción ineficientes y permite proyectar estrategias de reducción de costos, transformando una gran cantidad de datos geoespaciales en conocimiento práctico para la gestión de flotas.

Aguilar & Reyes (2021) en su estudio titulado “Recolección, análisis y envío de datos de la Unidad de Control de Motor (ECU) a bordo de un vehículo para monitoreo a través

de una aplicación móvil" tuvo como objetivo implementar un sistema móvil para el monitoreo remoto de los vehículos perteneciente a la empresa Arcotel, en su diseño de la metodología y arquitectura tecnológica optaron por reemplazar el uso tradicional de GPS vehiculares externos a utilizar un dispositivo móvil en el vehículo. Para establecer la geolocalización la aplicación implemento un sistema de posicionamiento que capturaba las coordenadas exactas directamente desde el hardware del GPS interno del teléfono celular. El diseño del proyecto se basó en subsistema de análisis y envío, donde el dispositivo móvil actúa como procesador (calculando la distancia con su propio Gps interno) enviando la información a la nube a través de la propia red móvil (plan datos). El código fuente y sus módulos respectivos fueron desarrollados utilizando el lenguaje de programación java, la base de datos utilizado para almacenar fue cloud Firestore, fue elegido por ser escalable y guarda los datos (como latitud, longitud, velocidad y kilometraje) mucho más flexible.

De los antecedentes analizados se evidencia que, la implementación de sistemas web adheridas en tecnologías de geolocalización y monitoreo contribuye de manera positiva a la mejora de la gestión operativa dentro de una organización. El uso de estos sistemas permite automatizar procesos y optimizar los tiempos de operación para mejorar la trazabilidad de la gestión operativa dentro de una organización, facilitando la toma de decisiones administrativas. A nivel metodológico demuestra una clara tendencia hacia la utilización de metodologías ágiles de desarrollo de software, destacando la aplicación de marcos de trabajo como Scrum y Programación Extrema (XP), los cuales permiten modular la arquitectura del sistema mediante historias de usuario e iterar entregables de valor técnico. Asimismo, para el levantamiento de requerimientos y evaluación de resultados, los proyectos de esta naturaleza respaldan en enfoques de investigación cuantitativa, descriptiva y en el modelado de procesos de negocio y se fundamenta en la integración de arquitecturas web, bases de datos relacionales y el consumo de servicios de geoposicionamiento como satélites GPS y APIs de mapas interactivos. En este contexto, la adopción de estas metodologías ágiles junto al uso de la geolocalización aplicada se presenta como la alternativa más eficiente para estructurar el control del arribo y despacho de unidades; una solución que no solo permite corregir las falencias operativas de la compañía, sino también mejorar la calidad del servicio a los usuarios.

2.2 Científico

2.2.1 Metodología del desarrollo de software

Bravo et al. (2022) señalan que los modelos tradicionales resultan poco flexibles o requieren métricas excesivamente complejas para responder a la evolución actual del entorno tecnológico, la implementación de una metodología apropiada permite optimizar la gestión de recursos, integrar la retroalimentación continua y asegurar la competitividad, e incluso la sostenibilidad, de las organizaciones dedicadas al desarrollo de software. Las metodologías de desarrollo de software representan varios enfoques orientados a desarrollar proyectos de manera eficiente dentro de entornos de trabajo dinámicos. Debido a que el proceso de creación de software se encuentra en constante cambio como resultado de las necesidades variables de los clientes y de los requisitos de entrega en plazos reducidos, muchas veces bajo condiciones financieras limitadas, la selección de una metodología adecuada se vuelve un factor crítico.

2.2.2 Metodologías Ágiles

García & Toscano (2025) señalan que las metodologías ágiles son un enfoque moderno de gestión de proyectos orientado a priorizar la adaptabilidad y la entrega continua de valor, resultando especialmente idóneas en proyectos de innovación con mejoras continuas que se desarrollan en entornos dinámicos.

En la práctica, las empresas de desarrollo de software implementan este modelo logrando equiparar sus requisitos sobre la marcha ante cualquier imprevisto, esto permite detectar los riesgos de manera más temprana e impulsar significativamente la motivación y la colaboración del equipo de desarrollo.

A diferencia de los modelos tradicionales que exigen fases secuenciales y requerimientos estrictos, el enfoque ágil se fundamenta en la comunicación constante, la autogestión de los equipos de trabajo, la flexibilidad del alcance y la generación de documentación que sea estrictamente necesaria.

Extreme Programming (XP)

Según Maida y Pacienzia (2015) una metodología ágil se centra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (p. 57).

Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. Podríamos decir que SCRUM se basa en cierto caos controlado, pero establece ciertos mecanismos para controlar esta indeterminación, manipular lo impredecible y controlar la flexibilidad (Maida & Pacienzia, 2015, p. 73).

Kanban

Según Maida & Pacienzia (2015) la metodología Kanban utiliza un mecanismo de control visual para hacer seguimiento del trabajo conforme este viaja a través del flujo de valor. Normalmente, se emplea un panel o pizarra con notas adhesivas o un panel electrónico de tarjetas para gestionar el flujo de trabajo y las asignaciones. Las mejores prácticas apuntan al uso de ambos métodos. El Kanban (o tarjeta visual) implica que se genera una señal visual para indicar que hay nuevos bloques de trabajo que pueden ser comenzados porque el trabajo en curso actual no alcanza el máximo acordado (p. 90).

Lean Development (LD)

Maida y Pacienzia (2015) menciona que esta metodología se fundamenta principalmente en constituir un equipo fuerte y altamente preparado capaz de llevar a cabo cualquier tarea en poco tiempo, legando todo a la eficacia y la cohesión de los componentes del equipo y obviando los procesos y la burocracia que conlleva normalmente el tener un sistema de producción preestablecido. La filosofía Lean consiste en tener un equipo muy preparado, altamente motivado y muy unido. Se considera que cuanto más hayan aprendido los miembros del equipo y más unidos se sientan, la cantidad de tiempo y dinero necesario para llevar a cabo los desarrollos será cada vez menor (p. 98).

2.3 Conceptual

Sistema operativo

Un sistema operativo funciona como el software central encargado de gestionar todo tipo de procesos técnicos de una computadora, abarcando desde la conexión con dispositivos de hardware hasta la administración de la seguridad y comunicación en red (Beekman, 2005, p. 135).

Lenguaje de programación

Un lenguaje de programación es una herramienta que se encuentra a medio camino entre la comprensión humana y el lenguaje de la máquina. Su propósito es permitir a las personas resolver problemas utilizando una terminología familiar en lugar de tener que usar instrucciones máquina enigmáticas (Beekman, 2005, p. 126).

Líneas de comando

Es un entorno de usuario que permite comunicarse con el sistema operativo mediante la escritura de líneas de comando (Beekman, 2005, p. 600).

Hardware

El hardware constituye las partes físicas de una computadora. Cada uno de estos componentes físicos está especializado en realizar alguna de estas cuatro funciones básicas: recibir una entrada, procesar información, producir una salida y almacenar información (Beekman, 2005, p. 63).

Software

El software son las instrucciones que indican al hardware lo que tiene que hacer para transformar la entrada en salida. Es el componente que permite que las personas puedan comunicarle problemas a la computadora para que ésta les ofrezca una solución (Beekman, 2005, pp. 37, 123).

Servidor

Es un componente tangible que permite almacenar información de páginas web y transmitirla a los navegadores, dando respuesta a las solicitudes emitidas por los clientes (Beekman, 2005, p. 324).

Tabla 1: *Lenguajes de programación*

Lenguaje de programación	Descripción
TypeScript	Es un lenguaje de programación de código abierto creado por Microsoft en al año 2012, que implementa muchos mecanismos de programación orientada a objetos. Se trata de un super conjunto de JavaScript, lo que significa que extiende la sintaxis (Puciarelli, 2020, p. 9).
HTML	Es un lenguaje de hipertexto que nos permite definir la estructura de la página web mediante etiquetas. A su vez, nos permite desplegar documentos en un servidor web y ser accedidos mediante un dominio (Flórez & Hernández, 2021, p. 1).
PHP	PHP es un lenguaje de programación de propósito general, orientado al desarrollo web del lado del servidor. Su ejecución requiere de un intérprete que procese el código, apoyándose comúnmente en servidores web como Apache para gestionar las peticiones (Flórez & Hernández, 2021, p. 3).

CSS	Es un lenguaje de hojas de estilo utilizados para desarrollar el diseño y dinamismo de una página web escrito en un documento HTML (Flórez & Hernández, 2021, p. 2).
JavaScript	Es un lenguaje de programación interpretado, fundamental en el desarrollo web del lado del cliente, que permite implementar funciones complejas y dotar de interactividad a las páginas web (Flórez & Hernández, 2021, p. 3).

Nota. Lenguajes de programación para el desarrollo del sistema web.

Fuente: Elaborado por Renan Sigcha y Alexander Agualongo

Lenguaje SQL

El SQL es un lenguaje de consulta disponible para muchos sistemas de administración de bases de datos diferentes. Combina el uso de tablas, filas y columnas para que usuarios y programadores puedan acceder a bases de datos de diversos fabricantes empleando el mismo estándar (Beekman, 2005, pp. 254–255).

Gestor de base de datos

El gestor de base de datos es una interfaz que interactúa con el usuario y que permite almacenar, organizar, recuperar, comunicar y administrar información de un modo especial en la computadora (Beekman, 2005, pp. 243–244).

MySQL

Es un SGBD relacional, multihilo, multiusuario y multiplataforma. Está diseñado para su exploración en entornos web, y también funciona bien en base de datos de tamaño medio (Postigo, 2023, p. 9).

Analista de sistema

Un analista de sistemas es un profesional especializado en la tecnología de la información, responsable de evaluar la estructura organizacional y guiar el desarrollo de un sistema (Beekman, 2005, p. 533).

GPS

Es un hardware empleado para transmitir señales del sistema de posicionamiento global, con el propósito de establecer la localización y transmitir esa información a un usuario o computadora (Beekman, 2005, p. 604)

Geolocalización móvil

Palomino y García (2023) contextualizan la geolocalización en dispositivos móviles como la ubicación estimada en tiempo real, al momento que procesa una búsqueda de un equipo terminal móvil que se encuentra asociado a una línea telefónica determinada. El concepto destaca la importancia de los dispositivos móviles en obtener información de manera precisa sobre la ubicación de un usuario, monitoreo y rastreo vehicular.

Frontend

El frontend es la capa visual y de presentación de un sistema web, diseñada específicamente para que el usuario pueda interactuar con la plataforma. Su propósito principal es darle estilo a la página y organizar la información de una manera que resulte agradable y fácil de usar (Pérez et al., 2021).

Aplicaciones web Progresivas (PWA)

Rodríguez et al. (2020) define un PWA es una aplicación web que utiliza las tecnologías que se encuentran disponibles en los navegadores, permite tener un icono de acceso y trabajar offline utilizando información almacenada internamente en el dispositivo, ofreciendo una similitud a las aplicaciones nativas dentro de un dispositivo móvil. Los objetivos de PWA son lograr el mayor rendimiento, que cargue de manera instantánea, las interfaces parezcan lo máximo posible a una app nativa y enviar notificaciones a los usuarios.

Transporte mixto

Presidencia de la República (2016) la definición sobre el servicio que prestan estas compañías se detalla de la siguiente manera. Es el transporte de terceras personas y sus bienes en vehículos de hasta 1.2 toneladas de capacidad de carga, desde un lugar a otro, permitiendo el traslado en el mismo vehículo de hasta 5 personas que incluye el conductor que sean responsables de estos bienes y personas, estos deberán estar

provistos de una protección adecuada a la carga que transporten y prestar el servicio solo en el ámbito intraprovincial.

Backend

El backend es el motor interno de la aplicación y se encarga de manejar toda la lógica del sistema y la manipulación profunda de los datos. Además, se ocupa de construir herramientas como las APIS, que permiten que los datos se conecten y consuman fluidamente para garantizar una buena experiencia (Pérez et al., 2021).

Tailwind CSS

Es un framework orientado a utilidades que permite construir interfaces altamente personalizadas y responsivas mediante clases de bajo nivel, eliminando la necesidad de escribir código CSS extenso y evitando las limitaciones de las componentes previamente diseñadas (Biradar, 2025).

Alpine.js

Es un framework que permite la incorporación de la interactividad a una estructura HTML, sin tener la necesidad de escribir código JavaScript de manera directa. Se fundamenta en la utilización de atributos HTML específicos, los cuales se encargan de definir y controlar la interactividad en la interfaz (Alberti et al., 2023).

Laravel

Laravel es un framework de código abierto, facilita el desarrollo de la lógica de negocio de aplicaciones web, utilizando el lenguaje de programación PHP. Su estructura se apoya en el patrón de arquitectura Modelo-Vista-Controlador (MVC), lo que permite mantener el código organizado de forma modular (Murgueytio et al., 2022).

Patrón Observer

Debrauwer (2013) menciona que “el patrón Observer tiene como objetivo construir una dependencia entre un sujeto y los observadores de modo que cada modificación del sujeto sea notificada a los observadores para que puedan actualizar su estado” (p. 135).

Relación entre el patrón de diseño gof y marco de trabajo.

En el desarrollo del sistema web propuesto, se establece una integración estructurada entre patrones de diseño, frameworks y modelos arquitectónicos, con el objetivo de garantizar un software escalable, mantenible y alineado a buenas prácticas de ingeniería de software.

En primer lugar, se emplean frameworks modernos tanto para el backend como para el frontend. En el lado del servidor se utiliza Laravel, el cual permite estructurar la lógica de negocio bajo el patrón Modelo-Vista-Controlador (MVC), facilitando la organización modular del código y la gestión eficiente de peticiones mediante APIs.

Por otro lado, en el frontend se prescinde de frameworks de JavaScript, en su lugar se implementa el motor de plantilla Blade, utilizando estilos con el framework orientado a utilidades Tailwind CSS, que permitirá construir interfaces personalizadas, responsivas y flexibles hacia usuario final. La comunicación de los datos se realiza mediante servicios web de API REST, los cuales son consumidos de forma asíncrona desde el cliente con Alpine.js; esta alternativa permite incorporar interactividad rápida en la estructura HTML y desacoplar el flujo de información de la presentación visual.

A nivel de diseño de software, se incorpora el patrón de diseño GoF Observer, aplicado bajo el concepto de la creación de dependencias entre un sujeto y los observadores, cuando él sujeto cambia su estado, los observadores son notificados para que actualicen su nuevo estado. En el contexto del sistema, este patrón se aplica en los módulos de monitoreo y control de eventos, tales como el arribo, despacho o cambios de estado de unidades, permitiendo que diferentes componentes del sistema reaccionen en tiempo real ante dichos cambios, mejorando la trazabilidad y consistencia de la información.

Desde el punto de vista estructural, el sistema se organiza bajo niveles del patrón MVC lo que implica la separación de responsabilidades en diferentes niveles, tales como:

Modelo: Es el encargado de la abstracción de datos y la persistencia que se aplica mediante el ORM Eloquent, interactúa directamente con el motor de base de datos MySQL para gestionar la información de turnos, multas y geolocalización.

Vista: Muestra el resultado visual mediante la interfaz al usuario final, aplicada mediante plantillas Blade y Tailwind CSS, proporcionadas con Alpine.js para consumir la API REST sin tener que recargar la página.

Controlador: Es el intermediario del flujo de trabajo en el framework laravel Su responsabilidad se limita a receptar solicitudes HTTP y valida los parámetros de entrada.

Esta organización permite mejorar la mantenibilidad, facilitar pruebas y reducir el acoplamiento entre componentes, alineándose con buenas prácticas de desarrollo de software.

Adicionalmente, para describir el sistema desde múltiples perspectivas, se adopta el modelo de vistas arquitectónicas 4+1, el cual permite representar la arquitectura del sistema de manera integral mediante: La combinación de estas vistas permite una mejor comprensión del sistema por parte de desarrolladores, analistas y stakeholders, facilitando su diseño, implementación y validación.

Para complementar este enfoque técnico, se adopta una metodología de desarrollo ágil híbrida, basada en la combinación de Kanban y Extreme Programming (XP), con el propósito de optimizar tanto la gestión del flujo de trabajo como la calidad del desarrollo del software. Por un lado, Kanban permite visualizar y gestionar las tareas mediante un tablero de trabajo, facilitando el control del progreso, la identificación de cuellos de botella y la mejora continua del proceso. Por otro lado, XP aporta prácticas técnicas como la programación incremental, la retroalimentación constante y la simplicidad en el diseño, lo que contribuye a mejorar la calidad del código y la adaptabilidad ante cambios en los requerimientos, sobre todo una metodología adaptada principalmente para proyectos que no tienen claramente definido los requerimientos desde un inicio.

2.4 Legal

2.4.1 Ley de Propiedad Intelectual

Según el Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, en lo referente a las disposiciones de los derechos de autor:
La protección de los derechos de autor se otorga sin consideración del género, mérito, finalidad, destino o algún modo de expresión de la obra. [...] Los derechos reconocidos y concedidos por el presente título son independientes de la propiedad del objeto

material en el cual está incorporada la obra [...] [y la] adquisición y ejercicio de los derechos de autor y de los derechos conexos no están sometidos a registro o depósito, ni al cumplimiento de formalidad alguno (Asamblea Nacional de La República Del Ecuador, 2016, arts.101, 102, 104)

Entre las diferentes obras protegidas por este código se encuentra el software referido a las disposiciones especiales sobre ciertas obras, el apartado del software y bases de datos estipula lo siguiente:

El software se protege como obra literaria. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea como código fuente, es decir, en forma legible por el ser humano; o como código objeto, es decir, en forma legible por máquina, ya sea sistemas operativos o sistemas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa (Asamblea Nacional de La República Del Ecuador, 2016, art. 131).

En cuanto a la titularidad y los derechos sobre dichas creaciones informáticas, la misma legislación manifiesta que:

Es titular de los derechos sobre el software el productor, esto es, la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se presumirá titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual. Dicho titular está además autorizado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación. El productor tiene el derecho exclusivo de impedir que terceras personas realicen sin su consentimiento versiones sucesivas del software y software derivado del mismo. Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor (Asamblea Nacional de La República Del Ecuador, 2016, art. 133).

2.4.2 Ley Orgánica de Protección de Datos Personales

El Artículo 42 de la Ley Orgánica de Protección de Datos Personales establece la obligatoriedad de una evaluación de impacto bajo ciertas condiciones:

Evaluación del impacto del tratamiento de datos personales. -El responsable realizará una evaluación de impacto del tratamiento de datos personales cuando se haya identificado la probabilidad de que dicho tratamiento, por su naturaleza, contexto o fines, conlleve un alto riesgo para los derechos y libertades del titular o cuando la Autoridad de Protección de Datos Personales lo requiera.

La evaluación de impacto relativa a la protección de los datos será de carácter obligatoria en caso de:

a) Evaluación sistemática y exhaustiva de aspectos personales de personas físicas que se base en un tratamiento automatizado, como la elaboración de perfiles, y sobre cuya base se tomen decisiones que produzcan efectos jurídicos para las personas naturales;

b) Tratamiento a gran escala de las categorías especiales de datos, o de los datos personales relativos a condenas e infracciones penales, o

c) Observación sistemática a gran escala de una zona de acceso público.

La Autoridad de Protección de Datos Personales establecerá otros tipos de operaciones de tratamiento que requieran una evaluación de impacto relativa a la protección de datos La evaluación de impacto deberá efectuarse previo al inicio del tratamiento de datos personales (Asamblea Nacional de La República Del Ecuador, 2021, p. 35, art. 42).

El Artículo 43 de la Ley Orgánica de Protección de Datos Personales establece los plazos para la notificación de vulneraciones de seguridad:

Art. 43. Notificación de vulneración de seguridad.-El responsable del tratamiento deberá notificar la vulneración de la seguridad de datos personales a la Autoridad de Protección de Datos Personales y la Agencia de Regulación y Control de las Telecomunicaciones, tan pronto sea posible, y a más tardar en el término de cinco (5) días después de que haya tenido constancia de ella, a menos que sea improbable que dicha violación de la seguridad constituya un riesgo para los derechos y las libertades de las personas físicas. Si la notificación a la Autoridad de Protección de Datos no tiene lugar en el término de cinco (5) días, deberá ir acompañada de indicación de los motivos de la dilación.

El encargado del tratamiento deberá notificar al responsable cualquier vulneración de la seguridad de datos personales tan pronto sea posible, y a más tardar dentro del término de dos (2) días contados a partir de la fecha en la que tenga conocimiento de ella (Asamblea Nacional de La República Del Ecuador, 2021, pp. 35–36, art. 43).

2.4.3 Código Ingenios

Artículo 1.- Objeto. - El presente Código tiene por objeto normar el Sistema Nacional de Ciencia, Tecnología, Innovación y Saberes Ancestrales previsto en la Constitución de la República del Ecuador y su articulación principalmente con el Sistema Nacional de Educación, el Sistema de Educación Superior y el Sistema Nacional de Cultura, con la finalidad de establecer un marco legal en el que se estructure la economía social de los conocimientos, la creatividad y la innovación. (Asamblea Nacional de La República Del Ecuador, 2016, Art. 1).

Artículo 2.- Ámbito. - Se rigen por el presente Código todas las personas naturales, jurídicas y demás formas asociativas que desarrollen actividades relacionadas a la economía social de los conocimientos, la creatividad y la innovación.

Las actividades relacionadas a la economía social de los conocimientos, la creatividad y la innovación son aquellas enfocadas a la creación de valor a partir del uso intensivo de la generación, transmisión, gestión y aprovechamiento del bien de interés público conocimiento, que incluye los conocimientos tradicionales; promoviendo en todos los sectores sociales y productivos la colaboración y potenciación de las capacidades individuales y sociales, la democratización, distribución equitativa, y aprovechamiento eficiente de los recursos en armonía con la naturaleza, dirigido a la obtención del buen vivir (Asamblea Nacional de La República Del Ecuador, 2016, Art. 2).

Artículo 3.- Fines. - El presente Código tiene, como principales, los siguientes fines:

1. Generar instrumentos para promover un modelo económico que democratice la producción, transmisión y apropiación del conocimiento como bien de interés público, garantizando así la acumulación y redistribución de la riqueza de modo justo, sostenible y en armonía con la naturaleza;
2. Promover el desarrollo de la ciencia, la tecnología, la innovación y la creatividad para satisfacer necesidades y efectivizar el ejercicio de derechos de las personas, de los pueblos y de la naturaleza;
3. Incentivar la producción del conocimiento de una manera democrática colaborativa y solidaria;
4. Incentivar la circulación y transferencia nacional y regional de los conocimientos y tecnologías disponibles, a través de la conformación de redes de innovación social,

de investigación, académicas y en general, para acrecentarlos desde la práctica de la complementariedad y solidaridad;

5. Generar una visión pluralista e inclusiva en el aprovechamiento de los conocimientos, dándole supremacía al valor de uso sobre el valor de cambio;
6. Desarrollar las formas de propiedad de los conocimientos compatibles con el buen vivir, siendo estas: pública, privada, comunitaria, estatal, asociativa y mixta;
7. Incentivar la desagregación y transferencia tecnológica a través de mecanismos que permitan la generación de investigación, desarrollo de tecnología e innovación con un alto grado de componente nacional;
8. Promover la distribución justa y equitativa de los beneficios derivados de las actividades vinculadas a la generación, transmisión, gestión, uso y aprovechamiento de los conocimientos, la tecnología, la innovación y los conocimientos tradicionales, así como el uso eficiente de los factores sociales de la producción para incrementar el acervo de conocimiento e innovación;
9. Establecer las fuentes de financiamiento y los incentivos para el desarrollo de las actividades de la economía social de los conocimientos, la creatividad y la innovación;
10. Fomentar el desarrollo de la sociedad del conocimiento y de la información como principio fundamental para el aumento de productividad en los factores de producción y actividades laborales intensivas en conocimiento;
11. Fomentar la protección de la biodiversidad como patrimonio del Estado, a través de las reglas que garanticen su aprovechamiento soberano y sustentable, proteger y precautelar los derechos de las comunidades, pueblos y nacionalidades sobre sus conocimientos tradicionales y saberes ancestrales relacionados a la biodiversidad; y evitar la apropiación indebida de la biodiversidad y los conocimientos tradicionales asociados a esta (Asamblea Nacional de La República Del Ecuador, 2016, Art. 3).

CÓDIGO INTERNO COMPAÑÍA DE TRANSPORTE ANTOÑO POLO S.A.

TERCERA: Estatuto de la compañía de transporte de carga liviana "Antoño Polo S.A"

CAPÍTULO PRIMERO

Art. 2.- Objeto social. - La Compañía tendrá por objetivo el servicio público de transporte de carga liviana dentro de la ciudad de Guaranda y fuera de ella, dentro del ámbito provincial, de acuerdo con las disposiciones emanadas para el efecto por la Comisión Nacional de Transporte Terrestre, Tránsito y Seguridad Vial; y los demás organismos de tránsito y transporte terrestre respectivos, y utilizando para esta actividad, vehículos tipo camioneta apropiados para el efecto, al fin de brindar un eficiente servicio dentro de esta modalidad de transporte.

CAPÍTULO SEGUNDO. Régimen normativo y Administrativo de la compañía.

Art. 13.- La junta general formada por los accionistas legalmente convocados y reunidos es el órgano supremo de la compañía; y, la compañía estará administrada por el presidente y Gerente General de la misma. Los administradores cumplirán con todas las obligaciones que la ley establece con su cargo y responsabilidad y de acuerdo con los presentes estatutos se dispone para cada uno de los órganos de administración.

Art. 31.- Los permisos de Operación que reciba la Compañía para la presentación del servicio de transporte de carga pesada, serán autorizados por los componentes organismos de tránsito, y no constituyen títulos de propiedad, por consiguiente, no es susceptible de negociación.

Art. 32.- La Reforma de Estatuto, aumento o cambio de unidades, variación de servicios y más actividades de tránsito y transporte terrestres. Lo efectuará la Compañía, previo informe favorable de los Organismos de Tránsito.

Art. 33.- La compañía en todo lo relacionado con el tránsito y transporte terrestres, se someterá a la Ley de Tránsito vigente y sus Reglamentos; y, a las regulaciones que sobre esta materia dictaren los organismos competentes de tránsito.

Art. 34.- En todo lo que no estuviera previsto en los presentes estatutos, la Sociedad se sujetará a la Ley de Compañías, en vigencia; y, a las demás leyes vigentes en la República del Ecuador.

Tras la revisión de la normativa legal se comprende que el desarrollo de software es aplicable y que la implementación del sistema web para la gestión operativa y control automatizado es viable acorde a las leyes establecidas y no existe ningún impedimento jurídico para su ejecución. La ley orgánica de protección de datos establece un marco explicito sobre el adecuado manejo de la información las cuales pueden ser utilizados dentro del sistema sin ningún impedimento, también el código Ingenios respalda este tipo de proyectos sobre el desarrollo tecnológico e innovación. Finalmente, los estatutos internos de la compañía se evidencian que no existen disposiciones o normas que limiten la implementación de herramientas tecnológicas; la implementación de este sistema está orientada a la modernización de los procesos que se desarrollan dentro de la compañía, sin que esto afecte, modifique o altere de manera negativa a la operación del transporte, los cuales están en competencia exclusiva de los organismos de tránsitos reguladoras.

2.5 Georreferencial

Tabla 2: *Ubicación Geográfica.*

UBICACIÓN GEOGRÁFICA DEL TRABAJO DE TITULACIÓN
TEMA: Sistema web para la gestión operativa y control automatizado del arribo y despacho de unidades mediante Geolocalización en puntos establecidos en la compañía de transporte Antoño Polo S.A.



Datos de referencia (Titulación)	**Fuente de información:** Google Maps
Elaborado por: Renan Sigcha y Alexander Agualongo	**Referencia:** Parroquia Salinas
Director: Ing. Darwin Carrión	
Ubicación Geografica:	
latitud: - 1,4050879	
Longitud: -79,0195530	

Nota: *Datos obtenidos mediante Google Maps*

CAPÍTULO III

METODOLOGÍA

3.1 Metodología de Desarrollo de Software

Maida y Pacienzia (2015, pp. 12–13) menciona que las metodologías del desarrollo de software es un modo sistemático de ejecutar, administrar y gestionar proyectos con una alta tasa de probabilidad de éxito en su ejecución. Estas metodologías comprenden seguir las reglas establecidas con el fin de tener noción, implementar y mantener un producto de software, abarcando desde el momento que nace una necesidad hasta llegar al final de los objetivos planteados en el proyecto. Las metodologías de desarrollo de software comprenden un marco de trabajo que sirve para estructurar, planificar y llevar el control del proceso del producto; se articula a través de dos componentes importantes debe tener una filosofía definida sobre el proceso del desarrollo y múltiples herramientas, métodos y sus modelos diseñados misma que sirve para la asistencia técnica y operativa dentro de su construcción.

Conde (2025) desarrollo aplicativo web de geolocalización de siniestros de tránsito donde menciona que Kanban “Forma parte de las metodologías ágiles y su objetivo es gestionar la realización de las tareas hasta su finalización, al ser un método visual permite que con un simple vistazo se conozca el estado de los proyectos y se puedan asignar nuevas tareas de manera ordenada".

Tomalá (2025) argumenta que la combinación de Extreme Programming (XP) y Kanban destaca por su capacidad para complementar el flujo de trabajo y asegurar una entrega eficiente y de alta calidad. El autor concluye en su investigación que tomar la decisión de fusionar ambas herramientas establece una organización clara, lo que a su vez ha permitido una priorización efectiva de tareas y una adaptación rápida a los cambios, lo que resulta en un proceso de desarrollo más eficiente y colaborativo.

Para el desarrollo de este sistema se decidió trabajar con la metodología ágil hibrida Kanban y XP, esta combinación integra estratégicamente sus fortalezas. Por un lado, la segunda metodología tiene una adaptabilidad a proyectos en el cual los requerimientos a menudo son cambiados y se busca estabilizar un código de calidad, mientras que la

primera maneja la gestión visual del trabajo, permitiendo organizar tareas, identificar bloqueos y estabilizar el proceso de desarrollo constantemente. La obtención de los requerimientos se manejará mediante las entrevistas y la observación del proceso operativo que lleva a cabo la compañía.

El uso combinado de XP y Kanban permitirá desarrollar el sistema de forma ágil, colaborativa y con entregas frecuentes, garantizando que el producto final sea funcional, escalable y logre resolver la desconexión actual entre la información administrativa y operativa.

Valores y principios de las metodologías

Valores de XP

Comunicación: Se establecerán canales directos y permanentes con el Gerente General y la tarjetera, quienes son los actores que conocen el funcionamiento real del sorteo manual, los intervalos de salida y las reglas de sanción. Las entrevistas presenciales y las consultas por mensajería serán el medio principal para resolver ambigüedades sobre las reglas internas, evitando interpretaciones erróneas durante el desarrollo

Simplicidad: El sistema se diseñará para resolver únicamente las necesidades reales identificadas en el levantamiento de requerimientos, evitando funcionalidades que son innecesarias.

Retroalimentación: Al finalizar cada iteración se va a realizar pruebas para verificar el funcionamiento, esto servirá para realizar ajustes de la gestión de turnos, rotaciones de turnos y las capturas de las coordenadas de geolocalización mediante el dispositivo móvil.

Valentía: El equipo asumirá la disposición de modificar partes del código al detectar mejoras en la lógica de negocio, especialmente en el cálculo automatizado del tiempo de trayecto y en la generación de multa.

Respeto: se valorará el conocimiento operativo de la tarjetera y de los conductores, quienes aportarán con todo el proceso manual que realiza y en consecuencia permitirá comprender las situaciones excepcionales que el sistema deberá contemplar.

Kanban

Visualizar el flujo de trabajo: se utilizará un tablero Kanban dividido en cinco columnas (Backlog, Por hacer, En desarrollo, En pruebas y Hecho) en el que se colocarán las historias de usuario derivadas de las entrevistas. Con el fin que ambos desarrolladores tengan visibilidad inmediata del estado de cada tarea relacionada con módulos como autenticación, registro de unidades, sorteo de turnos y captura de geolocalización.

Limitar el trabajo en progreso: se establecerá un máximo de dos historias en desarrollo simultáneo, considerando que el equipo está conformado por dos integrantes. Esta restricción evitará la dispersión y garantizará que cada funcionalidad sea completada antes de iniciar la siguiente, manteniendo la calidad del código.

Mejora continua: al cierre de cada iteración se realizará una breve revisión interna del tablero para identificar bloqueos recurrentes y plantear ajustes en el proceso, tales como la reorganización de prioridades o la incorporación de nuevas pruebas automatizadas.

Las fases del desarrollo no son secuenciales sino iterativas la primera metodología aporta las prácticas técnicas y el ritmo de trabajo por iteraciones, mientras que la segunda provee la visibilidad continua del avance mediante el tablero. Alineado a las metodologías seleccionadas, el proceso de comprensión y desarrollo software se presenta a través de las siguientes fases:

Planificación

La planificación se construirá a partir de entrevistas directas con la operadora y el personal administrativo, quienes describirán sus procesos actuales, sus dificultades diarias y las reglas internas que el sistema deberá respetar. De estas sesiones se extraerán las historias de usuario, que representarán funcionalidades especificas desde la perspectiva de quienes utilizarán el sistema como el Administrador, la Tarjetera y los Conductores.

Con base en esas historias se elaborará un plan de entregas organizado por prioridad operativa. Las funcionalidades que resuelvan los problemas más críticos de la compañía serán ubicadas en las primeras iteraciones: no exceder la llegada dentro de una hora, la

verificación de licencias y matrículas vigentes antes de asignar un turno, gestión de multas y el registro de llegada de unidades mediante coordenadas GPS (dispositivo móvil). Las funcionalidades de soporte, como la generación de reportes, se planificarán para iteraciones posteriores una vez que se estabilice la lógica central del sistema.

Diseño

En esta fase se busca definir la estructura y la funcionalidad del sistema de geolocalización con el fin de obtener la facilidad, simplicidad y la estabilidad. Dentro de esta fase se aplicarán los principios de la metodología Xp, donde se puede realizar la retroalimentación e incorporar cambios conforme se sigue avanzando. El sistema se va estructurar bajo el patrón arquitectónico MVC.

Vista: Desarrollo visual mediante el motor de plantillas Blade e interfaces dinamizadas con Alpine.js diseñadas con Tailwind CSS, mostrará al Administrador y a la Tarjetera un mapa interactivo con la posición de las unidades si llegó o salió y el estado de cada turno activo, consumiendo los datos de la API sin necesidad de recargar la página. Esta interfaz reemplazará definitivamente el registro manual en papel que actualmente utiliza la tarjetera.

Controlador: Utilizando el framework laravel, fungirá como motor central que expondrá la API REST y agrupará las reglas operativas de la organización. Mostrará el cálculo de resultado del tiempo de llegada a las paradas en el intervalo de 60 minutos, verificará los documentos del conductor y de las unidades antes de la jornada y la sanción correspondiente ante los incumplimientos.

Modelo: gestionado mediante el ORM Eloquent, encargado de abstraer y estructurar la búsqueda en bases de datos MySQL. El almacenamiento de los registros unidad, conductor, historias de tunro, coordenadas de GPS y multas se encontrarán de manera segura garantizando que la información se encuentre disponible en el sistema.

Para documentar el sistema desde distintos ángulos se utilizará el modelo de vistas 4+1 que permitirá representar tanto la estructura interna del software como el flujo real de operaciones de la compañía, la vista lógica, vista de procesos, vista física, vista de despliegue, vista de escenarios.

Codificación

En esta fase se desarrollará siguiendo los principios de Extreme Programming (XP) juntamente con Kanban los cuales son programación en parejas, integración continua y refactorización, se ejecutará en iteraciones, priorizando que cada ciclo produzca funcionalidades operativas reales.

Cada iteración abordará un conjunto específico de necesidades de Antoño Polo S.A.:

Iteración 1: Autenticación y control de acceso por roles, diferenciando las funciones del Administrador, socio(conductor) y la Tarjetera dentro del sistema.

Iteración 2: Registro de socios y unidades, donde se maneja la vigencia de licencias y matrículas al momento de asignar un turno.

Iteración 3: Lógica de turnos y control del proceso de arribo y despacho, incluyendo el cálculo del intervalo de 15 minutos entre salidas.

Iteración 4: Integración con la API de Google Maps, captura de coordenadas GPS desde dispositivo móvil y cálculo automatizado del tiempo de trayecto con generación de sanciones.

Iteración 5: Historial operativo y módulo de reportes para la toma de decisiones administrativas.

Durante todo el desarrollo se aplicarán prácticas de XP, la programación en parejas permitirá que un integrante construya la lógica de operativa de la compañía, mientras el otro verifica su correcta persistencia en la base de datos.

Pruebas

Cada iteración incluirá su propia fase de pruebas, lo que permitirá detectar errores en el momento en que se produzcan y no al final del proyecto. Las pruebas se diseñarán en función de las reglas operativas reales de la compañía.

Las pruebas unitarias validarán funciones específicas del negocio: por ejemplo, que el sistema calcule correctamente si han transcurrido más de 60 minutos desde el despacho de una unidad, o que bloquee la asignación de turno cuando una matrícula esté vencida.

Las pruebas de integración verificarán que las coordenadas capturadas por el dispositivo móvil (GPS) lleguen correctamente al backend y se reflejen en el mapa visible para la tarjetera sin pérdida ni distorsión de datos.

Las pruebas de aceptación se realizarán directamente con el personal de Antoño Polo S.A., quienes evaluarán el sistema frente a sus procesos reales. Su retroalimentación será determinante para los ajustes.

Las pruebas de regresión acompañarán cada nueva iteración para garantizar que la incorporación de nuevos módulos no afecte el funcionamiento de los componentes ya validados.

3.2 Técnicas e Instrumentos de Recopilación de Datos

Entrevista

Medina et al. (2023) establecen que "la entrevista es una técnica de investigación que involucra la interacción directa entre el entrevistador y el entrevistado con el objetivo de obtener información y opiniones detalladas sobre un tema específico, La efectividad de la entrevista depende en gran medida de las habilidades que posee el investigador como destrezas para realizar preguntas claras, capacidad de escucha y generar confianza con el entrevistado"

Para el desarrollo del presente proyecto la recolección de información se fundamentó de manera exclusiva en la técnica de la entrevista. En esta fase de levantamiento de requerimientos dentro de la compañía, se estableció la entrevista presencial y virtual con los actores clave involucrando directamente al gerente general y a la tarjetera responsable de la coordinación operativa en las paradas. Esta interacción se apoyó en una guía de preguntas abiertas, diseñada estratégicamente para delimitar las funciones exactas, las restricciones logísticas y las necesidades reales.

En el transcurso de las sesiones que realizaron la administración de la compañía y sus respectivos socios, demostró una clara disposición hacia la modernización de sus procesos. Hubo un consenso sobre la necesidad imperativa de sustituir los registros de papel por un ecosistema de sistema web operativa y de geolocalización, reconociendo que este salto tecnológico es vital para erradicar los conflictos diarios y garantizar el cumplimiento estricto del reglamento interno. Anexo 3

Observación

Arias (2020) la observación es la técnica de recolección que permite obtener la información de manera directa sobre una realidad, permitiendo examinar, medir, interpretar y evaluar de manera cuidadosa un fenómeno en particular. Mediante este instrumento el investigador tiene la capacidad de registrar y analizar los acontecimientos de un hecho empírico, incluso puede extraer datos subjetivos como las emociones, actitudes y comportamientos de la población, como resultado se logra recopilar esta información ya sea a través de un proceso sistemático o visualizando el estado de un fenómeno.

Esta técnica de la observación permitió conocer el funcionamiento adherido a la gestión operativa, registro de horarios de rutas, reglas de sanciones o multas, registro de unidades para su turno de manera manual en las paradas de salinas y Guaranda. Anexo 4.

CAPÍTULO IV

INGENIERÍA DEL PROYECTO

El desarrollo del sistema, teniendo un enfoque en la automatización de control de las operaciones de las unidades, transparencia en las multas y reglas implementadas en la compañía. Se adoptó una metodología híbrida basada en Kanban y Extreme Programming (XP), la combinación de estas permite desarrollar un trabajo eficiente mediante herramientas de visualización, garantizando un software de calidad mediante prácticas como pruebas continuas, refactorización y la integración constante del código. Entre esta combinación se estableció fases: planificación, diseño, codificación y pruebas.

4.1 Planificación

4.1.1 Análisis de instrumentos

Nombres de los entrevistadores: Renan Marcial Sigcha Punina y Steven Alexander Agualongo Morocho.

Nombre del entrevistado: Leonardo Guamán

Cargo: Gerente General

1. **¿Cómo inicia la jornada laboral diaria y cuál es el horario del primer turno establecido?**

Respuesta: En la parada de Salinas, el turno inicia a las 5:30 AM con dos unidades que denominamos punteros las cuales rotan diariamente. Para garantizar la puntualidad, los conductores de estas unidades deben enviar una fotografía al grupo de WhatsApp. Si no cumplen con este reporte o a su vez no asiste como puntero se le permite trabajar hasta las 9 am.

2. **Para el resto de los conductores, ¿de qué manera se realiza la asignación de turnos para el día laboral?**

Respuesta: Se realiza un sorteo manual. Los conductores que llegan a la parada meten la mano en una funda negra que contiene papeles numerados aleatoriamente del 1 al 20. El número que sacan determina su orden de salida, esto se procedimiento se realiza en cada parada.

3. **¿Existe alguna dinámica operativa especial debido a la alta demanda de pasajeros en horas pico?**

Respuesta: Sí. De lunes a viernes tenemos una gran demanda de estudiantes que bajan desde Salinas hacia Guaranda. Para cubrir esto, las primeras 8 unidades salen de manera continua entre las 5:45 AM y las 6:00 AM, llenándose casi de inmediato.

4. **Al manejar todo de forma manual los sorteos, papelitos y WhatsApp, ¿cuáles son los problemas más frecuentes que enfrentan?**

Respuesta: El principal problema son los atrasos o las confusiones con los números. Si un conductor llega tarde y su turno ya pasó, pierde su lugar y debe irse a la cola, es decir, esperar a que salgan todas las demás unidades para volver a integrarse.

5. **¿Cuáles son las mayores limitaciones de utilizar registros físicos en lugar de un sistema?**

Respuesta: El registro en papel dificulta tener un control exacto. Muchas veces los conductores parquean sus unidades cuadras antes para evitar roces o hacer ruido, y caminan hasta la parada para el sorteo. Controlar visualmente el listado de turno en un cuaderno y controlar el exceso de tiempo de llegada ya que tiene que comunicar hacia otra parada para saber qué hora salió tal unidad o comunicar directamente por teléfono con la tarjetera.

6. **¿Qué expectativas tiene la compañía con la implementación de este nuevo sistema web basado en geolocalización móvil?**

Respuesta: La expectativa principal es modernizar y transparentar el servicio. Esperamos que el sistema controle automáticamente los intervalos de despacho, valide las llegadas a través del GPS de los teléfonos móviles y genere las sanciones por

incumplimiento de forma imparcial, eliminando el desorden de los sorteos manuales y las discusiones.

7. **¿Cómo está distribuida la flota para la operatividad y los días de descanso?**
   Respuesta: La compañía cuenta con 44 unidades en total. Para garantizar la equidad, operan 37 unidades diariamente, mientras que se establece un grupo fijo de 7 unidades que tienen descanso obligatorio cada día.
8. **¿Cuál es la lógica de rotación anual para los grupos de descanso?**

Respuesta: Los grupos de 7 unidades rotan su día libre de forma escalonada. Si un grupo descansa el día lunes en la semana actual, la siguiente semana le corresponderá descansar el martes, y así sucesivamente, completando un ciclo rotativo ininterrumpido durante todo el año. Al finalizar el año, se reasignan los grupos.

9. **¿Cómo se gestionan los primeros turnos de la mañana denominados punteros?**
   Respuesta: Las unidades punteras son aquellas encargadas de abrir la jornada del día . Para evitar desorganización en la madrugada,
10. **¿Cuáles son los intervalos de tiempo reglamentarios para el despacho de las unidades?**

Respuesta: En horario de demanda normal las unidades se despachan con un intervalo de 15 minutos. Sin embargo, en horas de alta demanda o pico intervalo de salida se reduce a 10 minutos para agilizar el flujo de pasajeros.

11. ¿Qué excepción se aplica si una unidad alcanza su capacidad máxima antes de cumplir su tiempo de intervalo?

Respuesta: Si una unidad se llena sale antes de los 10 o 15 minutos reglamentarios, se le autoriza salir anticipadamente. En este caso, la tarjetera debe registrar la salida adelantada para que el turno sea asumido inmediatamente por la siguiente unidad en la fila.

12. **¿Qué sanción se aplica si un conductor no está presente en la parada al momento de su turno?**

Respuesta: El conductor pierde su lugar en el despacho y es enviado automáticamente a la cola. Esto significa que debe reincorporarse al final de la fila y esperar a que salgan todas las unidades que estaban detrás de él para poder operar.

13. **¿Cómo afecta el registro manual actual a la comunicación entre las paradas de Salinas y Guaranda?**

Respuesta: Actualmente, la tarjetera de una parada depende de enviar mensajes por WhatsApp o realizar llamadas telefónicas a la otra parada para informar que una unidad salió con carrera o para confirmar su hora de despacho. Esto genera vacíos de información si el mensaje no es leído a tiempo o si hay olvidos manuales.

14. ¿Cómo se gestiona actualmente el cobro de estas sanciones económicas?

Respuesta: Las multas se cobran físicamente en la parada. La regla principal de negocio es que el trayecto entre paradas no debe exceder los 60 minutos, si llega exceder se cobra $1.00 USD por cada minuto de retraso Si la infracción ocurre de lunes a viernes, el conductor debe pagar inmediatamente a la tarjetera. Si es los fines de semana, se permite cierta flexibilidad para que el socio opere y cancele el valor posteriormente a la compañía.

15. ¿Cómo maneja la cooperativa el volumen máximo de turnos asignados en las paradas principales?

Respuesta: Por lo general, se establece un límite operativo de hasta 20 unidades formadas en una parada al mismo tiempo. Las unidades restantes esperan su rotación o descansan, garantizando que no haya una aglomeración innecesaria en los puntos de despacho de Salinas o Guaranda.

Esta entrevista permitió conocer de manera clara el funcionamiento operativo de la compañía, identificar reglas de negocio vigentes y definir los problemas reales

existentes que justifican el desarrollo del tema antes mencionado. Los resultados se categorizan en tres áreas fundamentales para el desarrollo del software:

Dinámica Operativa Actual: La jornada laboral que posee la compañía incia entre las 5:00 a.m y 5:30 am. con las dos unidades designadas en el turno denominadas como "punteros". Posterior se designa turnos para el resto de unidades mediante el sorteo físico en cada uno de las paradas Salinas y Guaranda.

Reglas de Negocio: Se determinaron directrices operativas estrictas que el sistema web deberá calcular y automatizar de forma obligatoria:

- El despacho de unidades debe ejecutarse en intervalos regulares de 15 minutos, con una excepción de reducción de tiempo durante las horas pico.
- Existe una penalización logística estricta: el conductor que incumple su turno pierde su posición y es relegado al final de la cola de espera, requiriendo justificación previa, también si excede los 60 minutos entre la parada y llega existe una multa de 1 dólar por minuto.
- Se exige una rotación ininterrumpida que garantice el descanso obligatorio de 7 unidades diarias.

Identificación de la Problemática y Riesgos: Se identifico la gestión actualmente se encuentra basada en el registro visual cuadernos físicos y el uso de mensajería (WhatsApp), genera un alto grado de vulnerabilidad e inconsistencia en la información. Los puntos críticos detectados incluyen:

- Constantes confusiones operativas y pérdida de datos en la generación de turnos debido a la naturaleza manual del sorteo.
- Dificultad de los conductores para acceder visualmente a su posición en el listado de turnos.
- Un control ineficiente del tiempo de trayecto, ya que la validación de arribo depende exclusivamente de llamadas telefónicas entre los controladores de las paradas de origen y destino.

4.1.2 Análisis de observación

Para el desarrollo de la observación se realizó de manera directa, con el objetivo de realizar un diagnóstico actual de la gestión operativo de la compañía centrando en la logística de la movilidad y los turnos de las unidades en las paradas establecidas. Mediante este instrumento se desglosó el desarrollo de las actividades destacadas, a continuación, se detalla en la siguiente tabla.

Tabla 3: *Análisis de la observación.*

N°	**Actividades**	**Intervienen en el proceso**	**Herramientas utilizadas**	**Observaciones**
1	Registro de arribo origen	operadora de turno (Tarjetera).	Cuaderno de apuntes	El orden de salida de las camionetas de la parada se documenta a mano. genera reclamos inmediatos de chofer en caso de errores de turno.
2	Salida de la unidad (camionetas)	Chofer y operadora de turno (Tarjetera)	WhatsApp (dispositivo móvil)	Es notificado por la operadora verbalmente o revisa el listado de orden de turno en WhatsApp
3	Llegada de la unidad (camioneta)	Chofer de la unidad	Comunicación verbal o WhatsAp	Pregunta verbalmente el turno que le toca obligatoriamente y si no lo hace o la operadora está ocupada en otros registros no puede conocer el orden de turno designado.

4	Registro de llegada	operadora de turno (Tarjetera).	Cuaderno de apuntes	Para conocer el tiempo en la ruta cada unidad debe de comunicar con la operadora de otra parada.

A partir de la observación se determinaron los siguientes hallazgos:

Gestión documentaria vulnerable: El control de la salida y llegada dependen de anotaciones físicas en cuadernos manuales. Esta práctica en ocasiones genera errores al momento de transcribir misma que proporciona una desconfianza y reclamos por parte de los conductores

Centralización de comunicación: Se llego a identificar que los conductores carecen de la autonomía para conocer su turno, el cual depende de la tarjetera en informar verbalmente o mediante el grupo WhatsApp

Cálculo de tiempo de ruta: Al no existir un reloj centralizado la validación de viaje entre paradas, la comprobación se realiza mediante llamadas telefónicas generando dificultades al conocer la información de la salida de las unidades y realizar el cobro transparente de sanciones

Los resultados sostienen técnicamente el desarrollo del sistema web PWA donde se traslada el manejo manual de la compañía a una automatización de procesos en gestión operativa, gestión de turnos y sanciones respectivas.

Gracias a la información recopilada de las fases (Entrevista y Observación), se ha obtenido con claridad las reglas del negocio, los roles de cada actor involucrado y las dificultades en el proceso operativo de la compañía. Con base a la sustentación de los resultados obtenidos, se establecen y estructuran los requerimientos del sistema a través de las historias del usuario. Esto permitirá transformar las deficiencias de la organización en funcionalidades concretas de software, estableciendo un desarrollo alineado a la metodología ágil Extreme Programming (XP).

Restricciones del sistema

Tras utilizar las herramientas para recabar la información requeridas y comprender el funcionamiento de la compañía, se establece restricciones que condicionan el comportamiento del sistema. Cada restricción esta adherido con la historia de usuario y el componente del software que se aplica, esto permite llevar la trazabilidad en el desarrollo del sistema desde la regla del negocio de la compañía hasta la codificación.

Tabla 4: *Restricciones del sistema y su trazabilidad*

ID	Restricción	HU asociada	Componente que la aplica
RES-01	En la gestión de turnos, el sistema debe llevar una seguridad en el despacho de las unidades, ejecutando en intervalos de 15 minutos en flujo operativo normal. También debe llevar una parametrización, reduciendo a 10 minutos en alta demanda operativa y una salida anticipada, específicamente cuando la unidad alcance su capacidad máxima de pasajeros.	HU8 HU9	Verificación de los intervalos configurados en los puntos de operación respectivo, es ejecutada en el servidor antes de cada salida.
RES-02	El trayecto entre paradas tiene una duración de 60 minutos. Si lega exceder ese límite genera una multa equivalente a (minutos − 60) x $ 1,00 USD.	HU10	Cálculo del transcurso entre paradas con marcas del tiempo del servidor y generación de multa por retraso en el backend del sistema.
RES-03	La llegada de una unidad solo es válida cuando ocurre él envió de las coordenadas dentro de la geocerca de la parada de destino o viceversa.	HU10	Validación de la distancia mediante la fórmula de Haversine en el servidor antes de confirmar la llegada.

ID	Restricción	HU asociada	Componente que la aplica
RES-04	El sorteo diario asigna 16 unidades a la parada de Salinas y 17 unidades a la parada de Guaranda. Los números de unidades en las paradas pueden cambiar al paso del tiempo	HU6	La lógica del sorteo de todos los días, distribuye los cupos disponibles entre los puntos de operación.
RES-05	Se garantiza el descanso obligatorio de las unidades seleccionadas en grupos establecidos al inicio del ciclo anual para las operaciones diarias bajo un esquema de rotación mediante sorteo.	HU5	Calendario anual de descanso que excluye del sorteo a las unidades en reposo.

4.1.3 Historias de usuario

A partir de los análisis mediante las técnicas correspondientes se recabaron las funcionalidades relevantes que debe contar el sistema para responder a las necesidades de la compañía, estas funcionalidades se representan a través de las historias de usuarios, describen los actores involucrados, las acciones específicas que realizara el sistema. Este formato se encuentra estructurado bajo el formato característico de XP con los elementos que se muestra a continuación.

Tabla 5: *Historia de Gestión de usuarios del sistema.*

Numero:	1	**Usuario:**	Gerente
Nombre de historia: Gestión de usuarios del sistema.			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Medio
Estimación:	5 días de desarrollo		
Precondiciones:	El usuario debe estar autenticado como Gerente. Los roles deben estar predefinidos Deben estar registradas las paradas y existir al menos una unidad activa para asignar a un conductor.		

Programador responsable:	Steven Agualongo y Renan Sigcha
Descripción:	Como Gerente, quiero crear usuarios, consultarlos mediante filtros, editar, desactivar sin eliminar los datos históricos, restaurarlos y resetear su contraseña, para controlar el acceso del usuario acorde a su rol y punto de operación.
Criterio de validación:	1. Permite ingresar: nombres, apellidos, cédula, teléfono, correo (opcional), punto de operación, rol y unidad solo para conductor. 2. No se permite cédula duplicada. 3. Un nuevo usuario, restaurado o con contraseña reseteada su clave de 8 dígitos al azar y debe cambiar contraseña en el próximo inicio. 4. La edición permite modificar todos los campos salvo la cédula; al cambiar el punto se desasigna la unidad. 5. La desactivación pide confirmación, bloquea el inicio de sesión y habilita el botón "Restaurar".

Tabla 6: *Historia de Autenticación al sistema.*

Numero:	2	**Usuario:**	Gerente/tarjetera/ conductor
Nombre de historia: Autenticación al sistema.			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Bajo
Estimación	3 días de desarrollo		
Precondiciones:	El usuario debe tener una cuenta activa. El sistema debe tener configurado el método de autenticación y sus roles respectivos.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como usuario del sistema, quiero autenticarme con mis credenciales para acceder únicamente al menú y funcionalidades de mi rol.		
Criterio de validación:	1. El ingreso se realiza con el número de cédula (10 dígitos) y la contraseña. Si las credenciales son incorrectas muestra un mensaje		

	2. Un usuario inactivo recibe un mensaje “Usuario desactivado” y debe contactar al gerente, aunque las credenciales sean correctas. 3. En el primer ingreso o tras realizar un reseteo de contraseña, el sistema obliga a cambiarla antes de cualquier otra acción. 4. Cada rol ve únicamente su menú correspondiente tras autenticarse.

Tabla 7: *Historia de Gestión de unidades.*

Numero:	3	**Usuario:**	Gerente
Nombre de historia: Gestión de unidades			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Medio
Estimación:	4 días de desarrollo.		
Precondiciones:	El conductor debe estar registrado en el sistema y debe existir al menos un punto registrado.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Gerente, quiero registrar las unidades, consultarlas y cambiar su estado entre Activo, Mantenimiento y Baja. Para mantener un control exacto en el servicio operativo.		
Criterio de validación:	1. La sección Unidades lista el número interno, placa, punto, estado, celular GPS y conductor asignado, y puede buscar por punto y estado. 2. Al crear una unidad debe llenar obligatoriamente la lista la sesión de unidades y exige un número interno único; un duplicado en el mismo punto impide crear la unidad. 3. Pasar al estado de Mantenimiento una unidad asignada requiere confirmación; en ese estado la unidad no participa en sorteos ni puede asignarse a conductores. 4. Una unidad en Baja desaparece de los listados activos y puede reactivarse el estado.		

Tabla 8: *Historia de Configuración parámetro del sistema*

Numero:	4	**Usuario:**	Gerente

Nombre de historia: Configuración de parámetro del sistema			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Bajo
Estimación:	2 días de desarrollo.		
Precondiciones:	El Gerente debe haber iniciado sesión y deben existir los puntos Guaranda y Salinas.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Gerente, quiero configurar los parámetros generales del sistema y los parámetros propios de cada punto de operación de Guaranda y Salinas sin que esto afecte al otro punto.		
Criterio de validación:	1. La sección muestra los parámetros y permite editar los valores como: multa por minuto, teléfono de emergencia, días de cargo operativo y correo de notificaciones. 2. El cambio del valor de multa por minuto aplica solo a las multas generadas a partir de ese momento; las existentes no se modifican. 3. Los parámetros se editan por punto y aplican exclusivamente al punto seleccionado.		

Tabla 9: *Historia de Gestión de grupos de descansos*

Numero:	5	**Usuario:**	Tarjetera
Nombre de historia: *Gestión de grupos de descansos*			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Alto
Estimación:	7 días de desarrollo.		
Precondiciones:	Debe existir un ciclo anual activo y conductores activos con unidad asignada en el punto de la Tarjetera.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Tarjetera, quiero crear de forma automática los grupos de descanso de mi punto al inicio del ciclo anual, los días		

	correspondientes y eliminarlos para reiniciarlos, su estructura es la base del sorteo diario.
Criterio de validación:	1. Se debe crear grupos de manera aleatoria; si ya existen, el sistema pide confirmar antes de recrearlos. 2. En Guaranda se generan 5 grupos de 4 conductores y un Grupo adicional con 3 conductores fijos y una vacante rotativa 3. En Salinas se generan 7 grupos de 3 conductores, cada uno con un día de descanso. 4. Generar los días de descanso entre los grupos, en Guaranda el grupo 6 queda excluido. 5. El sistema debe permitir borrar los grupos y los días de descanso

Tabla 10: *Historia de Gestión del sorteo diario de turnos*

Numero:	6	**Usuario:**	Tarjetera
Nombre de historia: Gestión del sorteo diario de turnos			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Alto
Estimación:	8 días de desarrollo.		
Precondiciones:	Deben existir grupos de descanso del ciclo activo y conductores activos; el sorteo del día no debe haberse ejecutado aún.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como tarjetera quiero ejecutar el sorteo diario de manera automatizada en el sistema, según los grupos de descanso y el día de la semana. Se determina quienes son los punteros, descansan, operan y el orden de salida del resto.		

Criterio de validación:	1. Deben existir grupos creados y conductores activos, si existe ausencia el sistema muestra el aviso correspondiente y no genera los turnos 2. Los días del turno realizados se eliminan antes de generar el nuevo. 3. En la parada de Guaranda descansa el grupo cuyo día coincide con el actual y se eligen 2 punteros aleatorios y el resto desde la posición 3. 4. En la parada de Salinas descansa el grupo donde el día coincida con el actual se eligen 2 punteros y los respectivos turnos. 5. El resultado se muestra en tabla (posición, conductor, unidad, tipo), se incrementan los contadores de descanso y de puntero, cualquier error revierte toda la transacción.

Tabla 11: *Historia de Confirmación de presencia con geolocalización.*

Numero:	7	**Usuario:**	Tarjetera / Conductor
Nombre de historia: Confirmación de presencia con geolocalización.			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Alto
Estimación:	5 días de desarrollo.		
Precondiciones:	El conductor debe tener turno asignado para el día y su dispositivo debe contar con GPS y permisos de ubicación.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Conductor al ser designado como puntero, quiero confirmar mi presencia en el punto con un botón para capturar mi ubicación GPS y envié latitud y longitud mediante HTTPS. Como Tarjetera, quiero ver en tiempo real los punteros que ya confirmaron su presencia.		

Criterio de validación:	1. El panel del conductor (Puntero) muestra su turno del día y un botón "Confirmar Presencia". 2. La confirmación solicita permiso de geolocalización y captura latitud, longitud. 3. Si el conductor está fuera del radio de geocercas, el sistema muestra "No estás dentro del punto de trabajo" y no permite confirmar. 4. El panel de la Tarjetera muestra en tiempo real la lista "Conductores Presentes" (nombre, unidad, hora) y cada confirmación queda en auditoría.

Tabla 12: *Historia de Control de cola de conductores.*

Numero:	8	**Usuario:**	Tarjetera
Nombre de historia: Control de cola de los conductores			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Medio
Estimación:	5 días de desarrollo.		
Precondiciones:	Debe existir el sorteo del día con turnos y al menos un conductor en la cola operativa.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Tarjetera, quiero gestionar la cola de conductores en tiempo real para llamar al siguiente, bloquear a los ausentes con un motivo, liberar a los que llegan y sancionar a quienes no confirman presencia sin justificativo aprobado.		
Criterio de validación:	1. La cola muestra posición, nombre, unidad, tipo y estado de cada conductor; los de descanso no aparecen. 2. Cuando notifica al conductor cambia el estado a "Llamado" y llega teléfono del conductor y marca un indicador visual en la cola. 3. El sistema debe permitir bloquear, exige un motivo y deja al conductor visible y diferenciado en la cola, sin eliminarlo. 4. Sancionar a un conductor sin presencia ni justificativo aprobado genera una multa de $12 por inasistencia y la registra en multas.		

	5. La tarjetera puede seleccionar a un conductor que se encuentra bloqueado que ya llegó y presionar "Liberar" para que el conductor vuelva a su estado normal en la cola.

Tabla 13: *Historia de Registro de Salida del Conductor.*

Numero:	9	**Usuario:**	Conductor / Tarjetera
Nombre de historia: Registro de salida del conductor			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Medio
Estimación:	5 días de desarrollo.		
Precondiciones:	El conductor debe tener turno activo del día El dispositivo debe tener GPS activo. No debe tener una salida activa sin llegada registrada.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Tarjetera, quiero registrar las salidas desde mi panel Como Conductor, cuando la Tarjetera está ausente según los horarios configurados, quiero registrar mi propia salida capturando mi ubicación GPS.		
Criterio de validación:	1. La Tarjetera selecciona un turno activo y registra la salida; el sistema toma la última posición GPS de la unidad como origen. 2. Si la Tarjetera está presente, el sistema impide la salida del conductor con "La tarjetera está presente. No puedes registrar salida". 3. Con la Tarjetera ausente, la salida del conductor exige estar dentro del radio de geocerca y captura la ubicación GPS. 4. Cada salida queda asociada al turno, punto de origen, hora, usuario que registró y ubicación GPS de origen.		

Tabla 14: *Historia de Registro de llegada y multa por retraso.*

Numero:	10	**Usuario:**	Conductor
Nombre de historia: Registro de llegada y multa por retraso			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Alta
Estimación:	6 días de desarrollo.		
Precondiciones:	El conductor debe tener una salida activa y deben estar definidos los intervalos por punto y franja horaria. El dispositivo debe tener GPS activo. Debe existir el servicio de cálculo de multas configurado		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Conductor, quiero registrar la llegada al finalizar el viaje enviando la ubicación GPS. Como sistema, debo asignar la hora con la marca de tiempo del servidor, validar geocerca, calcular la duración del trayecto en el backend y generar la multa en caso de haber excedido el tiempo, exceptuando a punteros y a viajes con justificativo de carrera aprobado.		
Criterio de validación:	1. En el PWA el cliente envia únicamente las coordenadas como latitud y longitud mediante HTTPS y el servidor asigna la marca de tiempo de llegada (timestamp) 2. El sistema valida mediante Haversine que se encuentre dentro de la geocerca destino en el backend y registra la posición, la llegada y calcula el tiempo de trayecto en minutos. 3. Una salida que ya tiene llegada muestra "Esta salida ya tiene llegada registrada" y no permite duplicados. 4. Si la duración es menor o igual al intervalo, no se genera multa. 5. Si excede el intervalo de tiempo, la multa equivale a los minutos de retraso por el valor por minuto configurado.		

Tabla 15: *Historia de Gestión de justificativos.*

Numero:	11	**Usuario:**	Tarjetera/Conductor
Nombre de historia: Gestión de justificativos			
Prioridad de negocio:	Media	**Riesgo en desarrollo:**	Medio
Estimación:	5 días de desarrollo.		
Precondiciones:	El conductor y la Tarjetera deben haber iniciado sesión y el conductor debe tener un turno asignado.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Conductor, quiero crear un justificativo de atraso, inasistencia o sustitución sobre un turno. Como Tarjetera, quiero aprobarlos o rechazarlos y crear justificativos de tipo carrera para eximir un viaje del cálculo de multas.		
Criterio de validación:	1. El conductor crea el justificativo eligiendo turno y tipo como: sustitución, atraso o inasistencia y casos imprevistos. 2. La Tarjetera revisa los pendientes y los aprueba o rechaza desde su sección. 3. Al aprobar un justificativo de un turno bloqueado, el sistema lo desbloquea y la tarjetera puede reubicar al final de la cola y anula las multas pendientes. 4. Al rechazar, el estado pasa a "rechazado", se mantiene el bloqueo y se notifica la decisión al conductor. 5. La Tarjetera crea un justificativo de "carrera especial" sobre un turno operativo (único por turno y día) para eximirlo de la multa por trayecto.		

Tabla 16: *Historia de Monitoreo de GPS.*

Numero:	12	**Usuario:**	Gerente / Tarjetera
Nombre de historia: Monitoreo de GPS			
Prioridad de negocio:	Media	**Riesgo en desarrollo:**	Alto
Estimación:	7 días de desarrollo.		
Precondiciones:	Debe haber al menos un conductor con salida activa y deben estar configuradas las coordenadas y el radio de geocerca de los puntos.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Gerente o Tarjetera, quiero un mapa interactivo en tiempo real con la ubicación de las unidades, la geocercas de cada punto, las alertas de entrada y salida, también el historial de recorrido por salida.		
Criterio de validación:	1. La sección GPS/Mapa muestra marcadores como nombre de conductor, placa de vehiculo y velocidad, actualizándose sin recargar la página. 2. La geocercas de los puntos (Guaranda y Salinas) deben dibujarse como círculos en el mapa 3. El sistema debe generar automáticamente una alerta cada vez que un conductor entre o salga de una parada.		

Tabla 17: *Historia de Control documental de conductores y unidades.*

Numero:	13	**Usuario:**	Gerente / Tarjetera
Nombre de historia: Control documental de conductores y unidades.			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Media
Estimación:	7 días de desarrollo.		
Precondiciones:	El conductor o la unidad deben estar registrados y la Tarjetera debe haber iniciado sesión.		

Programador responsable:	Steven Agualongo y Renan Sigcha
Descripción:	Como Tarjetera, quiero registrar y validar las licencias de los conductores y las matrículas de las unidades; como Gerente, quiero supervisar el estado documental con filtros para evitar sanciones por documentos vencidos.
Criterio de validación:	1. La Tarjetera registra documentos de tipo Licencia y Matrícula con número, fecha de emisión, fecha de vencimiento y foto opcional; al guardar quedan en estado Vigente. 2. El sistema detecta los documentos por vencer, los resalta en lado de la Tarjetera y notifica al conductor asociado. 3. El Gerente supervisa el estado documental completo con filtros por tipo, estado y punto.

Tabla 18: *Historia de Gestión de ciclos anuales de operación.*

Numero:	14	**Usuario:**	Tarjetera
Nombre de historia: Gestión de ciclos anuales de operación			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Medio
Estimación:	3 días de desarrollo.		
Precondiciones:	La Tarjetera debe haber iniciado sesión y no debe existir un ciclo para el mismo año.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Tarjetera, quiero crear ciclos anuales de operación y activar uno a la vez, el ciclo activo define el período sobre el que se generan los grupos de descanso, los sorteos y las operaciones.		

Criterio de validación:	1. La creación de un ciclo exige año, fecha de inicio y fecha de fin. 2. La Tarjetera debe poder seleccionar un ciclo inactivo y presionar Activar. Si había otro ciclo activo previamente, el sistema debe desactivarlo automáticamente. 3. Solo debe existir un ciclo activo a la vez. 4. Un ciclo con turnos asociados no puede eliminarse

Tabla 19: *Historia de Reportes operativos del sistema.*

Numero:	15	**Usuario:**	Tarjetera/Gerente
Nombre de historia: Historia de Reportes operativos del sistema.			
Prioridad de negocio:	Alta	**Riesgo en desarrollo:**	Medio
Estimación:	4 días de desarrollo.		
Precondiciones:	El usuario (Tarjetera y gerente) debe haber iniciado sesión y deben existir datos en el período consultado.		
Programador responsable:	Steven Agualongo y Renan Sigcha		
Descripción:	Como Gerente o Tarjetera, quiero generar reportes operativos por área y exportarlos en PDF con resumen ejecutivo y desglose, limitados al punto de operación cuando el rol es Tarjetera.		
Criterio de validación:	1. El sistema debe permitir al Gerente acceder a reportes de unidades, conductores, multas, turnos, sanciones y GPS 2. El sistema debe permitir acceder al usuario tarjetera unidades/Conductores, sanciones/multas, turnos y justificativos. 3. El sistema debe permitir exportar en PDF los reportes respectivos.		

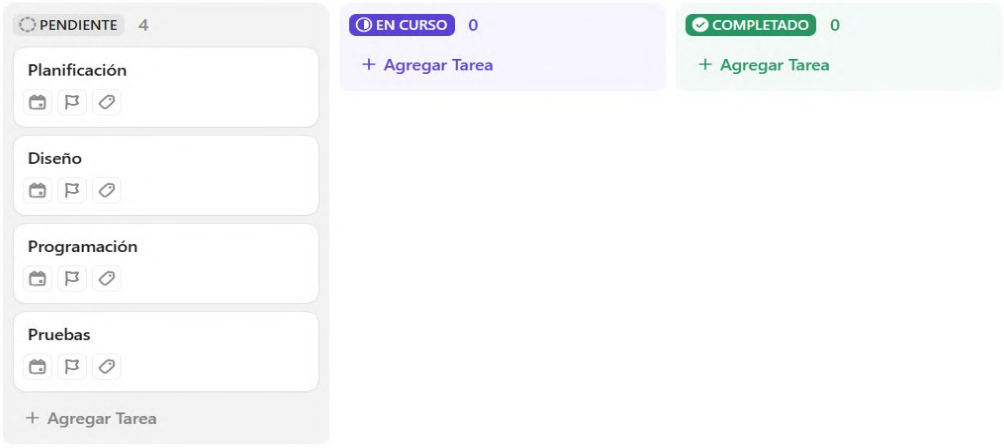
Tabla 20: *Roles de acceso al sistema*

Código de Rol	**Rol**	**Objetivo**	**Nivel de Acceso**	**Actividades**
RL-001	Gerente	Administrar y supervisar la operación integral de la compañía en ambos puntos de parada.	Alto	Gestión de usuarios, configurar parámetros operativos, supervisar la operación, generar reportes y administrar el ciclo anual de descansos.
RL-002	Tarjetera	Coordinar y controlar el flujo operativo de unidades en su punto de parada asignado.	Medio	Gestionar el sorteo diario, registrar salidas y llegadas, controlar carreras especiales y administrar multas y sanciones.
RL-003	Conductor	Operar su unidad de transporte conforme a los turnos y reglas establecidas por la compañía.	Bajo	Consultar turnos asignados, registrar su ubicación GPS, enviar justificativos y revisar el estado de sus multas.

4.1.4 Gestión de trabajo mediante Kanban

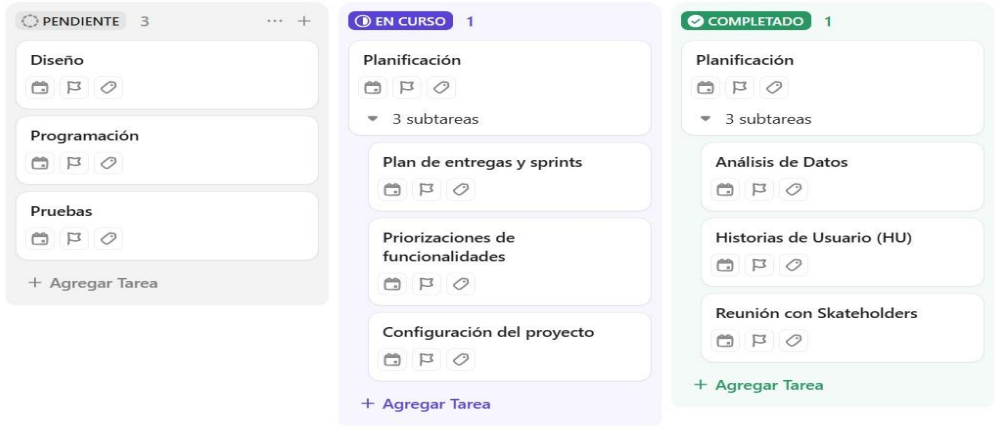
Para realizar la gestión y el seguimiento continuo de los requisitos del sistema se implementó un tablero de Kanban utilizando la herramienta ClickUp, mediante este tablero se organizaron columnas “Pendientes”, “En curso” y “completado”, permitiendo conocer el avance del proyecto y realizar el seguimiento respectivo ante posible bloqueo en el desarrollo del proyecto.

Figura 1: Tablero visual de Kanban del proyecto



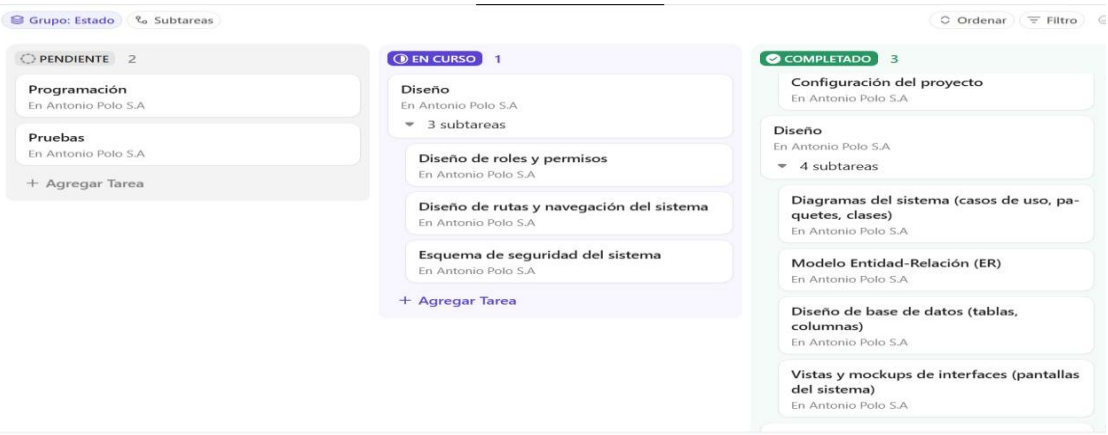
Fuente: Elaboración propia

Figura 2: Tablero visual Kanban proceso planificación.



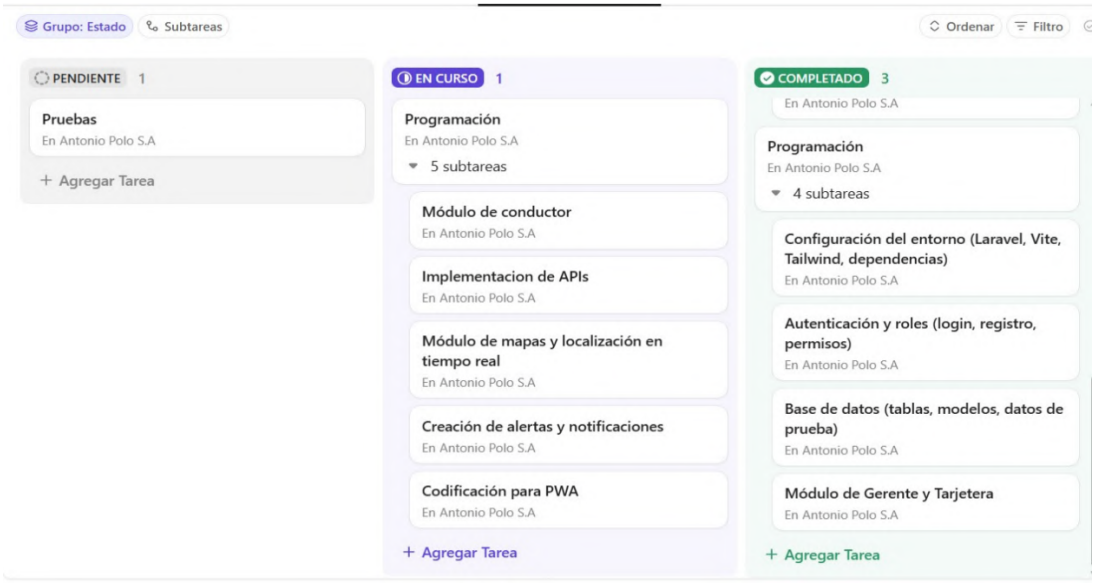
Fuente: Elaboración propia

Figura 3: Tablero visual Kanban proceso diseño



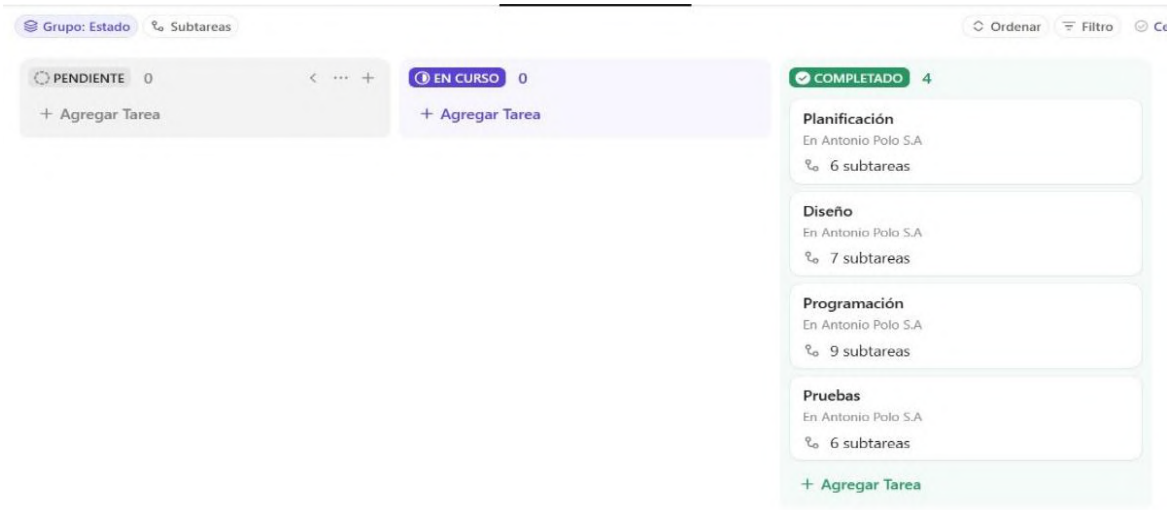
Fuente: Elaboración propia

Figura 4: Tablero visual Kanban proceso de programación



Fuente: Elaboración propia

Figura 5: Tablero final de pruebas



4.2 Diseño del sistema

En esta fase se estructura el núcleo arquitectónico del contenido del sistema transformando los requerimientos operativos de la compañía en una arquitectura robusta y escalable. Prioriza la flexibilidad, alta cohesión y el bajo acoplamiento lógico permitiendo individualizar responsabilidades y funcionalidades.

4.2.1 Vista Arquitectónica

Para desarrollar la documentación y representación formal de esta arquitectura sin dejar vacíos de información, se optó por implementar el Modelo de Vistas Arquitectónicas 4+1, propuesto originalmente por Philippe Kruchten (1995). A diferencia de los enfoques de diseño unidimensionales, este estándar internacional describe la arquitectura del software desde múltiples perspectivas, integrando las visiones y preocupaciones operativas de los usuarios, desarrolladores y administradores, lo que facilita enormemente la comprensión integral del sistema antes de su codificación.

Tabla 21: *Descripción de las vistas arquitectónicas 4+1 de kruchten*

Vista	**Descripción**	**Diagrama**	**Aplicación en el sistema**
Vista de Escenarios	Es la vista que articula a las demás vistas, aquí se representa las funcionalidades del sistema desde el lado de los actores.	Diagrama de casos de uso UML	Modela las interacciones de los tres actores (Gerente, Tarjetera y Conductor) con los casos de uso principales.
Vista Lógica	Describe la estructura estática del sistema: las clases del dominio, atributos, métodos y relaciones de multiplicidad.	Diagrama de clases UML	Representa las entidades del negocio y sus asociaciones.
Vista de Procesos	Representa el comportamiento dinámico del sistema y el flujo de operaciones en tiempo de	Diagrama de actividad	Clasifica las responsabilidades entre Usuario, Navegador/PWA,

	ejecución, mostrando cómo colaboran los componentes ante un evento.		servidor Laravel y MySQL para procesos críticos como la detección de llegada por geocerca.
Vista de Desarrollo	Muestra la organización del software desde la perspectiva del programador	Diagrama de paquetes	Refleja la estructura MVC de Laravel y el flujo de una petición como rutas controladores y modelos, esto facilita la división del trabajo entre los dos desarrolladores.
Vista Física	Describe la disposición del sistema sobre la infraestructura realizando una representación por nodos.	Diagrama de despliegue	Comprende los dispositivos del cliente (celulares con la PWA) Tablet o laptop (Tarjetera, Gerente) comunicando vía HTTP/HTTPS con el servidor del sistema y base de datos MySQL realizando la comunicación externa con Google Maps.

Fuente: Elaborado por Renan Sigcha y Alexander Agualongo

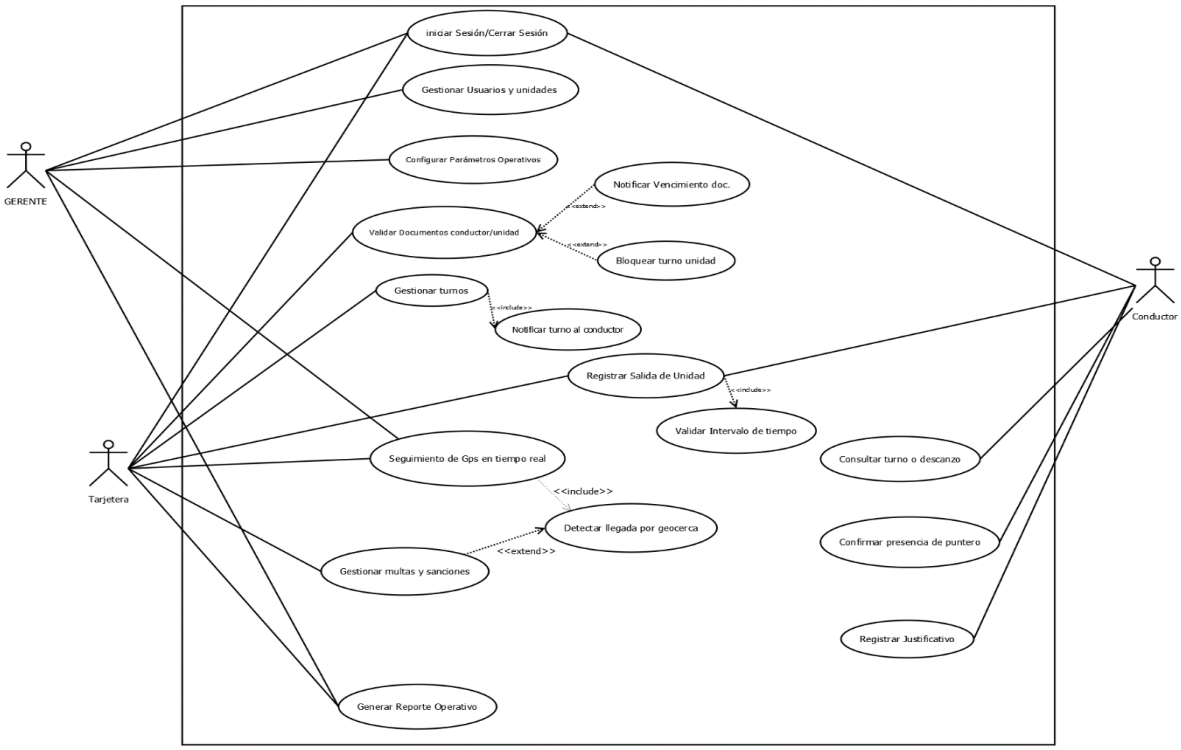
4.2.2 Vistas arquitectónicas 4+1

4.2.2.1 Vista de escenarios

Constituye un modelo arquitectónico que valida una estructura técnica del software que cumpla con los requerimientos recopilados en la fase de la planificación. Es representado mediante diagrama casos de uso en UML, el cual modela de manera dinámica el comportamiento del sistema desde la perspectiva del usuario final. Para el sistema el diagrama de casos de uso muestra las interacciones de los tres actores principales:

Gerente: Encargado de configurar las reglas, reportes y control de los usuarios
Tarjetera: responsable del control operativo, cobro de multas y sanciones en caso de existir, sorteo de turnos y validación de documentos.
Conductor: Interactúa mediante el PWA para el envío de coordenadas GPS y visualizar la lista de turnos y realizar las justificaciones respectivas.

Figura 6: Diagrama de casos de uso (escenarios)

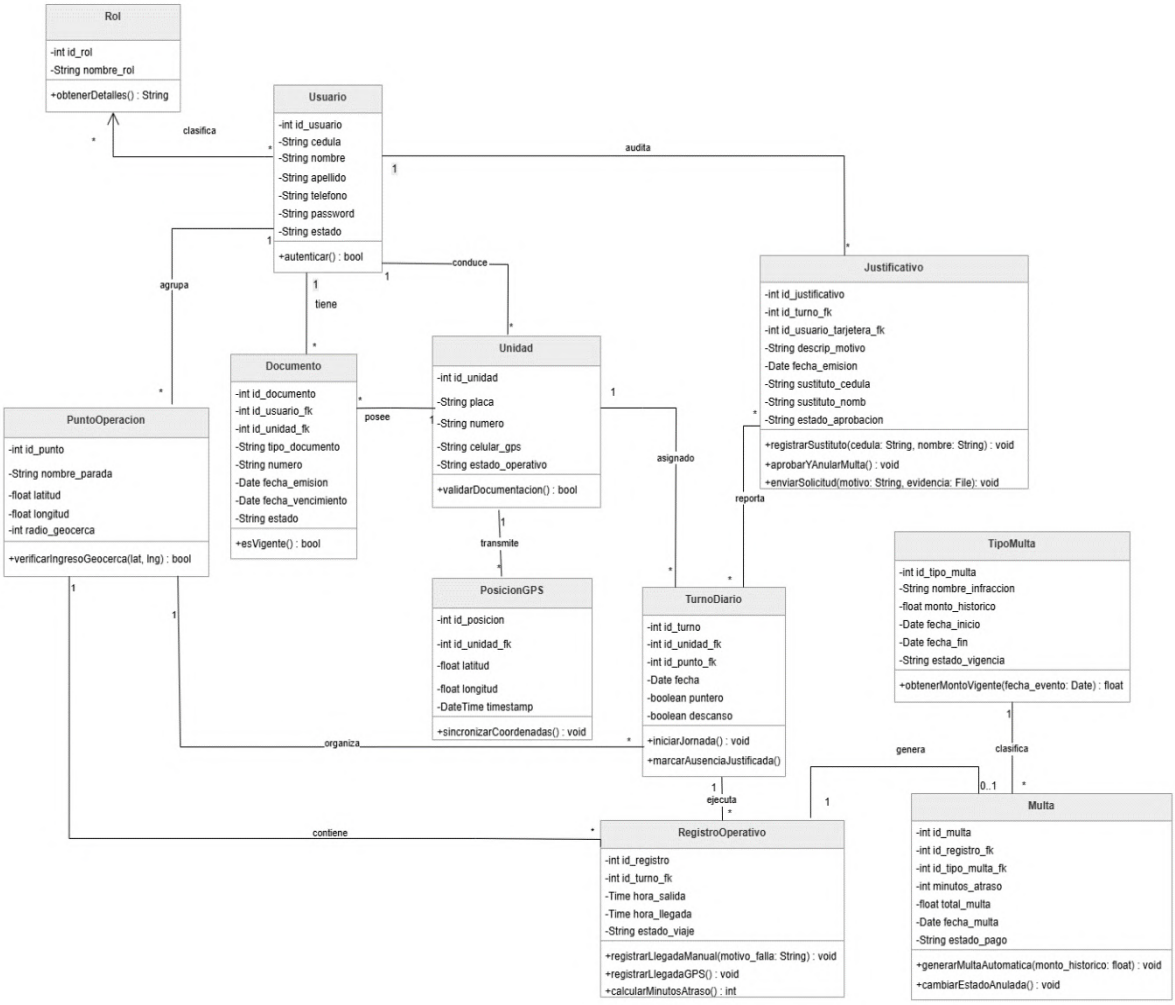


Nota: Actores involucrados del sistema

4.2.2.2 Vista lógica del sistema

La vista lógica tiene como propósito abstraer y modelar la estructura estática de la información, así como las interacciones entre los componentes conceptuales del sistema. Se representa mediante diagrama de clases en notación UML el cual resulta muy relevante para traducir las reglas operativas de la compañía bien definidas. Establecer los atributos, métodos o comportamientos y las relaciones de multiplicidad entre las entidades correspondientes, garantiza un diseño con alta cohesión, esto asegura que los procesos como sincronización de coordenadas y cálculo automatizado por retraso se mantenga integro en las bases de datos MySQL.

Figura 7: Diagrama de clases (Vista logica)

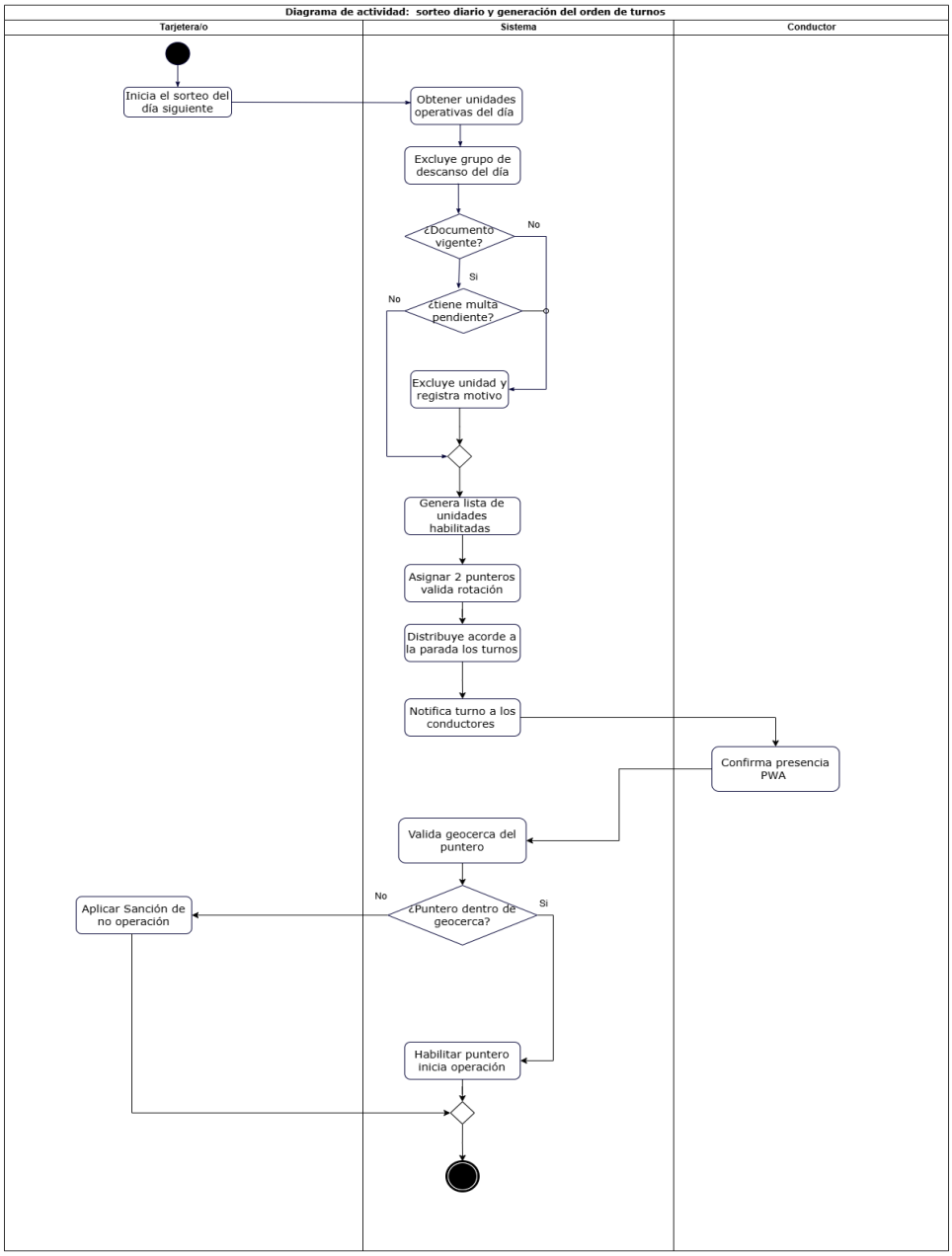


Nota: Estructura de diagrama de clases de la base de datos del sistema

4.2.2.3 Vistas de procesos

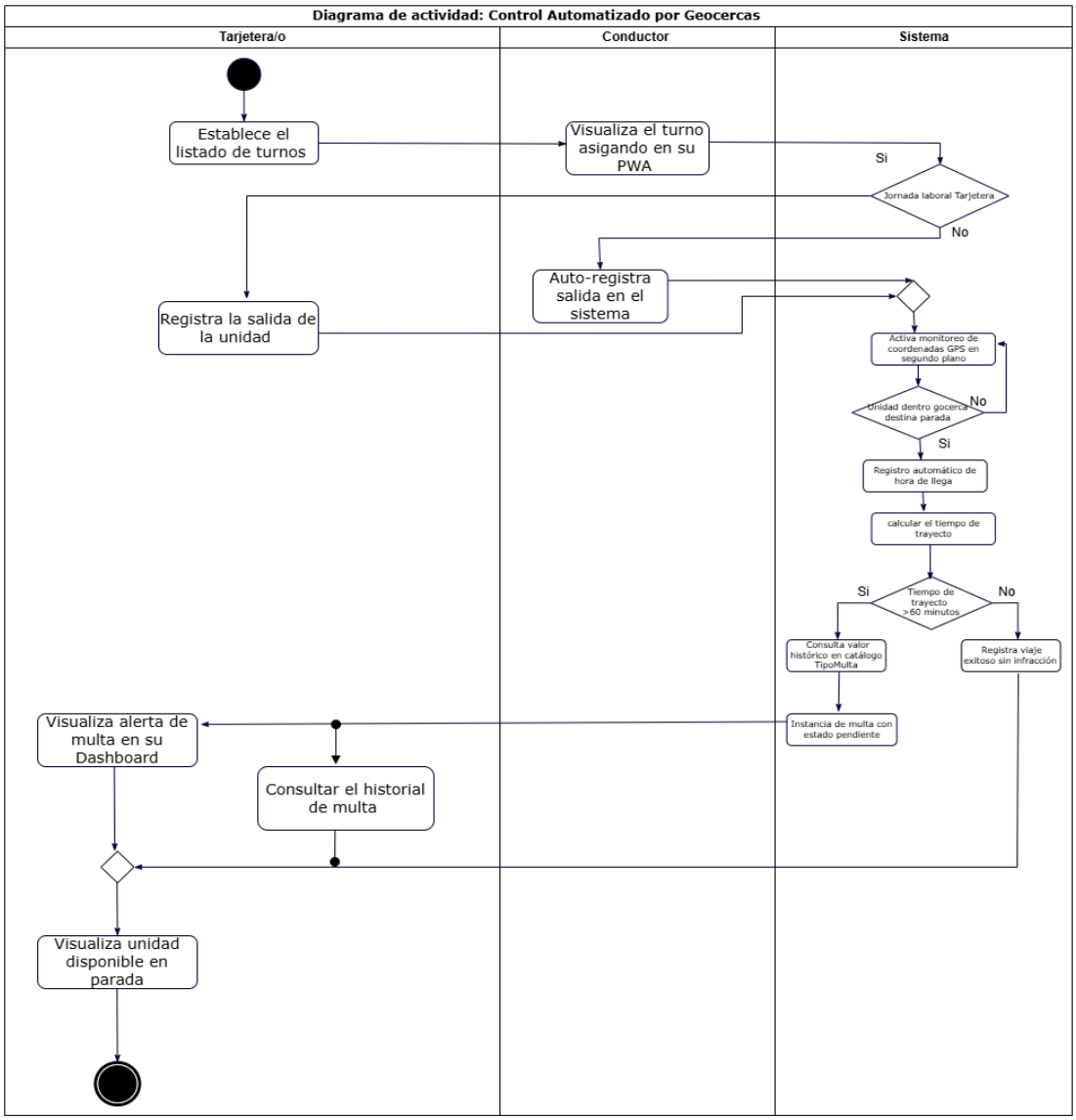
La vista de procesos define la estructura dinámica, con el propósito de detallar el comportamiento del sistema mediante flujo de operación de ejecución. Se representa mediante el diagrama de actividades en notación UML, la misma que ilustra de manera consecutiva como aportan los diversos componentes ante un evento, distribuyendo las responsabilidades entre usuarios y el sistema.

Figura 8: Diagrama de actividad, Sorteo diario y generación de turnos



Nota. Flujo de generación de turno y confirmación de los punteros al inicio de la jornada

Figura 9: Diagrama de actividad, control automatizado por Geocercas



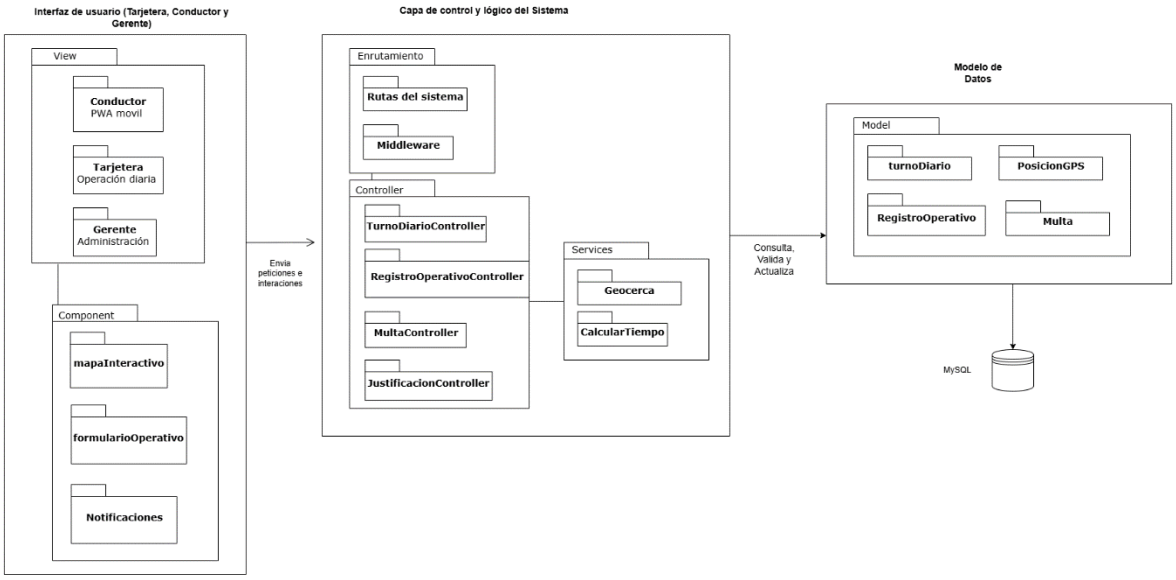
Nota. Flujo de control de llegada de las unidades a las paradas

4.2.2.4 Vista de desarrollo

La vista de desarrollo representa la organización de los módulos de software en la programación desde el lado del desarrollador, muestra cómo se agrupa las clases y componentes del sistema mismo que facilita la construcción y el mantenimiento. El diagrama se estructuro bajo el ecosistema MVC e ilustra la encapsulación del código en la capa de presentación que es el interfaz del usuario, se agrupan las vistas de los actores (PWA del conductor, interfaz de Tarjetera y Gerente) junto a los componentes dinámicos de Blade y Alpine.js, tales como el mapa interactivo y las notificaciones.

En el control lógico, los paquetes de enrutamiento canalizan las peticiones hacia los controladores específicos. También se implementó el paquete de servicios que calcula la intercesión de la geocercas y el procesamiento del tiempo transcurrido. Finalmente, el modelo de datos agrupa las clases del ORM Eloquent encargadas de la persistencia.

Figura 10: Vista de desarrollo (diagrama de paquetes)

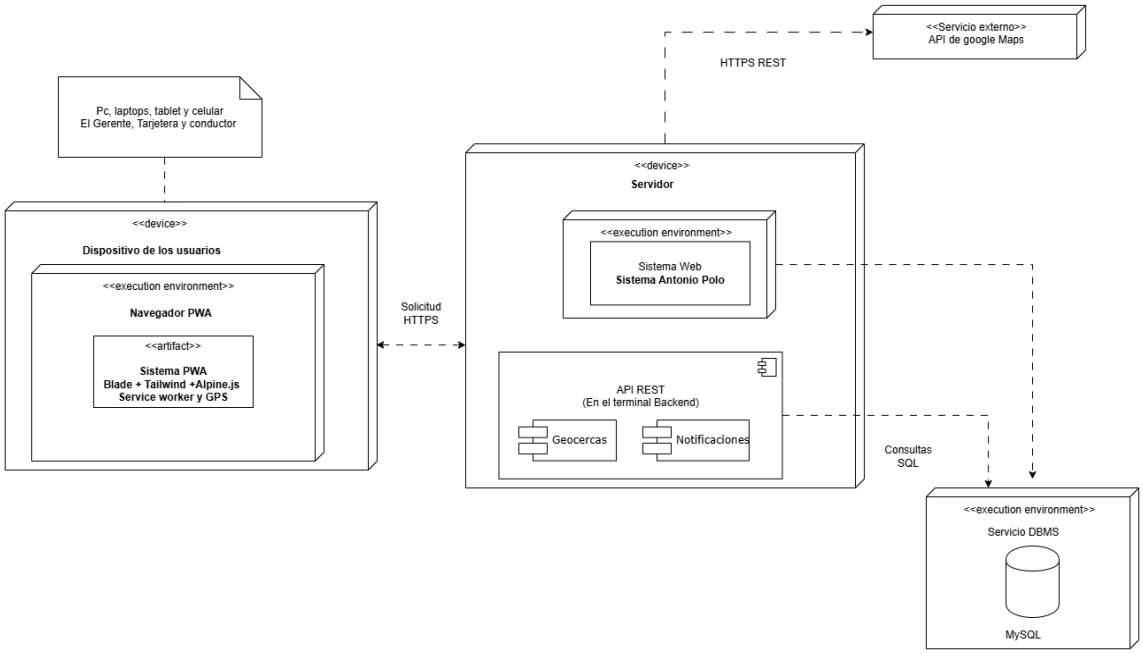


4.2.2.5 Vista física

Ilustra la distribución de los nodos de procesamiento, los canales de comunicación y los entornos de ejecución mismo que garantiza la disponibilidad, conectividad y rendimiento del sistema, se representa mediante el diagrama de despliegue. Dentro de la representación de la vista se detalla la interacción concurrente entre los nodos principales mediante el protocolo seguro HTTPS. En los nodos del cliente, se ilustran los dispositivos de los usuarios finales: computadoras de escritorio y tablets utilizadas por la Tarjetera y el Gerente, y dispositivos moviles portados por los conductores. En el caso de los conductores, sus dispositivos ejecutan la Aplicación Web Progresiva PWA que consume la API nativa del navegador para capturar las coordenadas exactas directamente desde el hardware GPS interno del celular.

Estos clientes interactúan directamente con el nodo central del Servidor y que a su vez se comunica externamente con la API de Google Maps para el renderizado. Finalmente, se describe el nodo de persistencia que aloja el motor relacional MySQL.

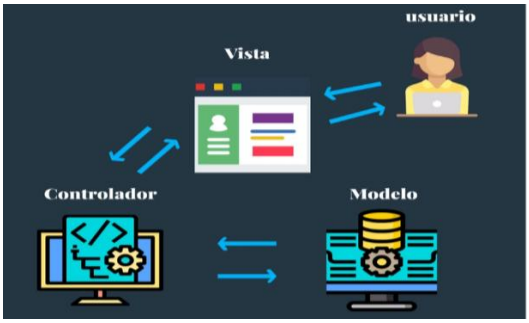
Figura 11: Diagrama de despliegue



4.2.3 Modelo arquitectónico

Se implemento el patrón arquitectónico MVC (Modelo-Vista-Controlado), principalmente la justificación de nuestra elección a este patrón es porque aparte de ser el patrón más conocido es muy bueno para trabajar con aplicaciones web ya que su estructura está bien diseñada, comprobable y de mantenimiento sencillo, porque podemos separar en partes la estructura de un sistema.

Figura 12: Modelo-Vista-Controlador

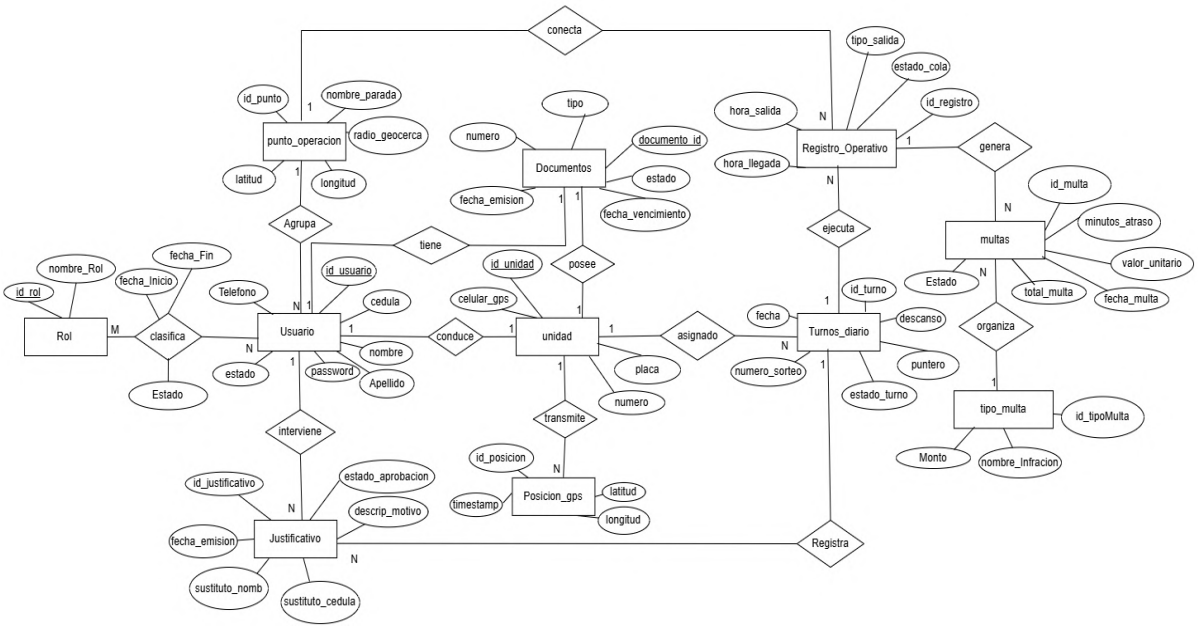


Nota: Estructura Patrón Arquitectónico Modelo-Vista-Controlador. (Guamán. V, 2021)

4.2.4 Modelo

El modelo entidad-relación es un instrumento que nos indica qué personas, objetos o conceptos se van relacionando entre ellos. Nos sirve para exponer la información de una manera más organizada de una base de datos.

Figura 13: Diagrama de Entidad-Relación



Autor: Elaborado por Renan Sigcha y Alexander Agualongo

Descripción de Diagrama Entidad - Relación.

La entidad rol almacena a los perfiles del sistema y se define tres actores: Gerente, Tarjetera y Conductor. En la entidad usuario se guarda los datos respectivos de los usuarios acorde al rol. Un punto de operación agrupa varios usuarios asignados en la parada con la cardinalidad de 1: N y esta se vincula con los registros operativos generados en el punto respectivo.

Un usuario conduce una unidad y también tiene su propia documentación al igual cada unidad posee su propia documentación respectiva.

Cada unidad recibe la asignación de múltiples turnos diarios a lo largo del tiempo con 1: N, como también transmite múltiples registros de posición GPS (1: N).

Un turno diario lleva la ejecución de uno o varios registros operativos (1: N) y cada registro que sobrepase el tiempo límite genera una multa y toda multa se clasifica bajo

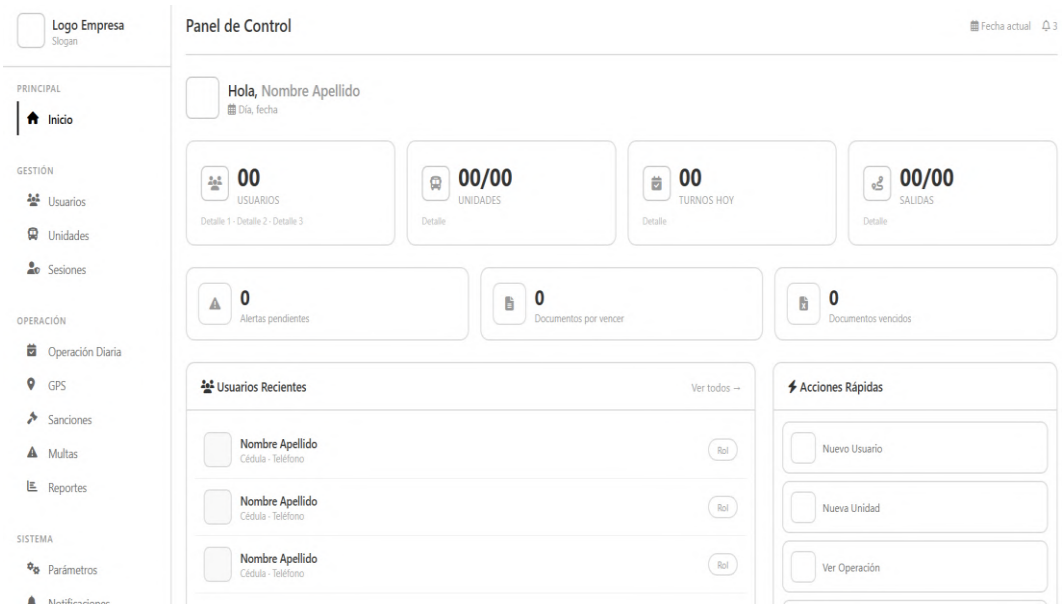
un tipo multa, si se llega a modificar los valores de la multa en un periodo especifico no alterara el historial de reportes.

Finalmente, un usuario interviene en la emisión de justificativos con la cardinalidad de 1: N y estos justificativos queda registrado sobre los turnos diarios correspondiente.

4.2.5 Vista

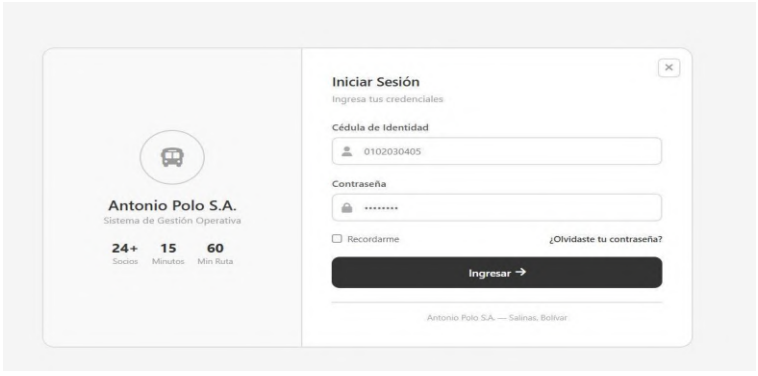
Este modelo es de especificación, donde muestra las rutas de una aplicación web, cómo se va a presentar el contenido con su respectiva información teniendo en cuenta que usa pantallas donde el usuario recorre donde se cumpla los objetivos dentro de su aplicación diseñada.

Figura 14: Vista de la Página de Inicio.



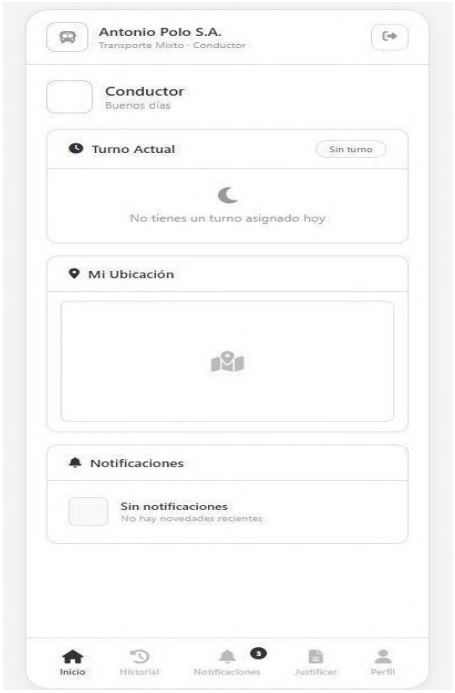
Nota: Elaboración Propia. Renan Sigcha y Alexander Agualongo

Figura 15: Vista de Inicio de Sesión



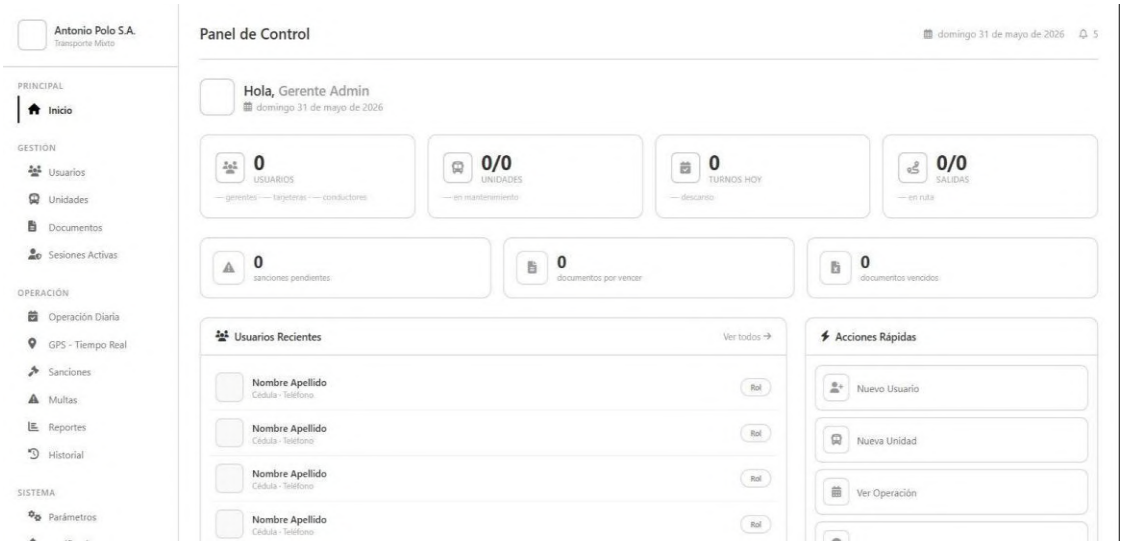
Nota: Elaboración Propia. Renan Sigcha y Alexander Agualongo

Figura 16: Vista de Conductor



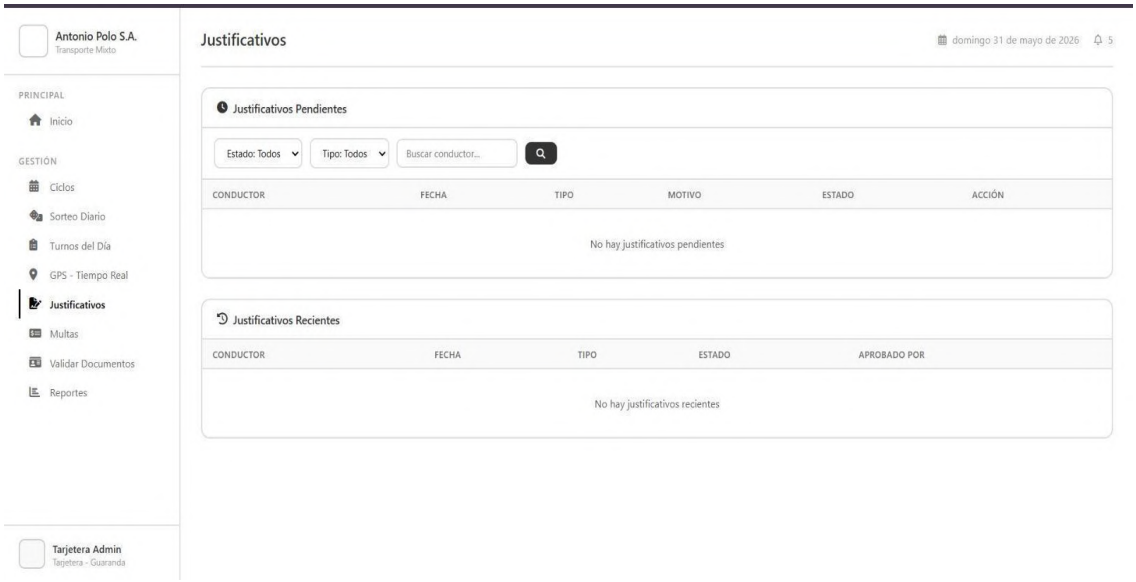
Nota: Elaboración Propia. Renan Sigcha y Alexander Agualongo

Figura 17: Vista de Gerente



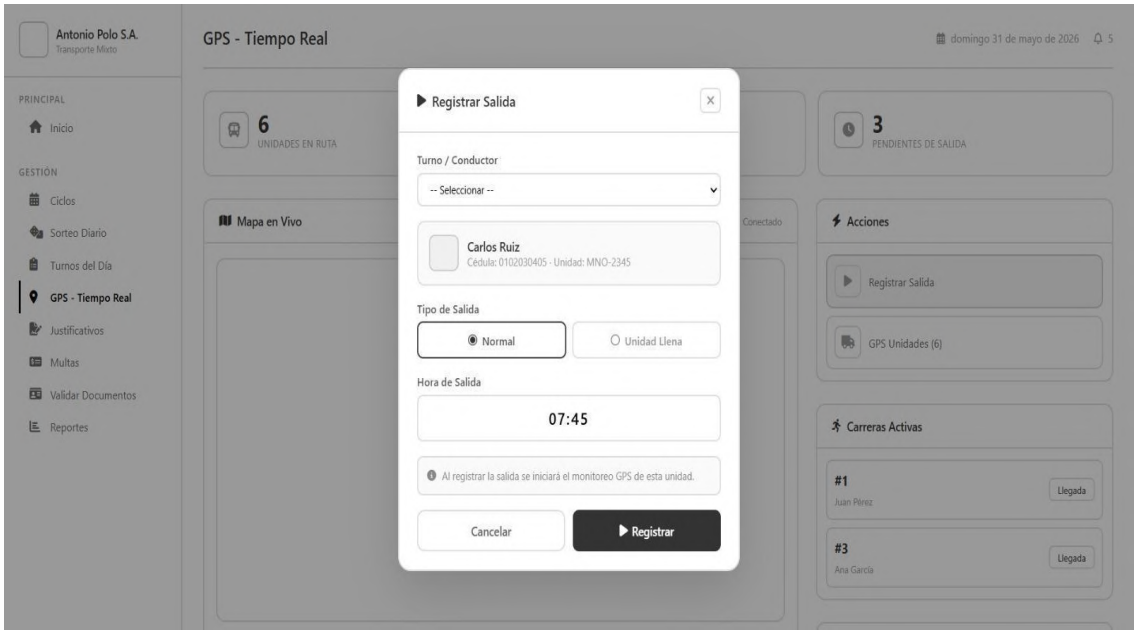
Nota: Elaboración Propia. Renan Sigcha y Alexander Agualongo

Figura 18: Vista de Justificativo



Nota: Elaboración Propia. Renan Sigcha y Alexander Agualongo

Figura 19: Vista de Registro - Salida



Nota: Elaboración Propia. Renan Sigcha y Alexander Agualongo

4.2.6 Controlador

Se procesan todas las peticiones que haga un usuario al usar un sistema o aplicación, procesa la lógica para interactuar con este Modelo para ingresar a los datos y actualizarlos junto con la selección de la Vista acorde con lo que le corresponde para mostrar su respuesta.

Tabla 22: *Algoritmo de Gestión de Unidades*

Nombre	Código Fuente	Detalle

CRUD de Unidades en el sistema	public function store(Request $request)
	{
	$validated = $request->validate([
	'placa' =>
	'required|string|max:20|unique:unidades,placa',
	'numero_interno' =>
	'required|string|max:10|unique:unidades,numero_interno',
	'celular_gps' => 'nullable|string|max:20',
	'punto_id' => 'required|exists:puntos,punto_id',
	]);
	
	$validated['estado'] = 'activo';
	$unidad = Unidad::create($validated);
	
	HistorialCambio::registrar('unidades', $unidad->unidad_id, 'crear',
	Auth::user()->usuario_id, null, null,
	"Unidad {$unidad->placa} - {$unidad->numero_interno}", $request->ip());
	
	return redirect()->route('gerente.unidades.index')
	->with('success', 'Unidad creada exitosamente.');
	}
		Valida campos únicos (placa, nº interno), crea el modelo, registra en historial, redirige.

Nota: Elaboración propia. Steven Agualongo y Renan Sigcha		

Tabla 23: *Algoritmo de Confirmación de Presencia con Geocerca.*

Nombre	Código Fuente	Detalle
	public function confirmarPresencia(Request $request) { $turno = Turno::where('usuario_id', auth()->user()->usuario_id)	

Confirmación del conductor puntero	->whereDate('fecha', today())->with('punto')->first();
	
	$validated = $request->validate([
	'latitud' => 'required|numeric|between:-90,90',
	'longitud' => 'required|numeric|between:-180,180',
	]);
	
	$distancia = $this->calcularDistancia(
	$validated['latitud'], $validated['longitud'],
	$turno->punto->latitud, $turno->punto->longitud
	);
	
	if ($distancia > ($turno->punto->radio_geofence_m ?? 50))
	return response()->json(['error' => 'No estás en el punto'], 403);
	
	if (Documento::where('usuario_id', $usuario->usuario_id)
	->whereDate('fecha_vencimiento', '<', now())->exists())
	return response()->json(['error' => 'Documentos vencidos'], 403);
	
	$turno->update([
	'presencia_confirmada' => true, 'presencia_at' => now(),
	'presencia_lat' => $validated['latitud'],
	'presencia_lng' => $validated['longitud'],
	]);
	}
		Verifica que el conductor esté físicamente dentro del radio geocerca del punto (Haversine), bloquea si tiene documentos vencidos, y marca presencia.

Nota: Elaboración propia. Steven Agualongo y Renan Sigcha		

Tabla 24: *Algoritmo de Confirmación de Bloqueo de Turno*

Nombre	Código Fuente	Detalle

	public function bloquearTurno(Request $request, $turnoId)
	{
	$turno = Turno::findOrFail($turnoId);
	
	if ($turno->bloqueado) {
	return redirect()->back()	

Bloqueo de turnos con la lógica del negocio.	->withErrors(['error' => 'Este turno ya está bloqueado.']);
	}
	
	if ($turno->es_puntero) {
	return redirect()->back()
	->withErrors(['error' => 'No se puede bloquear un puntero.']);
	}
	
	$validated = $request->validate([
	'motivo' => 'required|string|max:255',
	]);
	
	$turno->bloqueado = 1;
	$turno->motivo_bloqueo = $validated['motivo'];
	$turno->bloqueado_por = Auth::user()->usuario_id;
	$turno->bloqueado_en = now();
	$turno->save();
	
	HistorialCambio::registrar('turnos_diarios', $turno->turno_id,
	'modificar', Auth::user()->usuario_id,
	'bloqueado', 0, 1, $request->ip());
	
	return redirect()->back()
	->with('success', 'Turno bloqueado. El conductor no operará este día.');
	}
		Verifica que el conductor no sea de tipo "puntero", procede con el bloqueo de turno según las reglas de negocio. Guarda y lo registra en auditoría.

Nota: Elaboración propia. Steven Agualongo y Renan Sigcha		

4.3 Programación

En concordancia con los principios de la metodología Extreme Programming (XP), la fase de codificación se ejecutó priorizando la simplicidad del diseño, la refactorización continua y la implementación de un código limpio. El backend se estructuró sobre el framework Laravel bajo el patrón arquitectónico Modelo-Vista-Controlador, lo que permitió organizar el enrutamiento, la lógica de negocio y la persistencia de datos de manera modular.

4.3.1 Patrones de Diseño GoF

Para llevar a cabo la fase de codificación en el sistema web de la Compañía Antoño Polo S.A., se integraron los patrones de diseño (Gof). La aplicación de estos estándares tiende a ser flexible en el mantenimiento y extender a futuro si llegara a ser necesario. Permite resolver un problema recurrente y garantizar el desarrollo de un código fuente más limpio y ordenado. A continuación, se establece la descripción correspondiente.

Patrón State

El patrón State permite que un objeto altere su comportamiento cuando cambia su estado interno, delegando ese comportamiento en clases que representan cada estado. Permite controlar el ciclo de vida completo del turno del conductor, cada estado del turno tiene su propia clase, establece las acciones permitidas y el movimiento a sus estados correspondientes.

```
// 1. Interfaz que define lo que cada estado debe saber hacer
interface TurnoStateInterface {
    public function confirmarPresencia(TurnoDiario $turno): void;
    public function registrarSalida(TurnoDiario $turno): void;
    public function registrarLlegada(TurnoDiario $turno): void;
    public function bloquear(TurnoDiario $turno): void;
    public function sancionar(TurnoDiario $turno): void;
    public function puedeTransicionar(string $nuevoEstado): bool;
    public function getEstado(): string;
    public function getLabel(): string;
}
// 2. Clase base que evita repetir código en cada estado
abstract class BaseTurnoState implements TurnoStateInterface {
    public function confirmarPresencia(TurnoDiario $turno): void {
        throw new InvalidArgumentException("No se puede en estado {$this->getEstado()}");
    }
    public function registrarSalida(TurnoDiario $turno): void {
        throw new InvalidArgumentException("No se puede en estado {$this->getEstado()}");
    }
    public function registrarLlegada(TurnoDiario $turno): void {
        throw new InvalidArgumentException("No se puede en estado {$this->getEstado()}");
    }
    public function bloquear(TurnoDiario $turno): void {
        throw new InvalidArgumentException("No se puede en estado {$this->getEstado()}");
    }
    public function sancionar(TurnoDiario $turno): void {
        throw new InvalidArgumentException("No se puede en estado {$this->getEstado()}");
    }
```

```
}
// 3. Ejemplo de estado concreto: Pendiente

class PendienteState extends BaseTurnoState {
   public function confirmarPresencia(TurnoDiario $turno): void {
      $turno->update(['estado' => 'confirmado', 'presencia_confirmada' => true]);
   }
   public function bloquear(TurnoDiario $turno): void {
      $turno->update(['estado' => 'bloqueado']);
   }
   public function puedeTransicionar(string $nuevoEstado): bool {
      return in_array($nuevoEstado, ['confirmado', 'bloqueado', 'sancionado']);
   }
   public function getEstado(): string { return 'pendiente'; }
   public function getLabel(): string { return 'Pendiente'; }
}
// 4. Clase que recibe un turno y según su estado actual permite o niega acciones

class TurnoContext {
   protected TurnoStateInterface $state;
   protected TurnoDiario $turno;
   public function __construct(TurnoDiario $turno) {
      $this->turno = $turno;
      $this->state = $this->resolveState($turno->estado);
   }
   protected function resolveState(string $estado): TurnoStateInterface {
      return match ($estado) {
         'pendiente' => new PendienteState(),
         'confirmado' => new ConfirmadoState(),
         'en_curso' => new EnCursoState(),
         'completado' => new CompletadoState(),
         'bloqueado' => new BloqueadoState(),
         'descanso' => new DescansoState(),
         'sancionado' => new SancionadoState(),
         default => throw new InvalidArgumentException("Estado desconocido: {$estado}"),
      };
   }
   public function setState(string $estado): void {
      $nuevoState = $this->resolveState($estado);
      if (!$this->state->puedeTransicionar($estado)) {
         throw new InvalidArgumentException(
            "No se puede pasar de {$this->state->getEstado()} a {$estado}"
         );
      }
      $this->turno->update(['estado' => $estado]);
      $this->state = $nuevoState;
   }
   public function confirmarPresencia(): void { $this->state->confirmarPresencia($this->turno);
}
   public function registrarSalida(): void { $this->state->registrarSalida($this->turno); }
```

```
    public function registrarLlegada(): void { $this->state->registrarLlegada($this->turno); }
    public function bloquear(): void { $this->state->bloquear($this->turno); }
    public function sancionar(): void { $this->state->sancionar($this->turno); }
    public function getEstado(): string { return $this->state->getEstado(); }
    public function getLabel(): string { return $this->state->getLabel(); }
}
```

El turno es la parte principal del sistema, toda acción que realiza un conductor como confirmar su presencia cuando es puntero, salir, llegar, ser bloqueado o sancionado netamente dependen del estado actual y su nivel de integración es alta. Cuando se necesita bloquear un turno. El sistema automáticamente busca en qué estado está el turno y verifica si puede bloquear o no. Esto evita tener if-else enormes revisando estados en cada parte del código.

Patrón Command.

Consiste en transformar una petición en un objeto autónomo, parametrizando y registrando ejecuciones sin estar al tanto de la operación. En el sistema este patrón tiene la función de convertir la operación del sistema (registrar llegada o generar multas) en un objeto independiente, ejecuta la acción y de manera automática guarda sin que el desarrollador lo recuerde realizar manualmente.

```
// 1. Interfaz que define lo que debe tener cualquier comando
interface CommandInterface {
    public function execute(): mixed;
    public function getDescripcion(): string;
    public function getTipoAccion(): string;
    public function getTabla(): string;
    public function getRegistroId(): ?int;
    public function getDetalle(): string;
}
// 2. Clase base que ejecuta y guardar el historial
abstract class BaseCommand implements CommandInterface {
    protected ?int $usuarioId = null;
    public function __construct(?int $usuarioId = null) {
        $this->usuarioId = $usuarioId ?? auth()->id();
    }
    public function ejecutarYAuditar(): mixed {
        $resultado = $this->execute();
        HistorialCambio::create([
            'tabla' => $this->getTabla(),
            'registro_id' => $this->getRegistroId(),
```

```
            'accion' => $this->getTipoAccion(),
            'detalle' => $this->getDetalle(),
            'usuario_id' => $this->usuarioId,
        ]);
        return $resultado;
    }
    abstract public function execute(): mixed;
    abstract public function getDescripcion(): string;
    abstract public function getTipoAccion(): string;
    abstract public function getTabla(): string;
    abstract public function getRegistroId(): ?int;
    abstract public function getDetalle(): string;
}
```

// 3. Ejemplo de comando, Aplicar Multa

```
class AplicarMultaCommand extends BaseCommand {
    protected int $usuarioIdMulta;
    protected float $monto;
    protected string $tipo;
    protected string $motivo;
    protected ?int $turnoId;
    protected ?Multa $multa;
    public function __construct(int $usuarioIdMulta, float $monto, string $tipo,
                                string $motivo, ?int $turnoId = null,
                                ?int $usuarioAuditorId = null) {
        parent::__construct($usuarioAuditorId);
        $this->usuarioIdMulta = $usuarioIdMulta;
        $this->monto = $monto;
        $this->tipo = $tipo;
        $this->motivo = $motivo;
        $this->turnoId = $turnoId;
        $this->multa = null;
    }
    public function execute(): mixed {
        $this->multa = Multa::create([
            'usuario_id' => $this->usuarioIdMulta,
            'monto' => $this->monto,
            'tipo' => $this->tipo,
            'motivo' => $this->motivo,
            'turno_id' => $this->turnoId,
            'fecha_multa' => now(),
            'estado' => 'pendiente',
        ]);
        return $this->multa;
    }
    public function getDescripcion(): string {
        return "Aplicar multa de \${$this->monto} a usuario #{$this->usuarioIdMulta}";
    }
    public function getTipoAccion(): string { return 'aplicar_multa'; }
    public function getTabla(): string { return 'multas'; }
```

```
    public function getRegistroId(): ?int { return $this->multa?->multa_id; }
    public function getDetalle(): string {
        return "Multa {$this->tipo}: \${$this->monto} - {$this->motivo}";
    }
}
```

Es muy importante esta integración, porque garantiza que las acciones criticas queden registradas. En un sistema donde se maneja las multas, registro de turnos y sanciones es importante mantener la trazabilidad. Al momento de aplicar una multa con la integración de este patrón mediante (new AplicarMultaCommand(...))->ejecutarYAuditar() registra automáticamente en su historial.

Patrón Observer

Aquí se define el vínculo entre objetos, permitiendo que el sujeto principal convierta un estado o lleve a cabo una acción, notifique automáticamente sin que el objeto sepa quién fue ni qué hace. En el sistema cuando ocurre una acción relevante como: asignación de turno, bloqueo, multa, salida y llegada, notifica a los usuarios correspondientes (conductor, gerente y tarjetera) sin necesidad que un usuario conozca quien recibe a quien.

// 1. Interfaz del que avisa (sujeto)

```
interface SubjectInterface {
    public function attach(ObserverInterface $observer, string $evento): void;
    public function detach(ObserverInterface $observer, string $evento): void;
    public function notify(string $evento, array $datos): void;
}
```

// 2. Interfaz del que recibe el aviso (observador)

```
interface ObserverInterface {
    public function actualizar(string $evento, array $datos): void;
}
```

// 3. Ejemplo de observador: avisa al conductor cuando le pasa algo

```
class NotificadorConductorObserver implements ObserverInterface {
    public function actualizar(string $evento, array $datos): void {
        $conductorId = $datos['conductor_id'] ?? null;
        if (!$conductorId) return;
        $mensajes = [
            'turno_asignado' => 'Se te ha asignado un nuevo turno.',
            'salida_registrada' => 'Tu salida ha sido registrada.',
            'llegada_registrada' => 'Tu llegada ha sido registrada.',
            'multa_aplicada' => 'Se te ha aplicado una multa.',
            'turno_bloqueado' => 'Tu turno ha sido bloqueado.',
```

```
        ];
        $mensaje = $mensajes[$evento] ?? "Evento: {$evento}";
        Notificacione::create([
            'usuario_id' => $conductorId,
            'titulo' => $this->tituloPara($evento),
            'mensaje' => $mensaje,
            'tipo' => $evento,
            'fecha' => now(),
            'hora' => now()->format('H:i:s'),
            'leido' => false,
        ]);
    }
    private function tituloPara(string $evento): string {
        $titulos = [
            'turno_asignado' => 'Turno Asignado',
            'salida_registrada' => 'Salida Registrada',
            'llegada_registrada' => 'Llegada Registrada',
            'multa_aplicada' => 'Multa Aplicada',
            'turno_bloqueado' => 'Turno Bloqueado',
        ];
        return $titulos[$evento] ?? 'Notificacion';
    }
}
// 4. Sujeto concreto: junta todos los observadores y los avisa cuando algo pasa

class TurnoSubject implements SubjectInterface {
    protected array $observadores = [];
    public function attach(ObserverInterface $observer, string $evento): void {
        $this->observadores[$evento][] = $observer;
    }
    public function detach(ObserverInterface $observer, string $evento): void {
        $this->observadores[$evento] = array_filter(
            $this->observadores[$evento] ?? [],
            fn($o) => $o !== $observer
        );
    }
    public function notify(string $evento, array $datos): void {
        foreach ($this->observadores[$evento] ?? [] as $observer) {
            $observer->actualizar($evento, $datos);
        }
    }
    public static function crearConObservadoresPorDefecto(): self {
        $subject = new self();
        $subject->attach(new NotificadorConductorObserver(), 'turno_asignado');
        $subject->attach(new NotificadorConductorObserver(), 'salida_registrada');
        $subject->attach(new NotificadorConductorObserver(), 'llegada_registrada');
        $subject->attach(new NotificadorConductorObserver(), 'multa_aplicada');
        $subject->attach(new NotificadorConductorObserver(), 'turno_bloqueado');
        $subject->attach(new NotificadorGerenteObserver(), 'multa_aplicada');
        $subject->attach(new NotificadorGerenteObserver(), 'turno_bloqueado');
```

```
        $subject->attach(new NotificadorGerenteObserver(), 'sancion_aplicada');
        $subject->attach(new NotificadorTarjeteraObserver(), 'salida_registrada');
        $subject->attach(new NotificadorTarjeteraObserver(), 'llegada_registrada');
        $subject->attach(new NotificadorTarjeteraObserver(), 'turno_asignado');
        $subject->attach(new RegistradorHistorialObserver(), 'turno_asignado');
        $subject->attach(new RegistradorHistorialObserver(), 'salida_registrada');
        $subject->attach(new RegistradorHistorialObserver(), 'llegada_registrada');
        $subject->attach(new RegistradorHistorialObserver(), 'multa_aplicada');
        $subject->attach(new RegistradorHistorialObserver(), 'turno_bloqueado');
        $subject->attach(new RegistradorHistorialObserver(), 'sancion_aplicada');
        return $subject;
    }
}
```

El sistema lleva al mismo tiempo el manejo de 3 roles. Cuando se aplica una multa, el conductor debe recibir la notificación correspondiente, el gerente es informado, y el historial debe quedar registrado. Sin la implementación del Observer el código necesitaba ser escrito en cada controlador.

Patrón Chain of Responsability

Cada una de las validaciones que existe en el sistema se hace en cadena. Una solicitud pasa por esa secuencia en orden. Si uno llega a fallar se detiene y si todo pasa la operación sigue en curso. Antes de registrar una salida, la solicitud debe pasar por 4 validaciones las cuales son: verifica que el conductor no esté bloqueado, luego si se encuentra dentro del horario permitido, debe estar dentro de la goecerca o parada y finalmente no poseer una sanción activa, si cualquiera de estas acciones no se cumple la cadena se detiene.

```
// 1. Interfaz que define como funciona cada validador
interface ValidacionHandlerInterface {
    public function setNext(ValidacionHandlerInterface $handler): ValidacionHandlerInterface;
    public function handle(array $context): ?array;
}
// 2. Si esta validación pasa, pasa la petición al siguiente
abstract class BaseHandler implements ValidacionHandlerInterface {
    protected ?ValidacionHandlerInterface $next = null;
    public function setNext(ValidacionHandlerInterface $handler): ValidacionHandlerInterface
{
        $this->next = $handler;
```

```
        return $handler;
    }
    public function handle(array $context): ?array {
        $resultado = $this->validar($context);
        if ($resultado !== null) { return $resultado; }
        if ($this->next) { return $this->next->handle($context); }
        return null;
    }
    abstract protected function validar(array $context): ?array;
}
// 3. Revisa si el conductor está en la zona permitida

class VerificarGeocercaHandler extends BaseHandler {
    protected function validar(array $context): ?array {
        $distancia = $this->calcularDistancia(
            $context['latitud'], $context['longitud'],
            $context['geocerca_lat'], $context['geocerca_lng']
        );
        if ($distancia > ($context['geocerca_radio'] ?? 50)) {
            return ['error' => true, 'mensaje' => "Fuera de la geocerca", 'codigo' => 'GEOCERCA'];
        }
        return null;
    }
    protected function calcularDistancia(float $lat1, float $lng1, float $lat2, float $lng2): float {
        $earthRadius = 6371000;
        $dLat = deg2rad($lat2 - $lat1);
        $dLng = deg2rad($lng2 - $lng1);
        $a = sin($dLat / 2) ** 2 + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) * sin($dLng / 2) ** 2;
        return $earthRadius * (2 * atan2(sqrt($a), sqrt(1 - $a)));
    }
}
// 4. Armado de la cadena completa de validación

class ValidacionSalidaPipeline {
    protected ValidacionHandlerInterface $pipeline;
    public function __construct() {
        $bloqueo = new VerificarBloqueoHandler();
        $horario = new VerificarHorarioHandler();
        $geocerca = new VerificarGeocercaHandler();
        $sancion = new VerificarSancionHandler();
        $bloqueo->setNext($horario)->setNext($geocerca)->setNext($sancion);
        $this->pipeline = $bloqueo;
    }
    public function validar(array $context): ?array {
        return $this->pipeline->handle($context);
    }

    public function esValido(array $context): bool {
        return $this->validar($context) === null;
    }
```

```
}
```

Sin este patrón, las validaciones estarían en un método inmenso dentro del controlador, mezclando reglas de negocio con lógica HTTP. Además, agregar una nueva validación necesitaría solo crear una clase nueva y añadirla a la cadena, sin tocar el código que ya existe. Ejemplo antes de registrar una salida se ejecuta (new ValidacionSalidaPipeline())->validar($context) si funciona bien la operación sigue, sino es así devuelve error y se detiene.

Patrón Strategy

Este patrón define una serie de algoritmos denominados familia, se encapsulan y se intercambian al mismo tiempo de ejecución sin que sepan los detalles de cada uno. El propósito de la integración en el sistema es recubrir la fórmula de cálculo de un tipo de multa en su propia clase, donde hay tres tipos de multa como: retraso por llegar minutos tarde a una parada, inasistencia o no se presentó y si es puntero no cumplió el horario puede operar hasta las 9 am. Se destaca por realizar un cálculo diferente en cada uno de la función y selecciona el tiempo de ejecución.

```
// 1. Interfaz que define como debe funcionar cualquier cálculo de multa
interface MultaStrategyInterface {
   public function calcular(float $valorBase, array $params = []): float;
   public function getDescripcion(): string;
   public function getTipo(): string;
}
// 2. Multa por retraso (minutos x valor por minuto)
class MultaRetrasoStrategy implements MultaStrategyInterface {
   public function calcular(float $valorBase, array $params = []): float {
      $minutosRetraso = $params['minutos_retraso'] ?? 0;
      $multaPorMinuto = $params['multa_por_minuto'] ?? 0.50;
      return $minutosRetraso * $multaPorMinuto;
   }
   public function getDescripcion(): string { return 'Multa por retraso'; }
   public function getTipo(): string { return 'retraso'; }
}
// 3. Clase que recibe el tipo de multa y usa la fórmula adecuada
class MultaContext {
   protected MultaStrategyInterface $strategy;
   public function __construct(MultaStrategyInterface $strategy) {
      $this->strategy = $strategy;
```

```
    }
    public function ejecutar(float $valorBase, array $params = []): float {
        return $this->strategy->calcular($valorBase, $params);
    }
    public static function fromTipo(string $tipo): self {
        return match ($tipo) {
            'retraso' => new self(new MultaRetrasoStrategy()),
            'inasistencia' => new self(new MultaInasistenciaStrategy()),
            'puntero_9am' => new self(new MultaPuntero9AMStrategy()),
            default => throw new \InvalidArgumentException("Tipo desconocido: {$tipo}"),
        };
    }
}
```

Las multas en el sistema son de varios tipos, cada una se calcula distinto, en vez de usar un switch enorme que pregunte el tipo cada uno maneja su propia clase con su fórmula. Para calcular la multa en ese patrón solo se escribe MultaContext:: fromTipo('retraso')->ejecutar($base, $params). Cada tipo de multa tiene cálculo diferente, si se crea a futuro más multas bastaría con solo crear una clase nueva.

Patrón Adapter

El propósito de este patrón es establecer que dos clases con interfaces diferentes o no compatibles trabajen de manera igualitaria, convirtiéndose en una dependencia donde una espera algo de la otra. En el sistema la función del patrón es traducir el api de Google maps a una interfaz propia del sistema y el resto del sistema solo conoce sus métodos, pero no tiene conocimiento que internamente hace uso de Google Maps.

// 1. Interfaz propia del sistema para servicios de mapas

```
interface MapsAdapterInterface {
    public function getStaticMap(float $lat, float $lng, int $zoom = 15, int $width = 400, int
$height = 300): string;
    public function getEmbedUrl(float $lat, float $lng, int $zoom = 15): string;
    public function getDirectionsUrl(float $originLat, float $originLng, float $destLat, float
$destLng): string;
    public function geocode (string $address):? array;
    public function getDistanceMatrix(array $origins, array $destinations): ?array;
}
```

// 2. Adaptador que traduce Google Maps a la interfaz del sistema

```
class GoogleMapsAdapter implements MapsAdapterInterface {
    protected string $apiKey;
    public function __construct() {
        $this->apiKey = config('services.google_maps.api_key', '');
```

```
    }
    public function getStaticMap(float $lat, float $lng, int $zoom = 15, int $width = 400, int
$height = 300): string {
        return
"https://maps.googleapis.com/maps/api/staticmap?center={$lat},{$lng}&zoom={$zoom}&siz
e={$width}x{$height}&key={$this->apiKey}";
    }
    public function getEmbedUrl(float $lat, float $lng, int $zoom = 15): string {
        return
"https://www.google.com/maps/embed/v1/view?center={$lat},{$lng}&zoom={$zoom}&key=
{$this->apiKey}";
    }
    public function getDirectionsUrl(float $originLat, float $originLng, float $destLat, float
$destLng): string {
        return
"https://www.google.com/maps/embed/v1/directions?origin={$originLat},{$originLng}&dest
ination={$destLat},{$destLng}&key={$this->apiKey}";
    }
    public function geocode(string $address): ?array { /* ... */ }
    public function getDistanceMatrix(array $origins, array $destinations): ?array { /* ... */ }
}
```

El sistema implementa la utilización de Api de Google Maps para visualizar los mapas y calcular distancias. Gracias a esta abstracción los controladores y las vistas operan con métodos estandarizados sin acoplarse a la tecnología subyacente. Si en algún momento la Compañía desea cambiar a otra plataforma de GPS este patrón aísla los servicios externos y requeriría crear una sola clase y no modificar muchos archivos.

La implementación de los patrones de diseño Gof ayudó administrar mejor la fase de programación. Se llegó obtener una transición completa de una lógica de código acoplada hacia un entorno escalable, cada componente tiene su única responsabilidad definido esto permite tener una alta cohesión en la estructura. El sistema gestiona diversos roles como: normativas que están sujetos a cambios y operaciones ejecutadas de manera concurrente, con la integración de los patrones mencionados mejora y sustenta la integridad del software. Cada uno estructuró la lógica de negocio previniendo el acoplamiento excesivo de los módulos.

Como resultado el sistema adquirió un alto grado de escalabilidad y mantenibilidad, cumpliendo con el principio de Abierto/Cerrado (OCP). Esto permite asegurar el código

fuente pueda ser adaptable a futuros requerimientos sin ser comprometidos la estabilidad operativa de los componentes que ya se encuentran en producción.

4.4 Pruebas

El sistema web de gestión operativa y control automatizado de control de Salida y llegada en la Compañía Antoño Polo S.A. el propósito de las pruebas es medir el valor, del sistema culminado con todos los actores involucrados. Permitirá detectar errores, corregir y establecer un producto acorde a la calidad de software, con el fin de garantizar un producto excepcional.

Ejecución de las pruebas

Tabla 25: *Herramientas de pruebas*

Herramientas de pruebas	
Exploradores web	Chrome y Brave
Sistemas operativos	Windows
Lenguajes de código fuente	PHP, JavaScript
Servidor Web	Apache
Framework	Laravel
Motor de bases de datos	MySQL

Tabla 26: *Usuarios finales*

Equipo de testeo	El deber del equipo que va ejecutar las pruebas del sistema es garantizar su funcionamiento según el análisis de las pruebas.
Usuarios finales del sistema (Gerente, tarjetera y conductor)	El gerente tiene un rol de administrador como; gestionar usuarios, crear nuevas unidades (camionetas), configurar los parámetros operativos, monitorear en el GPS de la ruta de cada unidad y generar reportes.

	La tarjetera podrá ejecutar el sorteo diario, controlar las colas, aprobar o rechazar justificativos, registrar salida y generar reportes diarios de la jornada El conductor registra la salida, ver los tunos y puede justificar las multas o sanciones en caso de existir

4.4.1 Pruebas unitarias

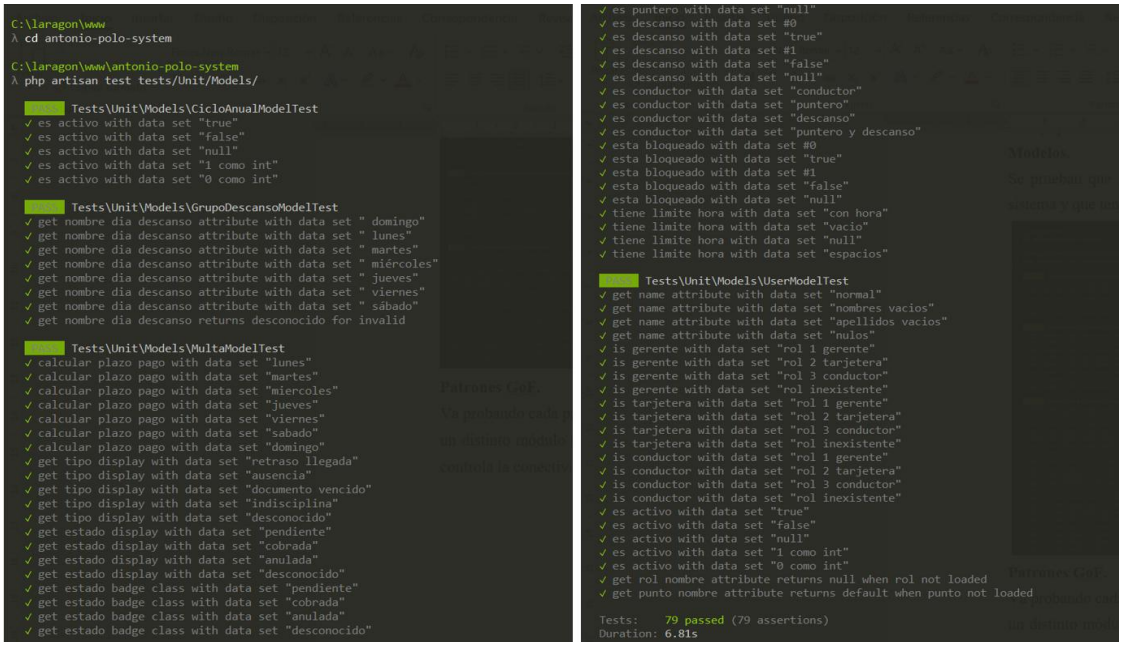
Las pruebas unitarias son automatizadas que cumplen con el rol de validar una función dentro del código del programa, es aislada sin depender de la base de datos ni HTTP.

Estas se ejecutan con comandos en la terminal del proyecto buscando que sea rápida y garantiza que cada componente en nuestro sistema funcione de manera adecuada a lo que se espera. Nos permite ver si la prueba pasa o no y en que parte exactamente está la falla y que la causó. En nuestro sistema aplicamos las siguientes pruebas unitarias:

Modelos.

Se prueban que los datos tengan lógica interna de los modelos validando métodos y que cada acceso nos devuelva datos según corresponda (verificación de roles, tipos de turno, multas, etc).

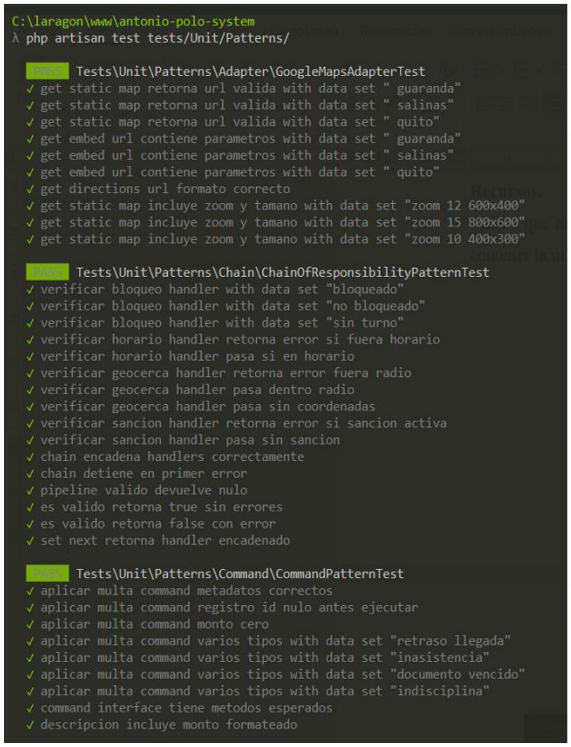
Ejecución de prueba de los módulos



Patrones GoF.

Va probando cada patrón de diseño implementado dentro del sistema cada uno controla un distinto módulo (multas, notificaciones, etc) y sobre todo lo más importante como controla la conectividad con Google Maps.

Ejecución en el código





Servicios.

Prueba el servicio de multas verificando que haya un control en la tolerancia establecida en cada punto de operación y la hora del día, así también calcula minutos de retraso del conductor al verificar si este llegó dentro o fuera del tiempo establecido.

Ejecución en el código

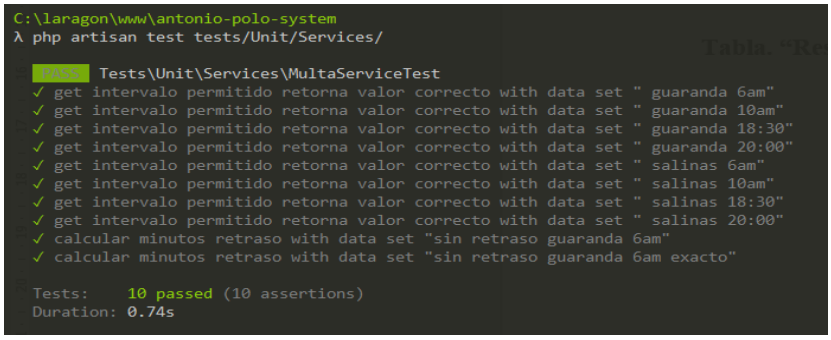


Tabla 27: *Tabla de pruebas unitarias.*

Modulo/Componente	**Función evaluada**	**Cumple**	
		SI	**NO**
Gestión de usuarios	Validación de crear perfiles, encriptación de contraseñas y control de cédulas duplicadas.	✓	
Gestión de turnos	Generación de cálculo automático del intervalo de 15 minutos para el despacho y ordenación de la cola de salida.	✓	
Servicios de multas	Ejecución de cálculo de tiempo de ruta, esto genera una operación exacta de sanción si el viaje supera los 60 min.	✓	
Grupos de descanso	Asignación de manera automática en aleatorio de las unidades para la jornada del día, tomando en cuenta que los grupos inactivos sean excluidos del sorteo diario.	✓	
Ciclos anuales	Creación del periodo anual estableciendo la restricción lógica, que exista un ciclo activo a la vez.	✓	
Patrones GoF			
Adapter	Construcción de URLs válidas para mapas estáticos y enrutamiento de direcciones	✓	
Chain of Responsibility	Validación secuencial de precondiciones como despacho, horario, punto dentro y fuera de radio establecido, donde la cadena se detiene al primer error.	✓	
Command	Encapsulamiento de la aplicación de multas como comandos según el tipo.	✓	
Observer	Notificación de eventos del turno como: asignación, salida, llegada, multa y bloqueo.	✓	

	Estos eventos muestran a los observadores conductor, gerente y tarjetera		
State	Control de cambio de estado de turno, aceptando las válidas y rechazando las no permitidas	✓	
Strategy	Selección de las multas depende el tipo como retraso de minutos a la parada destino inasistencia a la parada sin presentar la justificación	✓	

4.4.2 Pruebas de funcionalidad

Tabla 28: *Caso de pruebas de Modulo Administración del sistema*

ID Prueba	Historia de usuario	Descripción	Resultado	Cumple	
				Si	No
Cp-1.1	HU-1: Gestión de usuarios	Registrar un nuevo conductor en el sistema	El conductor se crea exitosamente y se genera una clave temporal de 8 digitos	✓	
Cp-1.2	HU-2: Autenticación	Ingresar al sistema con usuario y contraseña en caso de ser nuevo usuario debe cambiarlo obligatoriamente. Verificar mensaje de error con credenciales incorrectas, mensaje de usuario inactivo	El sistema valida correctamente las credenciales. Redirige al formulario de cambio de contraseña. Cada rol es redirigido a su propio dashboard.	✓	
Cp-1.3	HU-3: Gestión unidades	Registrar nuevas unidades (camionetas). Ejecutar cambio de estado a activo a unidad dado de baja.	Cuando la unidad esta activa participa en el sorteo y asignación de turnos y si no es excluida en el sorteo.	✓	
Cp-1.4	HU-4: Parámetros	Configurar parámetros de multa por minuto,	Los parámetros globales como;		

		tiempo estimado de ruta, multas y sanciones.	multa, tiempo, horarios, tolerancias, contacto se guardan y aplican correctamente.	✓	

Tabla 29: *Caso de prueba de Modulo planificación de turnos*

ID Prueba	Historia de usuario	Descripción	Resultado	Cumple	
				Si	No
Cp-2.1	HU-14: Ciclos anuales	Crear un ciclo anual con año, fecha inicio y fecha fin. Activarlo y verificar que solo exista un ciclo activo.	La creación y activación del ciclo funcionan correctamente. El sistema impide activar dos ciclos simultáneamente	✓	
Cp-2.2	HU-5: Grupos descanso	Generar grupos de descanso aleatoriamente para Guaranda 6 y Salinas 7. Si ya existen grupos creados, el sistema debe pedir confirmación antes de recrearlos.	Los grupos se crean con asignación aleatoria de manera correcta. Los días de descanso se asignan entre grupos		X
Cp-2.3	HU-6: Sorteo diario	Ejecutar el sorteo diario del punto. Verificar que se excluya del sorteo al grupo de descanso y se asignen 2 punteros aleatorios y el resto de unidades. Verificar que elimine turnos del día antes de generar nuevos.	El sorteo funciona correctamente. Excluye el grupo de descanso. Elimina turnos anteriores antes de generar un nuevo.	✓	

Tabla 30: *Caso de prueba de Modulo Operación diaria*

ID Prueba	Historia de usuario	Descripción	Resultado	Cumple	
				Si	No
Cp-3.1	HU-7: Presencia	El conductor confirma su asistencia desde su dashboard GPS. Verificar que valide la geocerca,	El sistema solicita permiso de geolocalización al navegador. Valida		**X**

		bloquea si tiene documentos vencidos y notifique a la tarjetera.	que este dentro de la geocerca.		
Cp-3.2	HU-8: Control cola	La tarjetera realiza dos llamadas y bloqueo de un conductor que no se presenta a la tercera. Verificar que la cola se renumere automáticamente.	notificación push al conductor de su turno correspondiente. La tarjetera puede sancionar con $12.00 a los punteros que no se presentaron	✓	
Cp-3.3	HU-9: Registro salida	La tarjetera registra la salida de un conductor en su panel. El conductor registra su propia salida, valida que esté dentro de del punto de origen.	La tarjetera registra la salida de la unidad. En horas que no labora, el conductor puede registrar su propia salida.	✓	
Cp-3.4	HU-10: Registro llegada	El sistema mediante el GPS de conductor detecta automáticamente la llegada al geocerca. Verificar que se registre la hora y el tiempo de trayecto en minutos.	El panel activa seguimiento GPS continuo. Al detectar que las coordenadas están dentro del radio, el sistema registra la llegada automáticamente		**X**

Tabla 31: *Caso de prueba de modulo Multa y sanciones*

ID Prueba	Historia de usuario	Descripción	Resultado	Cumple	
				Si	No
Cp-4.1	HU-10: Multa retraso	El conductor llega al destino después del tiempo permitido 60 minutos y genera automáticamente $1 dólar por minuto.	El sistema genera la multa automáticamente al registrar la llegada si el tiempo supera el límite	✓	

Cp-4.2	HU-8: multa por no presentación	El conductor puntero no confirma presencia. La tarjetera aplica la multa de $12.00 y el conductor reciba notificación.	La tarjetera sanciona al puntero desde el control de cola. El sistema genera automáticamente multa de $12.00	✓	
Cp-4.3	HU-11: Justificativos	Un conductor bloqueado crea un justificativo. La tarjetera aprueba o no. Verificar que el turno se desbloquee, se reubique al final de la cola y también permite desbloquear las multas	El conductor crea el justificativo. Al aprobar se desbloquea, se mueve al final de la cola, las multas pendientes quedan anuladas y se notifica al conductor.		**X**

Tabla 32: *Caso de prueba de Modulo de monitoreo y documentación*

ID Prueba	Historia de usuario	Descripción	Resultado	Cumple	
				Si	No
Cp-5.1	HU-12: Monitoreo GPS	Verificar que el mapa muestre en tiempo real las posiciones de las unidades en ruta en salinas y Guaranda	El mapa muestra marcadores de unidades. Los dos puntos se visualizan como círculos en el radio. El sistema genera alertas automáticas al detectar entrada/salida de geocercas	✓	
Cp-5.2	HU-13: Control documental	Registrar licencia de conducir y matrícula de un con tipo, número y fechas. Verificar que el sistema notifique con anterioridad y bloquee la operación si el documento está vencido	El registro de documentos funciona correctamente. El sistema detecta documentos vigentes próximos a vencer. Si tiene documentos vencidos, el sistema bloquea la confirmación de presencia.	✓	

4.2.3 Pruebas de rendimiento

Estas pruebas son simulaciones que actúan como un caso real de uso en el sistema con muchos usuarios que hacen diversas peticiones al mismo tiempo y ayuda controlar el sistema si responde rápido, no se caiga y soporte la carga de trabajo que puede tener del día al día.

Lo que vamos a medir con estas pruebas son tiempo de respuesta, errores en la carga, usuarios usando el sistema al mismo tiempo y operaciones que procesa un servidor.

La herramienta a usar es K6 v2.0 Grafana.

Tabla 33: *Tabla de prueba de rendimiento*

Prueba	**Descripción**	**Que se mide**	**¿Cumple?**	
			SÍ	**NO**
Sorteo Diario	Ejecutar sorteo para 30 conductores y crear grupos de descanso	Tiempo de respuesta < 3s	X	
Turnos del Día	Confirmar presencia, llamar conductor, registrar salida	Tiempo de respuesta < 2s	X	
Calcular Multas	Calcular multas automáticas por retraso para todos los turnos del día	Tiempo de respuesta < 4s	X	
Logeo	50 usuarios iniciando sesión al mismo tiempo	Tiempo de respuesta < 2s	X	
Reportes	Gerente consulta dashboard con estadísticas en tiempo real	Tasa de error < 2%	X	
GPS	50 conductores enviando posición GPS simultáneamente	Tiempo de respuesta < 3s	X	
Estrés General	Subir de 10 a 150 usuarios hasta encontrar el punto de quiebre	Registrar en qué número falla	X	

4.2.4 Pruebas de seguridad

Estas pruebas son especializadas en controlar que el sistema desarrollado no presente vulnerabilidades, las cuales pueden ser; permitir que otros usuarios ingresen a roles que no les corresponden, que puedan ver la información de otra persona, robar contraseñas o vulnerar directamente a la base de datos. Con estas pruebas determinaremos si el sistema cumple con los requisitos de seguridad para operar evitando riesgos legales y económicos.

La herramienta a usar PHPUnit 12.5 (test automatizados) y OWASP ZAP (escáner de vulnerabilidades.)

Tabla 34: *Tabla de Pruebas de seguridad*

Prueba	**Descripción**	**¿Cumple?**	
		SÍ	**NO**
Control de roles	conductor, tarjetero y gerente no acceden a rutas de otros roles.	X	
Sin sesión redirige al login	Usuario no autenticado no entra a rutas protegidas	X	
Sesión cerrada no se reusa	Después de salir del sistema, no se vuelve a reutilizar la cookie.	X	
Aislamientos de datos	Conductor no visualiza las multas de otro.	X	
Inyección SQL	Inyección SQL no daña la BD.	X	
Ataque XSS	Scripts en nombres se escapan.	X	
API no expone contraseñas	JSON de usuarios no contiene las contraseñas del usuario.	X	

4.2.5 Pruebas de integración

Estas pruebas verifican gran parte de los módulos del sistema funcionen juntas de manera correcta, mas no por separado. En comparación con las pruebas unitarias es que, una unitaria verifica que las funciones trabajen bien por sí solo, las de integración revisa que una función X llame a la función Y, se guarde en la base de datos, y luego Z lea los

datos, es decir, que haya un camino que funcione completamente sin errores. Lo que verdaderamente importa en esta prueba es la comunicación entre las funciones y que al ejecutarlas no se rompa nada en el sistema.
La herramienta a usar PHPUnit 12.5 + SQLite en memoria.

Tabla 35: *Prueba de integración flujo de multas*

Prueba	**Descripción**	**¿Cumple?**	
		SÍ	**NO**
Cobrar Multa	Que, al cobrar una multa pendiente, el sistema la marca como cobrada y guarda quién y cuándo la cobró	X	
Guardar multa por ausencia	Crear una multa de tipo "ausencia" con un monto, los datos quedan bien guardados en el sistema	X	
Estado según día de semana	El sistema decide correctamente si la multa queda como "pendiente" (día normal) o "pendiente_semana" (fin de semana)	X	
Calcular minutos de retraso	Calcular cuántos minutos de retraso tuvo un viaje (06:00 a 06:30), el resultado no da números negativos	X	
Obtener intervalo permitido	Consultar cuántos minutos se permite entre salida y llegada, el sistema da un valor válido	X	

Tabla 36: *Prueba de integración flujo de notificaciones.*

Prueba	**Descripción**	**¿Cumple?**	
		SÍ	**NO**
Evento crea cola de trabajo	Cuando ocurre un evento (ej: se genera una multa), el sistema encola automáticamente un trabajo para crear la notificación	X	

Trabajo crea notificación	Se crea el registro de notificación en la base de datos para que el conductor la vea	X	
Notificación con referencia	La notificación está relacionada a algo (ej: una multa específica #5), esa referencia se guarda correctamente	X	
Varias notificaciones independientes	Ocurren dos eventos distintos (una multa y un recordatorio), se crean dos notificaciones separadas, una para cada una	X	

Tabla 37: *Prueba de integración flujo de historial de cambios.*

Prueba	**Descripción**	**¿Cumple?**	
		SÍ	**NO**
Guardar cambio en el historial	Cuando se hace un cambio, el sistema guarda un registro en el historial con la tabla afectada, el registro, la acción y quién lo hizo	X	
Guardar valor anterior y nuevo	Cuando algo cambia de estado, el sistema guarda cómo estaba antes y cómo quedó después	X	
Guardar descripción	Se escribe una descripción al registrar un cambio, esa descripción se guarda junto con el historial	X	

Tras llevar a cabo la ejecución de pruebas (pruebas unitarias, funcionales, integración, rendimiento y seguridad) acorde a las métricas de calidad de software, se determina que, el sistema ha alcanzado un nivel óptimo de estabilidad cumpliendo con los requerimientos previamente planteados para el desarrollo del sistema. Los posibles errores detectados durante las pruebas de estrés y escaneo mediante la herramienta OWASP ZAP fueron escaneados con anterioridad bajo las metodologías ágil XP, garantizando un sistema seguro, escalable y estable.

CONCLUSIONES

- Se implementó con éxito un sistema web para la gestión operativa de la compañía Antoño Polo S.A. Esta plataforma tecnológica remplazó los procesos logísticos manuales, consolidando un entorno centralizado que garantiza el monitoreo en tiempo real, el cálculo automatizado de tiempos y la trazabilidad de la flota de unidades entre los puntos de Guaranda y Salinas mediante geolocalización.
- La fase de análisis fue útil para la identificación de los problemas de la compañía. La asignación manual de turnos junto con el registro de sanciones generaba inconsistencias y pérdida de información. Este levantamiento, gestionado mediante enfoques ágiles, permitió estructurar con exactitud las historias de usuario y delimitar los requerimientos funcionales críticos del sistema.
- El diseño arquitectónico del sistema demostró ser altamente escalable fundamentado en la implementación del patrón Modelo-Vista-Controlador (MVC) y documentado a través del modelo de vistas 4+1 de Kruchten. al igual que las bases de datos, coincidieron con los requerimientos previamente establecidos permitiendo poseer una estructura rustico en el modelaje.
- La implementación, se codificaron módulo tras módulo utilizando los patrones de diseño de software, logrando establecer una funcionalidad adecuada en el sistema y la integración de Api de Google Maps para la interactividad en la Geolocalización de la gestión operativa de las unidades y las interacciones entre los usuarios del sistema.
- Las pruebas ejecutadas ayudaron a identificar los errores y corregir de manera específica mismo que ayudo a mejorar el sistema y establecer el adecuado funcionamiento, verificando que el sistema final cumpla con los requisitos establecidos previamente, esto permitió obtener un software de calidad y escalable.

RECOMENDACIONES

- La compañía debería darle continuidad al uso del sistema web que se ha implementado para que se maneje una gestión ordenada de sus procesos operativos y de uso necesario. Esto garantizará realizar el control automatizado y la optimización del desempeño diario del personal y operativo.
- Es importante fomentar la alfabetización digital al personal sobre el uso de esta herramienta tecnológica, esto ayuda a mantener el mejor uso de la información y dejar atrás las actividades manuales que generaban problemas.
- Se recomienda aprovechar la alta escalabilidad que provee el patrón arquitectónico Modelo-Vista-Controlador (MVC) y el diseño normalizado de la base de datos para desarrollar nuevos módulos complementarios. A medida que las necesidades operativas evolucionen dentro de la compañía.
- Se recomienda revisar periódicamente el funcionamiento de Google Maps dentro del sistema, porque al ser un servicio externo puede actualizarse sin avisar y si no se le presta atención, el mapa donde se ven las unidades puede dejar de funcionar justo cuando más se necesita.
- Se debe establecer políticas rigurosas de Aseguramiento de Calidad (QA) para cualquier modificación futura que se requiera hacer en el código fuente. Esto evitará la introducción de vulnerabilidades, garantizará la disponibilidad continua del sistema y mantendrá la confianza de los usuarios finales.

BIBLIOGRAFÍA

Aguilar, J., & Reyes, J. (2021). *Recolección, análisis y envío de datos de la unidad de control de motor (ecu) a bordo de un vehículo para monitoreo a través de una aplicación móvil* [Tesis de Pregrado, Universidad Politécnica Salesiana, Sede Cuenca, Carrera de Ingeniería Electrónica]. https://dspace.ups.edu.ec/handle/123456789/21348

Arias, J. (2020). *Técnicas e instrumentos de investigación científica. Para ciencias administrativas, aplicadas, artísticas, humanas*. ENFOQUES CONSULTING EIRL. www.cienciaysociedad.org

Asamblea Nacional de La República Del Ecuador (2016).

Asamblea Nacional de La República Del Ecuador (2021).

Beekman, G. (2005). *Introducción a la Informática* (6th ed.). Pearson Educación, S.A. www.pearsoneducacion.com

Biradar, V. (2025). Tailwind CSS. *International Journal of Advance and Applied Research*. https://doi.org/10.5281/zenodo.15194897

Bravo, E., Cieza, S., Flores, L., & Uceda, L. (2022). *Revisión Sistemática de la Literatura sobre Metodologías Ágiles en el Desarrollo de Software aplicadas a la Gestión de Proyectos*.

Cabrera, C., & Briceño, W. (2025). *Desarrollo de un programa estadístico para el análisis y gestión de información de geoposición vehicular de la empresa ETAPA EP* [Tesis de Pregrado, Universidad Politécnica Salesiana]. https://dspace.ups.edu.ec/bitstream/123456789/31488/1/UPS-CT012398.pdf

Conde, K. (2025). *Desarrollo de una aplicación web aplicando modelos predictivos para la gestión y notificación de siniestros de tránsito* [Tesis de Pregrado]. Universidad Internacional del Ecuador.

Damiani, F. (2021). *Sistema de gestión operativa y automatización de procesos utilizando RPA en una compañía de servicio de monitoreo vehicular GPS.* [Tesis de Pregrado, Universidad Peruana de Ciencias Aplicadas, Facultad de Ingeniería, Programa Académico de Ingeniería de Sistemas]. http://hdl.handle.net/10757/657828

Ecuador. Presidencia de la República. (2016). Decreto No. 975: Refórmese el Reglamento General para la Aplicación de la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial. In *Registro Oficial Suplemento Nº 741*.

Flórez, H., & Hernández, J. (2021). *Aplicaciones web con PHP*. Ediciones de la U.

García, P., & Saavedra, F. (2026). *Efectos del sistema web de control por GPS en la gestión de la flota vehicular de la empresa GARSAA, Callao – 2025* [Tesis de Pregrado]. Universidad Peruana de Ciencias e Informática, Facultad de Ciencias e Ingeniería, Carrera Profesional de Ingeniería de Sistemas e Informática.

García, S., & Toscano, J. (2025). Metodologías de Gestión de Proyectos Tradicionales y Ágiles y su Aplicación en el Sector Industrial. *Ciencia Latina Revista Científica Multidisciplinar, 9*, 6148–6164. https://doi.org/10.37811/cl_rcm.v9i6.21753

Maida, E., & Pacienzia, J. (2015). *Metodología de desarrollo de software* [Tesis de licenciatura]. Pontificia Universidad Católica Argentina.

Medina, M., Rojas, R., Bustamante, W., Loaiza, R., Martel, C., & Castillo, R. (2023). *Metodología de la investigación: Técnicas e instrumentos de investigación*. Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú. https://doi.org/10.35622/inudi.b.080

Palacios, J. (2025). *Desarrollo de una aplicación web para la gestión de la logística de transporte de carga pesada en una empresa* [Tesis de Pregrado, Universidad Técnica de Machala, Facultad de Ingeniería Civil, Carrera de Ingeniería de Sistemas .]. https://repositorio.utmachala.edu.ec/server/api/core/bitstreams/819e60b3-c1a7-4ae8-9b60-4ed8c959f1d8/content

Palomino, E., & García, A. (2023). La geolocalización como medio de prueba. *Revista Alegatos*, *37*(114).

Pérez, S., Quispe, J., Mullicundo, F., & Lamas, D. (2021). *HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE EL FRONTEND AL BACKEND*. https://www.campusmvp.es/recursos/post/Desar

Postigo, A. (2023). *Administración de sistemas gestores de bases de datos*. Ediciones Paraninfo.

Puciarelli, L. (2020). *Angular: TypeScript - Arquitectura - Instalación - Directivas y Bindings*. Six Ediciones.

Reinoso, P. (2024). *Aplicación web y móvil para el monitoreo de la operatividad de las unidades de trasporte de taxi para la empresa de servicio de radio taxi "Ibarra", haciendo uso de geoposicionamiento* [Titulo de Pregrado, Pontificia Universidad Católica del Ecuador, Escuela de Informática e Inteligencia Artificial .]. https://repositorio.puce.edu.ec/server/api/core/bitstreams/8f5d1ef4-5436-4384-8648-078da3830d1f/content

Rodríguez, R., Vera, P., Martínez, M., Alderete, C., & Dogliotti, M. (2020). *Aplicaciones Web Progresivas Enfocadas en el Uso y Optimización de Cache*.

Tomalá, S. (2025). *Sistema web para gestión de información de actividades de postcosecha en cultivo banano para la empresa ENROBAN SAS del cantón el triunfo* [Tesis de Pregrado]. Universidad Agraria del Ecuador.

Yuqui, C., & Motoche, K. (2024). *Desarrollo de un sistema web para la automatización de procesos de transporte pesado en la ciudad de Quito* [Tesis de Pregrado, Universidad Iberoamericana del Ecuador, Facultad de Tecnología de la Información y Comunicación]. http://repositorio.unibe.edu.ec/xmlui/bitstream/handle/123456789/749/YUQUI%20DE

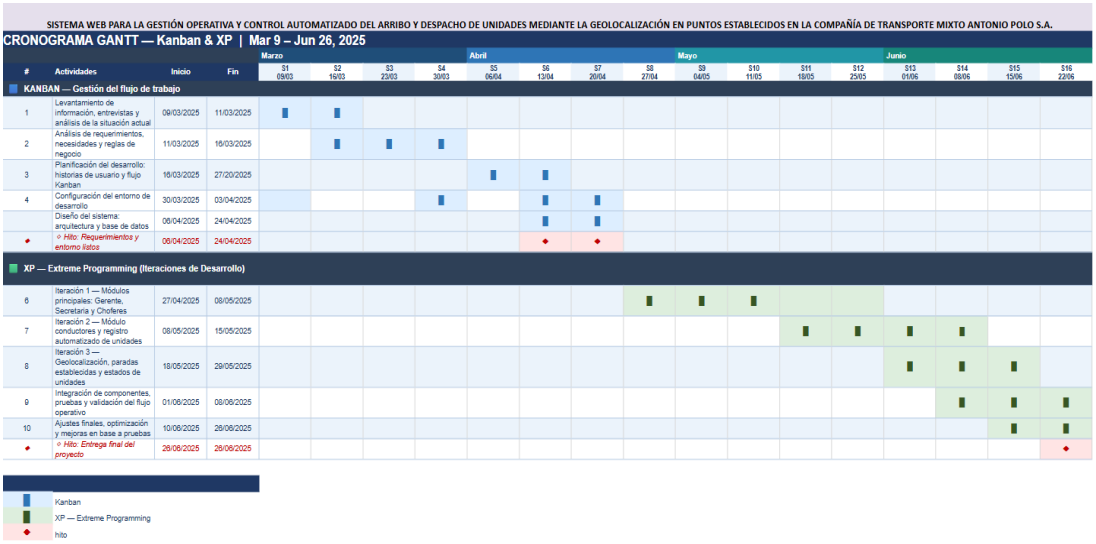
MERA%20CAROLINA%20MISHELL%20y%20MOTOCHE%20ROM%C3%81N%20KEVIN%20JOEL.pdf?sequence=1&isAllowed=

Anexo 1

(Cronograma de Actividades)

Diagrama de Gantt

Figura 20: Cronograma de Gantt



Elaborado: *Renan Sigcha y Alexander Agualongo*

Nota: Diagrama de actividades estructurado por fases.

Anexo 2

(Presupuesto del proyecto)

Tabla 38: *Tabla de presupuesto del proyecto*

Recursos	**Subtotales**
Dispositivo móvil	$130.00
Laptop	$905.00
Internet	$25.00
Impresiones	$10.00
Dominio	$25.00
Subtotal	$1095.00
Imprevisto 5%	$54.75
Total	$1149.75

Anexo 3

Instrumento de recopilación de datos (Entrevista)

FICHA DE ENTREVISTA

Universidad Estatal de Bolívar

Facultad de Ciencias Administrativas, Gestión Empresarial e Informática

Carrera de Ingeniería de Software

Sistema web para la gestión operativa y control automatizado del arribo y despacho de unidades mediante Geolocalización en puntos establecidos en la compañía de transporte Antoño Polo S.A.

Objetivo: Obtener información detallada sobre los procesos manuales actuales de arribo, despacho, asignación de turnos, problemas operativos frecuentes y expectativas para la implementación del nuevo sistema web.

Nombres de los entrevistadores: Renan Marcial Sigcha Punina y Steven Alexander Agualongo Morocho.

Nombre del entrevistado: Leonardo Guamán

Cargo: Gerente General

CUESTIONARIO DE LA ENTREVISTA

1. ¿Cómo inicia la jornada laboral diaria y cuál es el horario del primer turno establecido?
2. Para el resto de los conductores, ¿de qué manera se realiza la asignación de turnos para el día laboral?
3. ¿Existe alguna dinámica operativa especial debido a la alta demanda de pasajeros en horas pico?
4. Al manejar todo de forma manual los sorteos, papelitos y WhatsApp, ¿cuáles son los problemas más frecuentes que enfrentan?
5. ¿Cuáles son las mayores limitaciones de utilizar registros físicos en lugar de un sistema?

6. ¿Qué expectativas tiene la compañía con la implementación de este nuevo sistema web basado en geolocalización móvil?
7. ¿Cómo está distribuida la flota para la operatividad y los días de descanso?
8. ¿Cuál es la lógica de rotación anual para los grupos de descanso?
9. ¿Cómo se gestionan los primeros turnos de la mañana denominados punteros?
10. ¿Cuáles son los intervalos de tiempo reglamentarios para el despacho de las unidades?
11. ¿Qué excepción se aplica si una unidad alcanza su capacidad máxima antes de cumplir su tiempo de intervalo?
12. ¿Qué sanción se aplica si un conductor no está presente en la parada al momento de su turno?
13. ¿Cómo afecta el registro manual actual a la comunicación entre las paradas de Salinas y Guaranda?
14. ¿Cómo se gestiona actualmente el cobro de estas sanciones económicas?
15. ¿Cómo maneja la cooperativa el volumen máximo de turnos asignados en las paradas principales?

Anexo 4

Instrumento de recopilación de datos (Observación)

Ficha de observación

Sistema web para la gestión operativa y control automatizado del arribo y despacho de unidades mediante Geolocalización en puntos establecidos en la compañía de transporte Antoño Polo S.A.

Datos generales

Lugar	Parada de la compañía de Antoño en la ciudad de Guaranda		
Nombre de la compañía o empresa	Compania de Transporte Antoño Polo S.A		
Area de observación	Gestión operativa de unidades		
Fecha	25/03/2026		
Hora de inicio	12:00 pm	**Hora de finalización**	14:00 pm
Observadores	Renan Sigcha y Alexander Agualongo		

Propósito de la observación

El objetivo central fue diagnosticar la situación actual de la gestión operativa de la Compañía de Transporte Antoño Polo S.A la misma que se basa en la logística de movilización de sus unidades entre las paradas autorizadas. A través de la observación directa y se ha desglosado este procedimiento en las siguientes actividades principales para evidenciar cómo se ejecuta actualmente de forma manual y dónde radican las fallas.

N°	Actividades	Intervienen en el proceso	Herramientas utilizadas	Observaciones
1	Registro de arribo origen	operadora de turno (Tarjetera).	Cuaderno de apuntes	El orden de salida de las camionetas de la parada se documenta a mano. genera reclamos inmediatos de chofer en caso de errores de turno.
2	Salida de la unidad (camionetas)	Chofer y operadora de turno (Tarjetera)	WhatsApp (dispositivo móvil)	Es notificado por la operadora verbalmente o revisa el listado de orden de turno en WhatsApp
3	Llegada de la unidad (camioneta)	Chofer de la unidad	Comunicación verbal o WhatsAp	Pregunta verbalmente el turno que le toca obligatoriamente y si no lo hace o la operadora está ocupada en otros registros no puede conocer el orden de turno designado.
4	Registro de llegada	operadora de turno (Tarjetera).	Cuaderno de apuntes	Para conocer el tiempo en la ruta cada unidad debe de comunicar con la operadora de otra parada.

ANEXO 5

Manual de usuario



MANUAL DE USUARIO

SISTEMA PARA LA COMPAÑÍA ANTONIO POLO S.A.

Guía Completa para el Personal

Introducción **4**
¿Dónde opera el sistema? 5
CAPITULO I **6**
Requisitos Técnicos 6
Navegadores Compatibles 6
Dispositivos 6
Conexión a Internet 6
Capítulo II **7**
¿Cómo iniciar sesión? 7
¿Cómo cerrar sesión? 8
Capítulo III **9**
Roles de Usuario 9
Tabla comparativa de permisos en el sistema. 9
Capítulo IV **10**
Para Gerentes (Administradores) 10
Panel Principal o Dashbord 10
Gestión de usuarios 12
Gestión de Unidades 15
Multas 17
Documentos 18
Turnos Asignados 19
Reportes 19
GPS y mapa en tiempo real 20
Parámetros del Sistema 21
Capítulo V **23**
Para Tarjetera 23
Panel Principal o Dashboard 23
Turnos del Día 23
Confirmar presencia de conductor 24
Multas del punto 25
GPS en tiempo real 26
Registrar Salida. 27
Sorteo Diario 28

Justificativos 28
Reportes 29
Capítulo VI 31
Para Conductores 31
Pantalla de Inicio 31
Mi turno del Día 32
Historial y Mis Multas 32
Justificativos 33
Notificaciones 33
Perfil y cambio de contraseña 34
Capítulo VII 36
Seguridad 36
Sesión cerrada por inactividad 36
Restablecimiento de la contraseña. 36
Contraseñas Temporales 36
ANEXO I 37
Glosario de términos 37
ANEXO II 38
Diagrama de Flujo de la Operación 38

INDICE DE FIGURAS

Figura 1: Pagina principal 4
Figura 2: Iniciar Sesión 7
Figura 3: Botones de acceso rápido del administrador (Gerente) 10
Figura 4: Pantalla principal del administrador 11
Figura 5: Menú lateral (Gerente) 11
Figura 6: Gestión de usuarios 12
Figura 7: Pantalla Modificar usuario 13
Figura 8: Pantalla de restear contraseña y desactivar 15
Figura 9: Pantalla de registro de nuevo unidades 16
Figura 10: Pantalla de operación diaria 19
Figura 11: Pantalla de reportes 20
Figura 12: Pantalla de monitoreo en tiempo real 21
Figura 13: Pantalla de Parámetros del sistema 21
Figura 14: Pantalla de acceso rápido (Tarjetera) 23
Figura 15: Pantalla de turnos del día (Tarjetera) 23
Figura 16: Pantalla de Multas (Tarjetera) 25
Figura 17: Pantalla de GPS en tiempo real 27
Figura 18: Pantalla de registrar salida 27
Figura 19: Pantalla de sorteo diario 28
Figura 20: Pantalla de justificativos 29
Figura 21: Pantalla Principal de reportes 29
Figura 22: Pantalla del conductor 31
Figura 23: Pantalla de visualización de turnos 32
Figura 24: Pantalla del historial turnos y multas 32
Figura 25: Pantalla de justificativo (Conductor) 33
Figura 26: Diagrama de flujo de operación 38

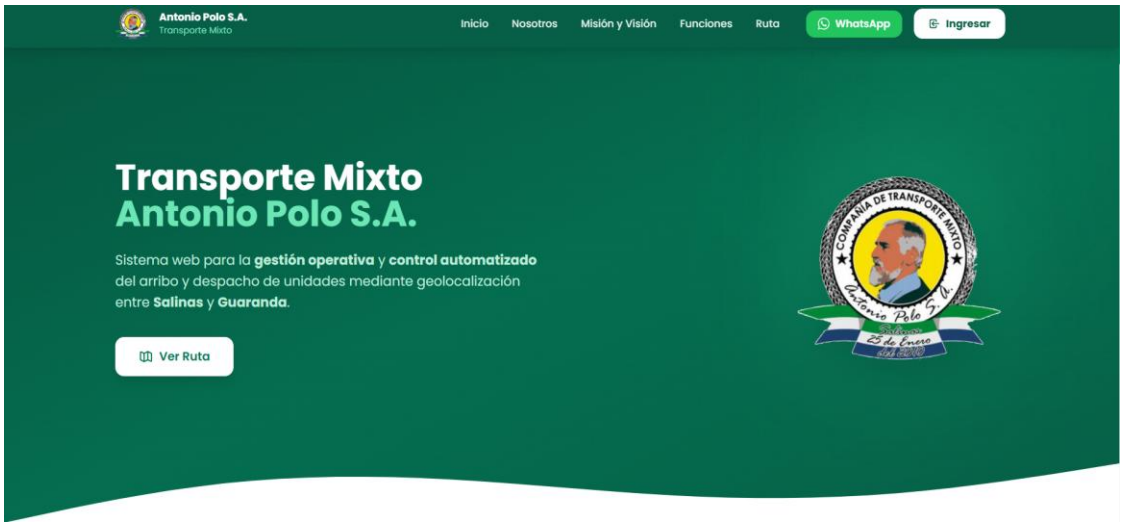
Introducción

El sistema Antoño Polo S.A es un programa que ayuda a manejar la operabilidad de la compañía de transporte. Este sistema brinda los siguientes beneficios:

- Control de conductores y turnos operativos.
- Registro de salidas y llegadas de cada trayecto.
- Aplicar multas según sea necesario.
- Ver mapa donde cada unidad se maneja por tiempo real.
- Generación de reportes para tener visualización de cómo se maneja la operación.

Para ingresar al sistema web, utiliza el siguiente link: https://sistemaantoniopolo-production.up.railway.app/#inicio

Figura 1: Página principal



Descripción de usuarios del sistema

El sistema tiene tres tipos de usuario, cada uno visualiza y realiza su función que le corresponda según las funcionalidades habilitadas.

Gerente: El dueño y administrador que puede ver todo, puede gestionar usuarios y unidades, ver reportes etc.

Tarjetera: Es la encargada de llevar a cabo gran parte del sistema, es la que maneja la asignación de turnos, grupos de descanso, salidas y llegadas, multas, etc.

Conductor: Es el que maneja la unidad, el visualiza su turno, confirma su presencia si es necesario, revisa sus multas, crea justificativos, etc.

¿Dónde opera el sistema?

El sistema que se ha implementado se gestiona y lleva operabilidad en la provincia de Bolívar en los sectores de Guaranda y Salinas. Los puntos de operación son los siguientes:

PUNTO	DIRECCIÓN
GUARANDA	Calle Selva Alegre y Salinas
SALINAS	Confites El Salinerito frente a la cooperativa Salinas

Capítulo I

1. Requisitos Técnicos

Navegadores Compatibles

El sistema funciona y está perfectamente soportado por navegadores modernos como:

- **Google Chrome:** versión 90 o superior.
- **Mozilla Firefox:** versión 88 o superior.
- **Microsoft Edge:** versión 90 o superior.
- **Safari:** versión 14 o superior.

Dispositivos

COMPUTADOR	Con pantallas de 1024px o más asegura una experiencia óptima
TABLET	Funciona muy bien en orientación horizontal
CELULAR	El sistema se adapta a las pantallas pequeñas

Nota: Para la función de la geolocalización en el sistema, el conductor debe tener una conexión de internet y el GPS activado.

Conexión a Internet

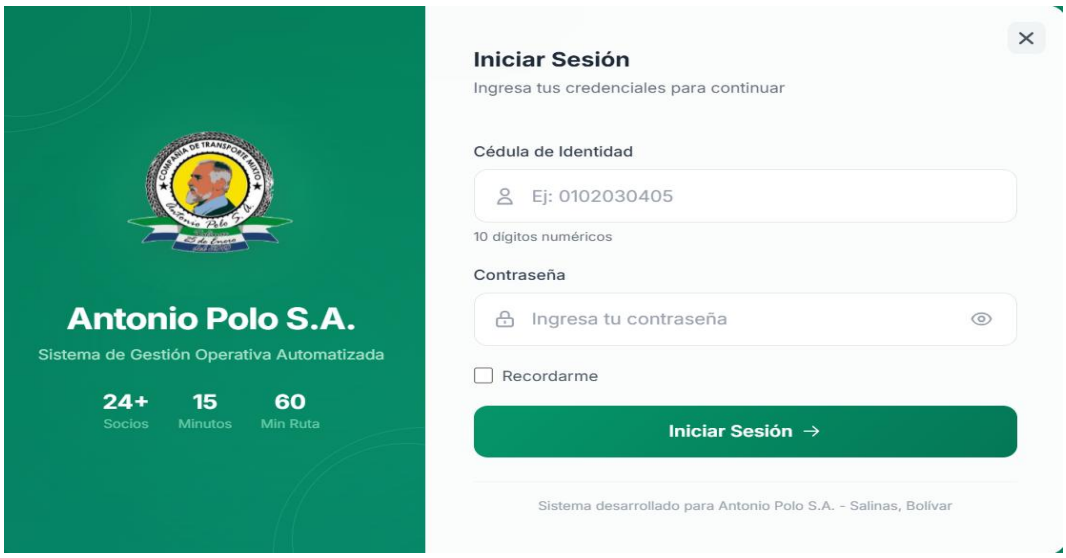
- Se necesita tener conexión a una red de internet para hacer el uso del sistema y poder iniciar sesión.
- La función de GPS en tiempo real requiere obtener una conexión estable.
- Los datos se guardan temporalmente en caso de perder la conexión y recupera los datos al recuperar la señal.

Capítulo II

¿Cómo iniciar sesión?

1. Abra su navegador sea Chrome, Firefox o Edge y vaya a la dirección del sistema en:
2. Va a ver la pantalla principal del sistema y se va a ir al botón que dice “Ingresar” y va a completar los campos con los datos para su respectivo logeo.
3. Escriba su “numero de cedula” en el campo Cédula de Identidad. (use los 10 dígitos sin usar ni espacio ni guiones. (Ejemplo: 0105687956).
4. Escriba su “contraseña” en el campo de Contraseña.
5. Ya hechos estos pasos le da al botón “Iniciar Sesión”.

Figura 2: Iniciar Sesión



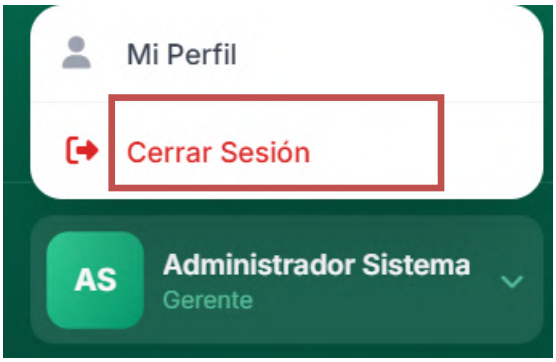
Si los datos que ingresó son correctos, el sistema lo llevará a su pantalla principal según el tipo de usuario. (gerente, tarjetera o conductor).

Notas:

- Si la cédula o contraseña están mal, se visualizará un mensaje que dirá:Las credenciales no son válidas.
- Si su usuario está desactivado, se verá un menaje: Usuario Inactivo.
- Si es la primera vez que entra al sistema o el gerente le restablece su contraseña el sistema le pedirá obligatoriamente que cree una nueva contraseña por motivos de seguridad.

¿Cómo cerrar sesión?

Para cerrar sesión presione el nombre en la parte del menú lateral al final aplasta el nombre de usuario y aplasta en el botón de **“Cerrar Sesión”.**



Capítulo III

Roles de Usuario

Tabla comparativa de permisos en el sistema.

Funcionalidad	Gerente	Tarjetera	Conductor
Ver panel o dashboard	X	X	X
Gestionar Usuario	X		
Gestionar Unidades	X		
Ver Documentos de conductor	X (Solo lectura)	X	
Ver Multas	X (De todos)	X (Solo por punto)	X (Individual)
Cobrar Multas		X	
Ejecutar Sorteo Diario		X	
Confirmar presencia de conductores		X	X (Tarjetera Ausente)
Resgistrar Salida		X	
Aprobar Justificativos		X	
Enviar Justifiactivos			X
Ver GPS en tiempo real	X	X	
Configurar parámetros	X		
Cambio de contraseña	X	X	X
Ver Reportes	X	X (Solo por punto)	

Capítulo IV

Para Gerentes (Administradores)

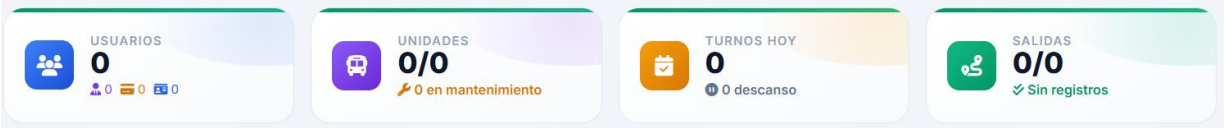
Panel Principal o Dashbord

Cuando un usuario de rol gerente inicia sesión, lo que primero va a visualizar es el "Panel de Control" donde tiene las siguientes funciones:

Tarjetas de Resumen

- Usuarios Activos
- Unidades disponibles
- Turnos del Día
- Salidas en operación

Figura 3: Botones de acceso rápido del administrador (Gerente)



- **Botones de Acceso Rápido:** Botones que llevan a funciones más usadas como creación de nuevo usuario, unidades, ver operaciones y sanciones).

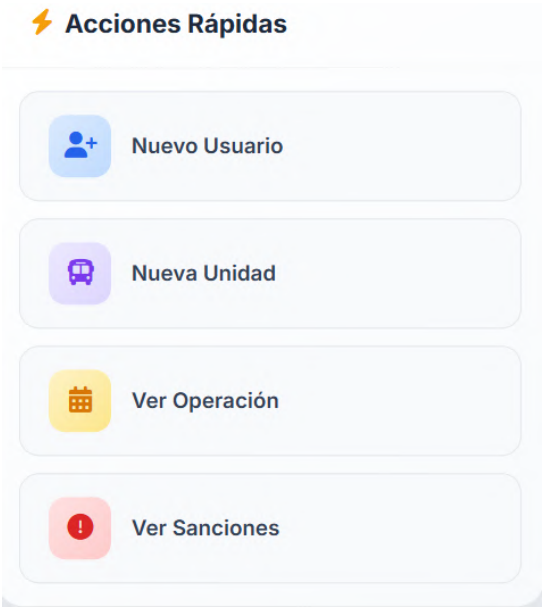
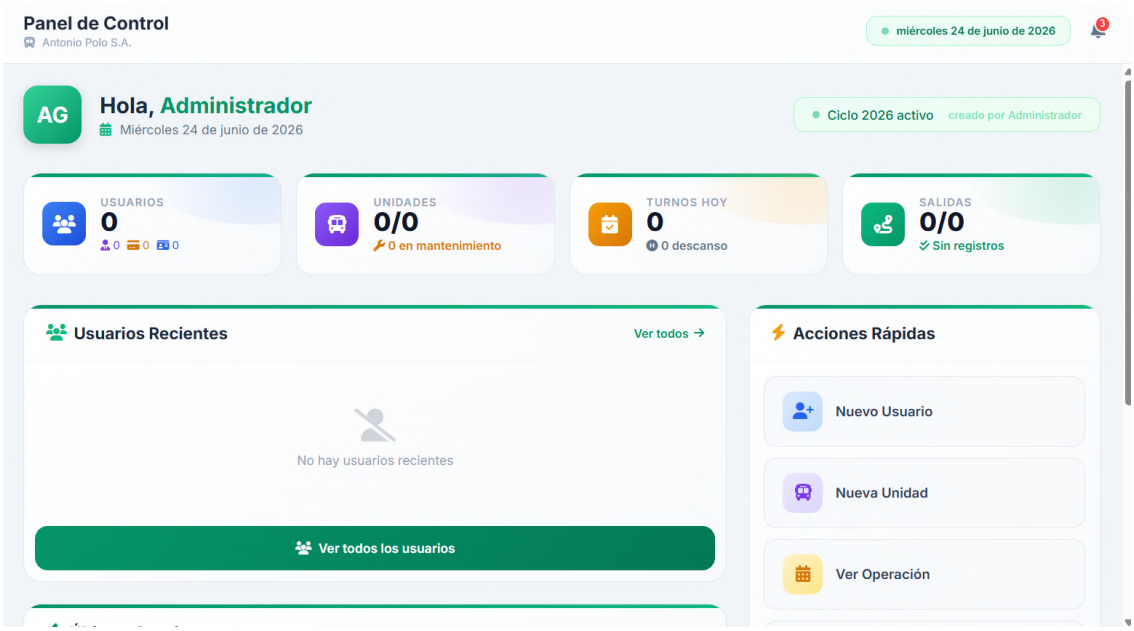
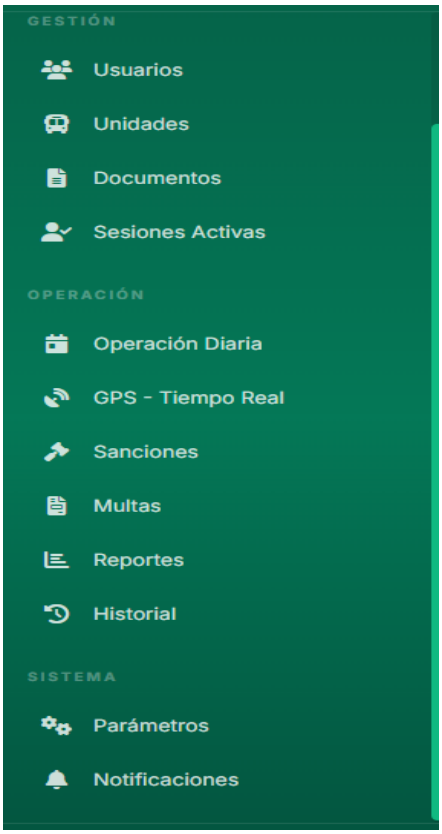


Figura 4: Pantalla principal del administrador



Para acceder a las demás funciones del gerente ir al menú lateral que está a la izquierda del panel principal.

Figura 5: Menú lateral (Gerente)

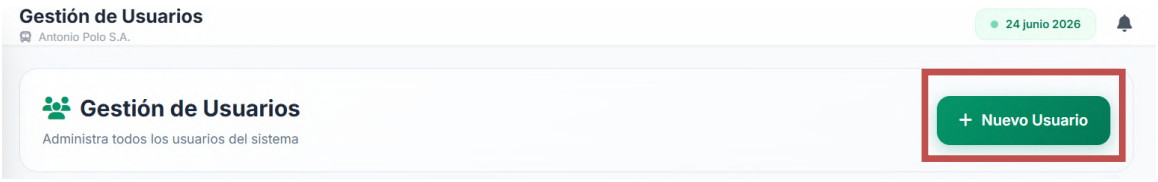


Gestión de usuarios

En el menú lateral presione donde dice “Usuarios” para dirigirse a la sección.

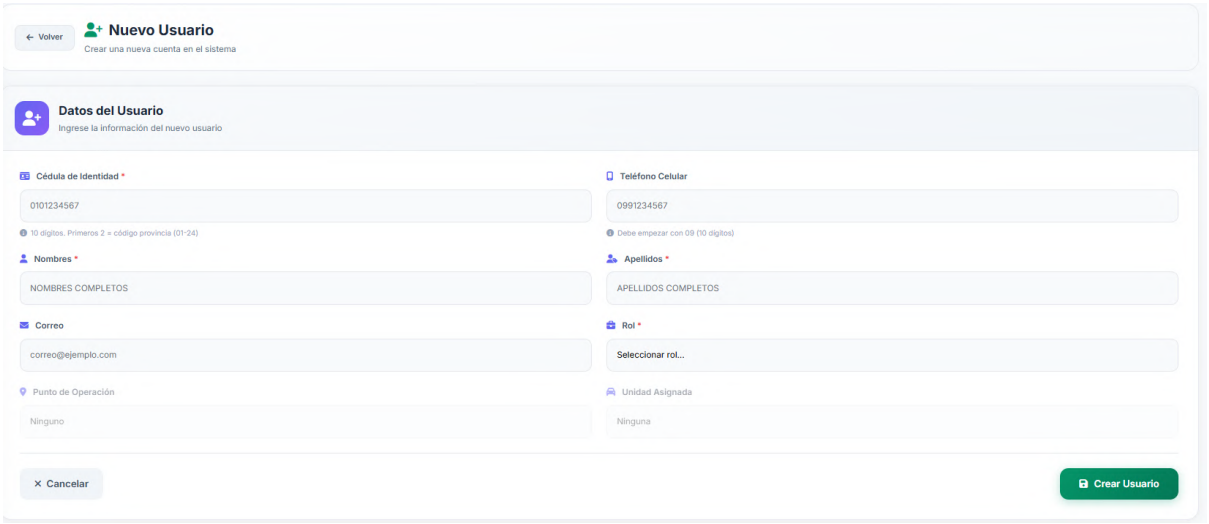


Presione el botón que diga “Nuevo Usuario”.

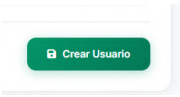


Se le aparecerá un formulario y debe llenar con los datos del nuevo usuario a registrar.

Figura 6: Gestión de usuarios



Una vez completada con la información necesaria darle clic en “**Guardar Usuario”.**



El sistema le dará una contraseña temporal que debe entregarle al usuario para que pueda acceder al sistema.



Modificar un usuario

En la misma sección nos encontramos con la lista de usuarios que se han registrado, para modificar algún campo primeramente se busca al usuario para modificar su información (se puede usar filtros de búsqueda).



Al encontrar al usuario que va a ser modificado presionar en el botón que dice editar es de color verde. Cambia los datos que sean necesarios y se presiona en “Guardar Cambios”

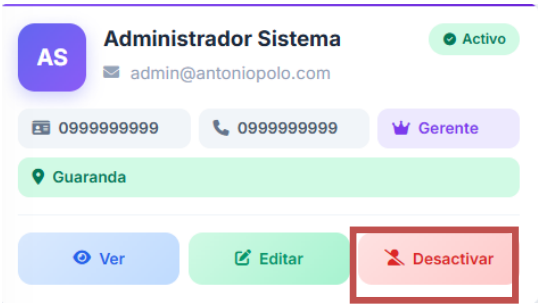
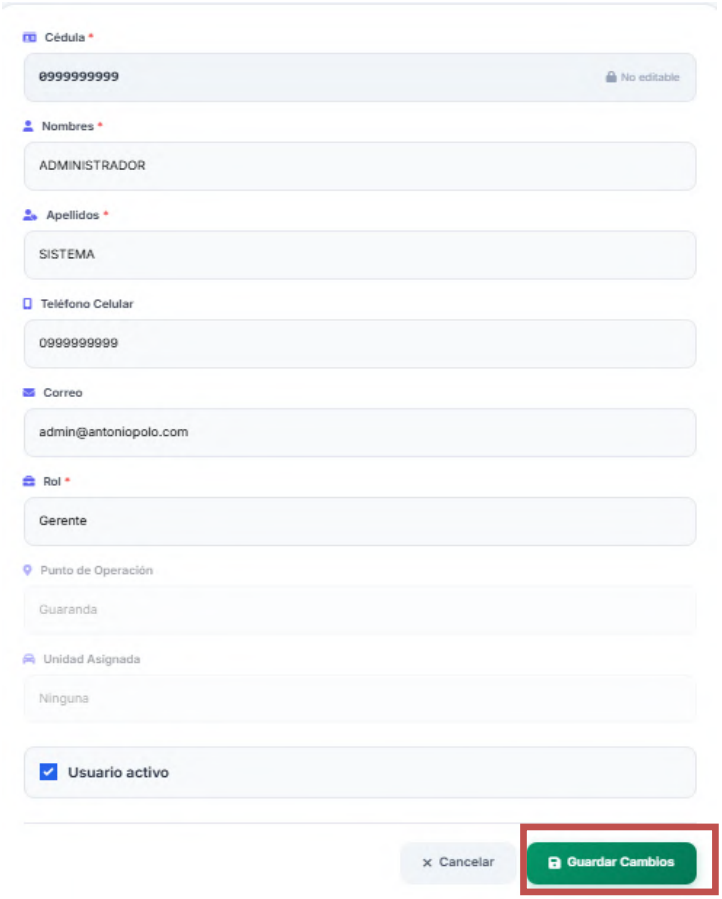
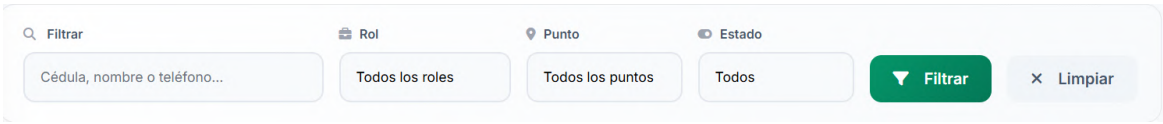


Figura 7: Pantalla Modificar usuario

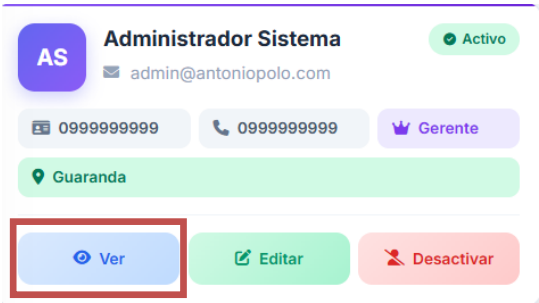


Reseteo de Contraseña

Para el reseteo de la contraseña en caso de haber olvidado, un usuario puede pedir que se le restablezca y el gerente lo hace de la siguiente manera. Busca al usuario que va a ser reseteado su contraseña.

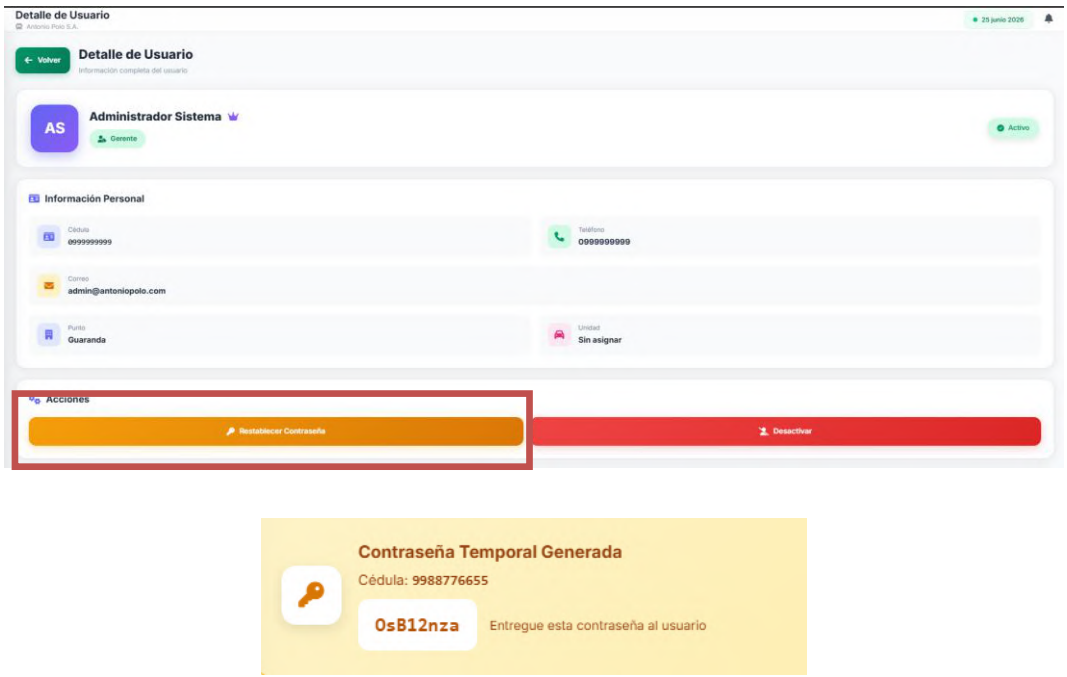


Visualiza la tarjeta con la información y pone en el botón de “Ver”.



Ahí tiene un despliegue de información del usuario y la opción de “Resetear Contraseña” al aplastar en la opción automáticamente le devuelve una contraseña temporal.

Figura 8: Pantalla de restear contraseña y desactivar



Al iniciar sesión el usuario con la contraseña reseteada tiene que cambiar obligatoriamente la contraseña por motivos de seguridad.

Gestión de Unidades

Las unidades tienen un estado que determina si puede actuar en la operación o no:

ESTADO	SIGNIFICADO
Activo	La unidad puede operar con normalidad.
Mantenimiento	La unidad está en un mecánico, entonces no participa en el sorteo de turnos.
Dado de Baja	La unidad ya no está en la compañía, entonces no se puede usar.

Nota: Cuando una unidad se pone en estado de “Mantenimiento”, el sistema lo excluye automáticamente del sorteo diario que hace la tarjetera. El conductor que esté con la unidad no recibirá un turno hasta que vuelva a estar en estado “Activo”.

Registro de una unidad nueva

En la parte del menú lateral identificamos la sección de “Unidades”, presionamos y nos llevará a una sección donde podemos ver tarjetas que contiene la información de las unidades.



Para el registro de la unidad identificamos el botón que dice “Nueva Unidad” y se nos desplegará un formulario que se debe llenar con la información necesaria.

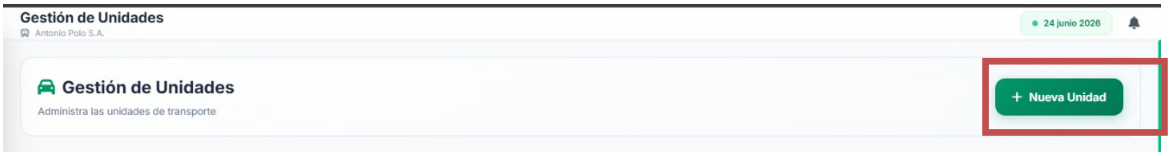
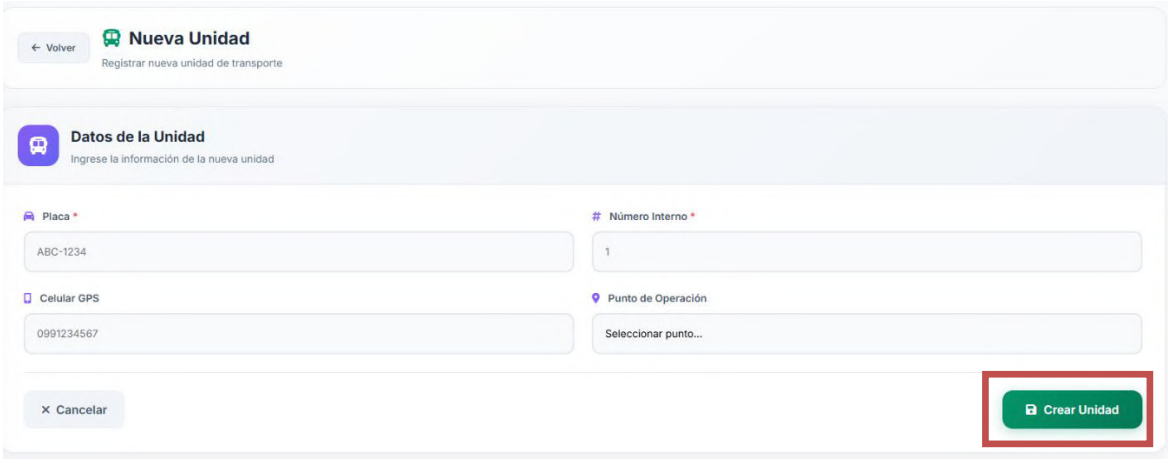


Figura 9: Pantalla de registro de nuevo unidades

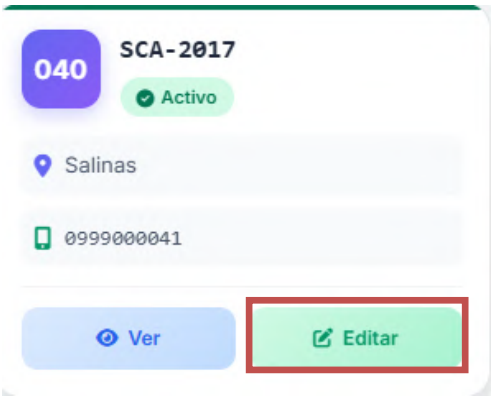


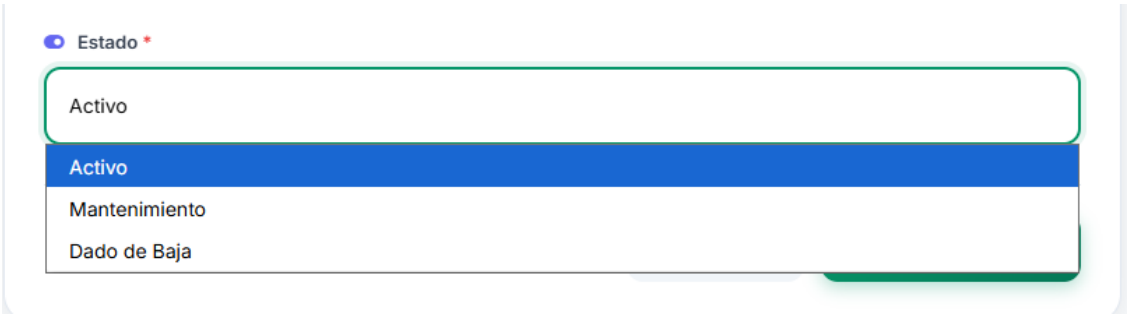
Aplastamos en “Crear Unidad” y se creará la unidad y estará lista para ser asignado con un conductor para su respectiva operación.

Cambiar el estado de la unidad

En la misma sección de “Unidades” hacemos la búsqueda de la unidad que va a ser modificada aplastamos en el botón de “Editar” vemos la información y al final encontramos el estado que podemos elegir entre las 3 opciones:

- Activo
- Mantenimiento
- Dado de Baja





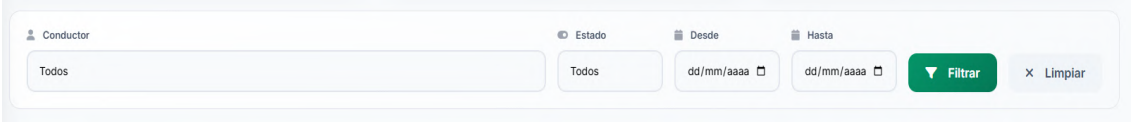
Para reactivar o cambiar el estado a “Activo” simplemente se dirige a la sección y se cambia según lo que corresponda. Al final, solo se guarda y se actualiza el estado de la Unidad.

Multas

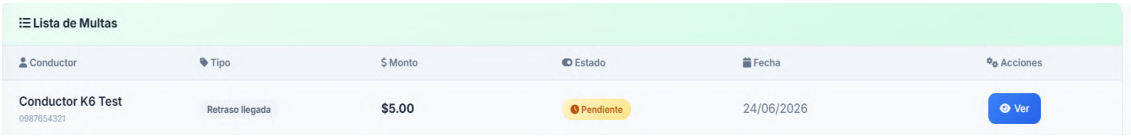
Para ver las multas que se han registrado en el sistema en el menú lateral tenemos una sección para las multas aplicadas.



Las multas también cuentan con filtros para su respectiva búsqueda.



Y también tenemos una lista que despliega todas las multas que existen, con su estado si es Pendiente o Cobrada.

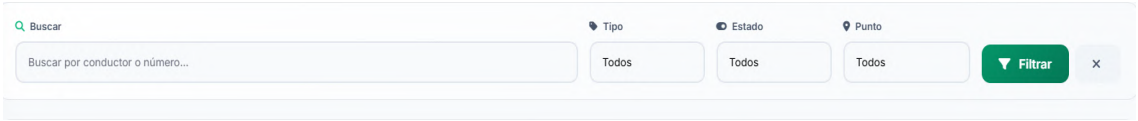


Nota: El gerente nomás puede ver las multas. La creación, cobro y anulación de estas lo realiza la tarjetera en su panel de control.

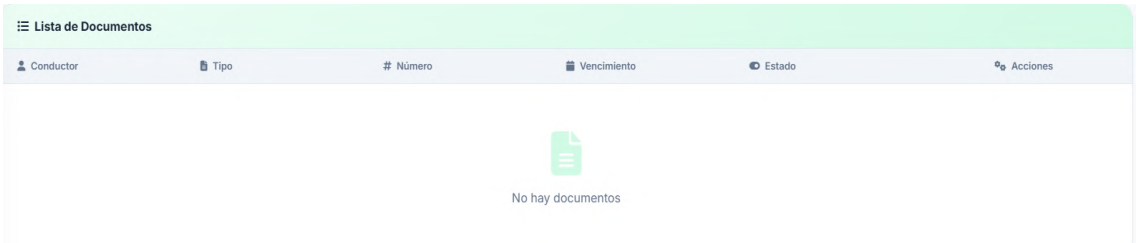
Documentos

Para ver los documentos de los conductores nos dirigimos a la sección de documentos en el menú lateral, donde se puede aplicar búsquedas sea por conductor, tipo, número, etc.





Hay una lista que nos muestra la cantidad de documentos ingresados al sistema.



En la parte de acciones cuando el documento haya vencido se habilitará un botón con el nombre de “Notificar Conductor”, al presionar enviará una notificación de recordatorio al conductor.

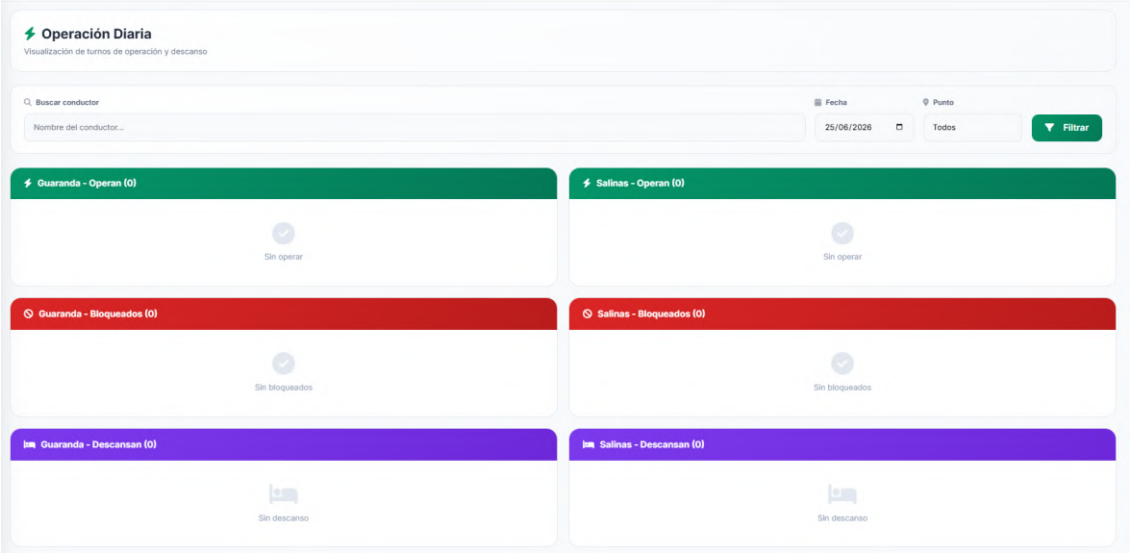
El sistema ya alerta automáticamente al conductor en caso de tener un documento próximo a vencer. Esta función permite al gerente reforzar un recordatorio si al conductor le ha pasado por alto la advertencia.

Turnos Asignados

Al dirigirse a la sección de “Operación Diaria” nos encontraremos con la lista de los conductores que hacen la operación según el día que les haya tocado, también la lista del grupo que descansa y ver aquellos conductores que fueron bloqueados en su día de operación.



Figura 10: Pantalla de operación diaria



Nota: El sorteo de los turnos lo realiza la tarjetera desde su rol.

Reportes

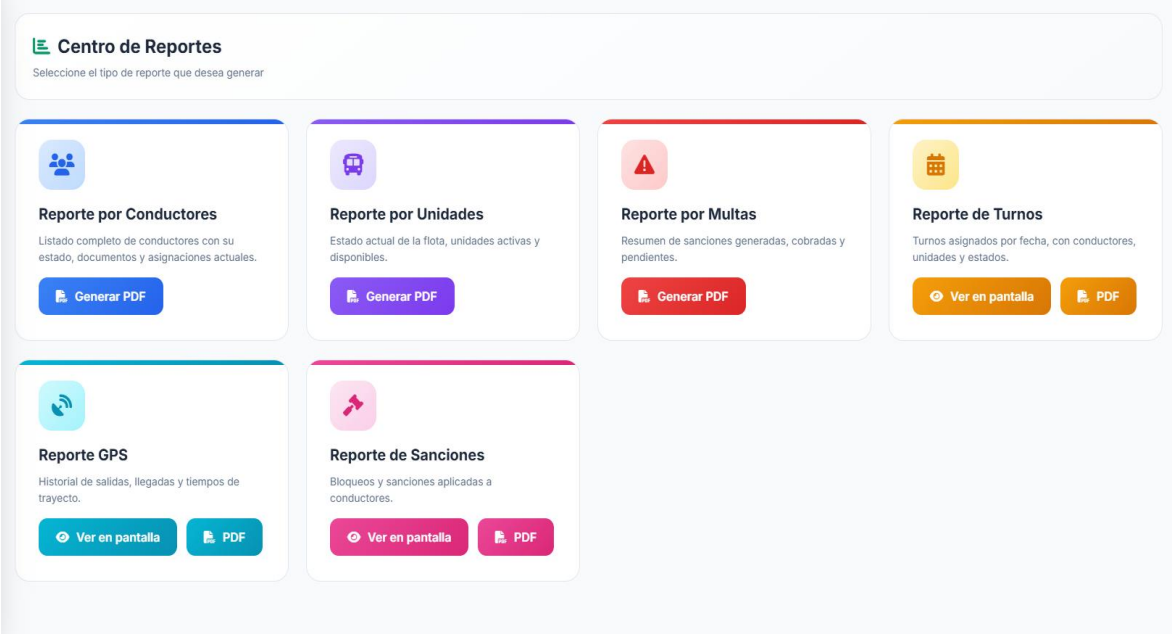
Para tener un buen control sobre las operaciones del sistema el gerente puede generar varios reportes para eso se dirige a la parte de “Reportes” en el menú lateral.



En esa sección podemos ver que podemos generar varios tipos de informes que son:

- **Conductores:** Listados de los conductores activos.
- **Unidades:** Listado de las unidades
- **Multas:** Resumen de las multas por período.
- **Turnos:** Turnos Asignados por fechas.
- **GPS:** Historial de Posiciones
- **Sanciones:** Bloqueos y sanciones aplicadas.

Figura 11: Pantalla de reportes



El reporte se puede ver en pantalla de algunas opciones como se muestra en la imagen y poder imprimirlo o guardarlo en PDF.

GPS y mapa en tiempo real

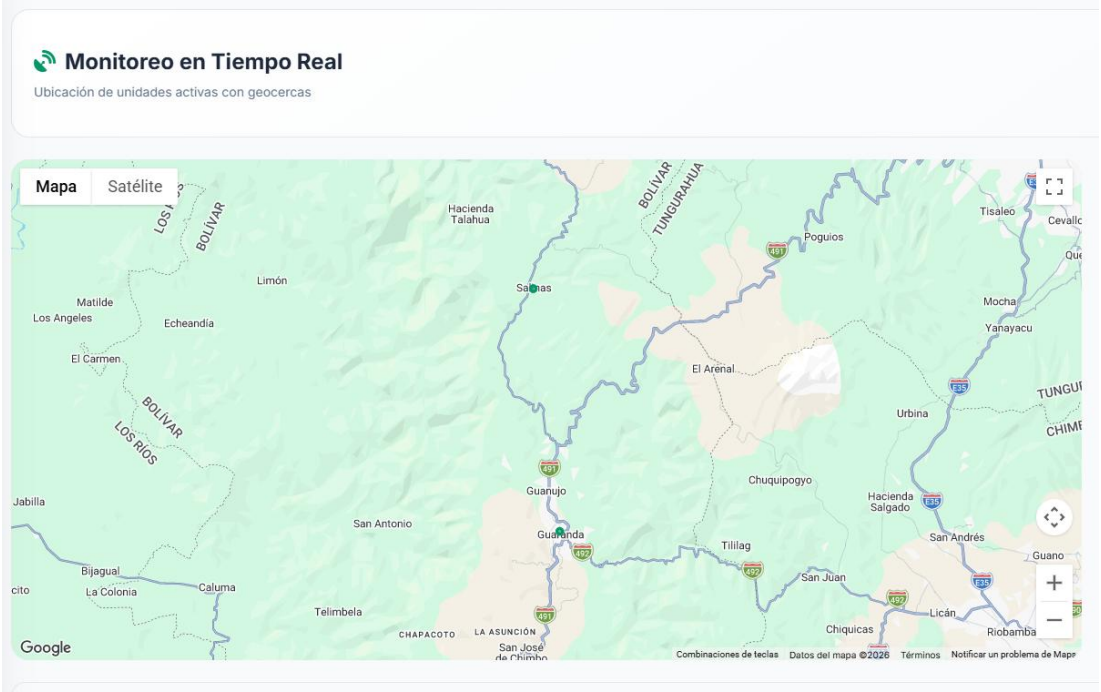
Para visualizar el trayecto de las unidades que están en ruta se dirige a “GPS – Tiempo Real” y va a visualizar un mapa con las unidades activas que están en el punto.

Muestra cada punto la placa de la unidad, conductor, etc.

Se puede aplastar en la unidad para monitorear la ruta que está tomando en el día.



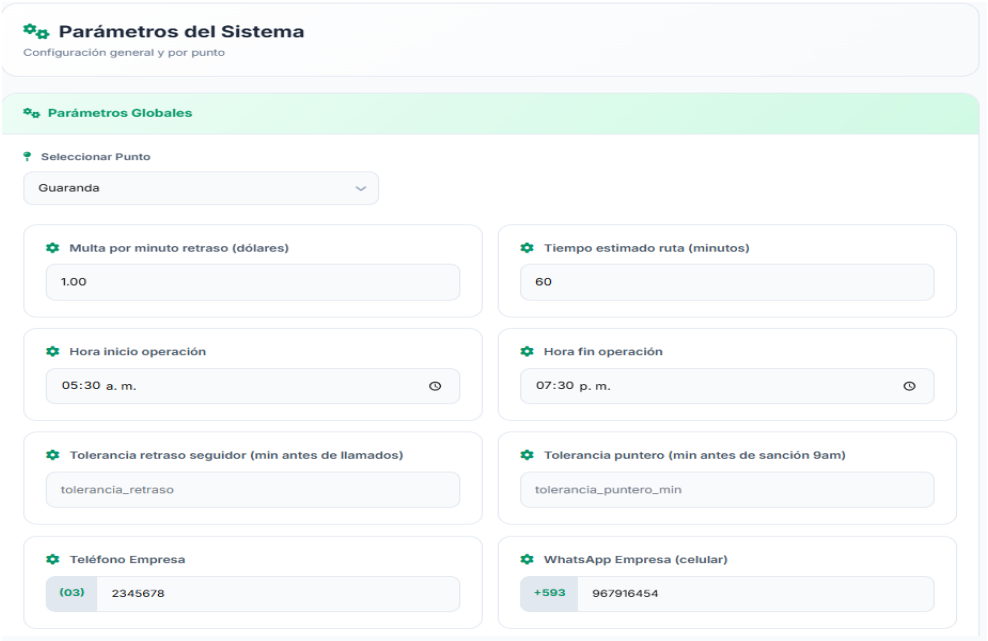
Figura 12: Pantalla de monitoreo en tiempo real



Parámetros del Sistema

Los que realiza esta función es modificar las reglas de negocio tales como: la multa por retraso o el radio de geocerca. Nos vamos a “Parametros” y podemos visualizar los valores a cambiar dentro del sistema.

Figura 13: Pantalla de Parámetros del sistema



En esa parte podemos cambia lo que es:

- Valor de la multa por minuto de retraso.
- Tiempo estimado de la ruta.
- Teléfono y WhatsApp de la compañía.
- Horario de operación.
- Tolerancias permitidas.
- Radio de geocerca.
- Coordenadas

Los radios de las geocercas y coordenadas de los puntos de operación se hacen por punto dependiendo de cómo sea la lógica de la compañía.

Al finalizar se termina con **“Guardad Cambios”.**



Capítulo V

Para Tarjetera

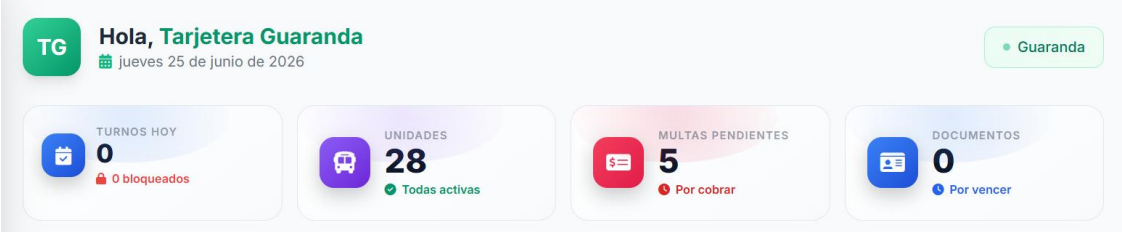
Panel Principal o Dashboard

Cuando un usuario de rol Tarjetera inicia sesión, observa su Panel de control con:

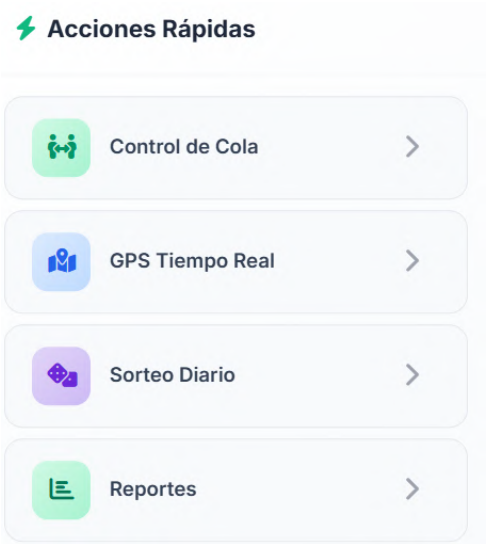
Resumen del día

- Cuantos conductores han llegado, salidas registradas, etc.

Figura 14: Pantalla de acceso rápido (Tarjetera)



Botones de acceso rápido

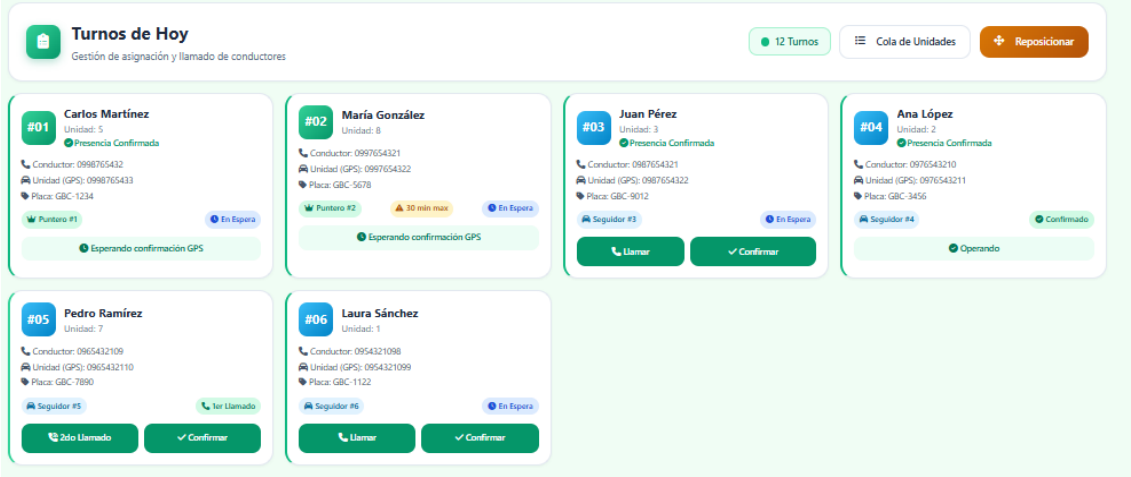


Turnos del Día

Al aplastar en la sección de “Turnos del Día” en el menú de tarjetera podemos ver una lista de conductores con el turno asignado.

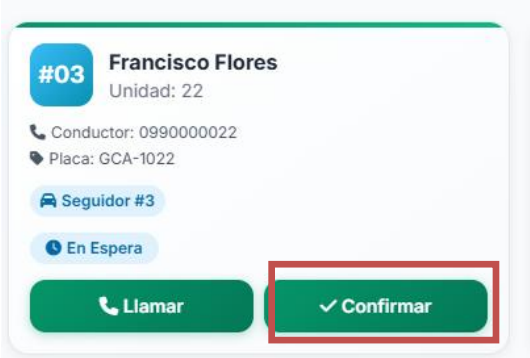
Cada turno muestra información necesaria.

Figura 15: Pantalla de turnos del día (Tarjetera)



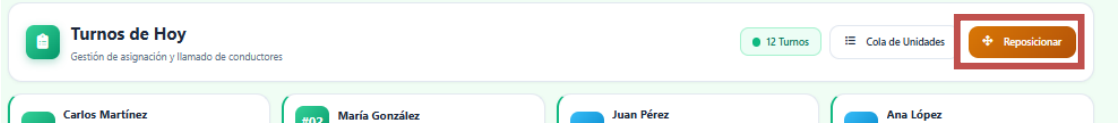
Confirmar presencia de conductor

Cuando el conductor arriba al punto donde va a iniciar sesión en "Turnos del Día" buscamos al conductor en la lista y pulsamos el botón de "Confirmar", el sistema marca que el conductor está presente en su punto.



Reposicionar Conductores

Después que un conductor completa un viaje y llega al destino, la tarjetera lo puede reposicionar con el botón **"Reposicionar"** con esa opción reasigna a los conductores que no han llegado o no están confirmado o que hay algunos ya disponibles en la cola de la salida.



¿Cuándo se usa?

Al finalizar una carrera, si salió por unidad llena o cuando conductores salieron y ya llegaron y otros conductores aún no.

Cuando un conductor no respondió a los dos llamados y fue bloqueado y se necesita ajustar posiciones de los turnos.

Para reordenar la cola manualmente según la operación del día.

El sistema ajustará automáticamente las posiciones y enviará una notificación al conductor informándole su nueva posición asignada.



Nota: Solo puede reposicionar conductores que no hayan confirmado presencia aún. Los que ya están confirmados mantienen su posición.

Multas del punto

En el menú lateral encontramos la parte de “Multas” y al presionar nos dirigimos a la sección donde contine todas las multas que se han generado.



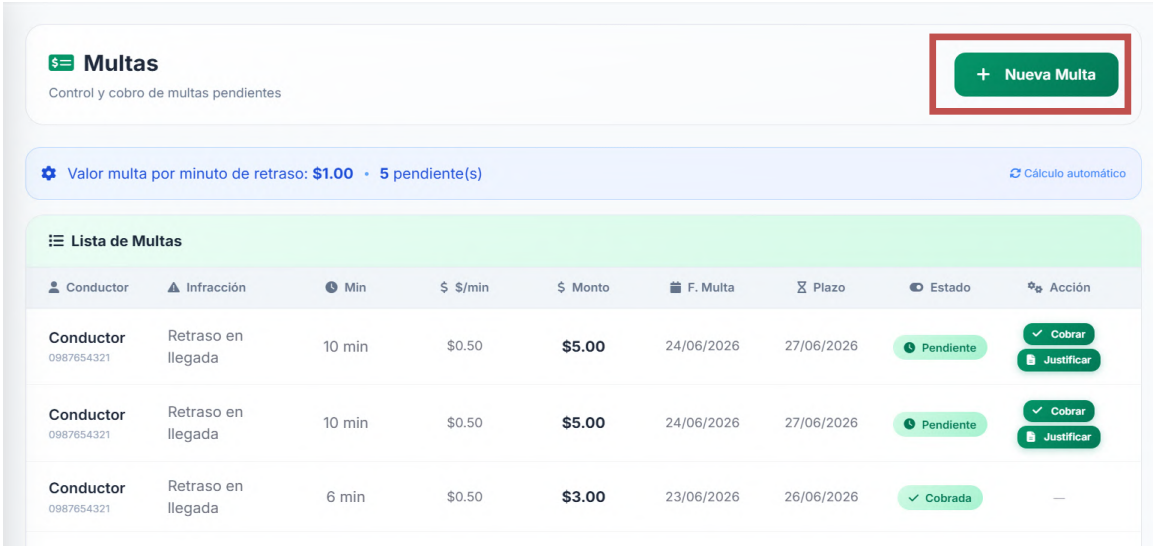
Ahí podemos ver una lista con las multas y con los conductores que han sido multados.

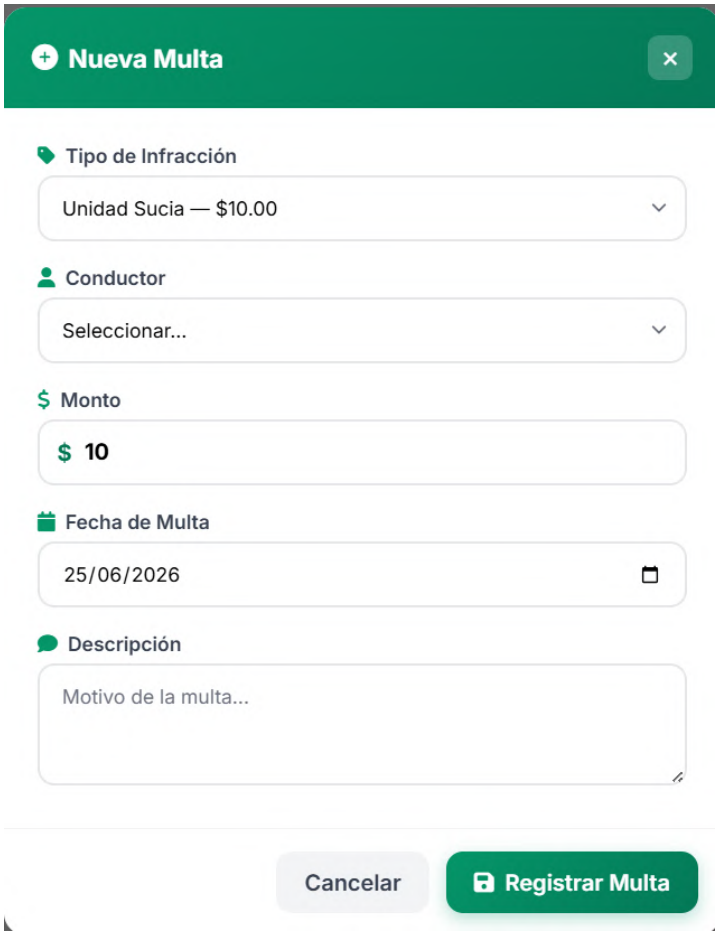
Podemos también en esa sección crear una nueva multa de tipo: Unidad Sucia o Uniforme Incompleto con valores editables.

Calcula las multas automáticamente por retraso de minutos.

Se determina el cobro de multas y su respectiva anulación dependiendo si la justificación es validada.

Figura 16: Pantalla de Multas (Tarjetera)

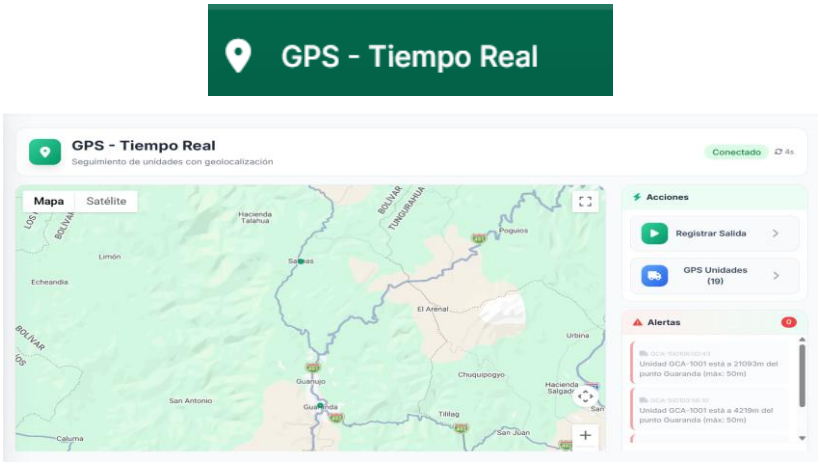




GPS en tiempo real

Para ver la geolocalización de las unidades que están en ruta se dirige a la opción de “GPS-Tiempo Real”, en esa sección podrá encontrar el mapa con las unidades activas en los puntos y alertas si una unidad sale de la ruta, esas alertas se marcan como leídas.

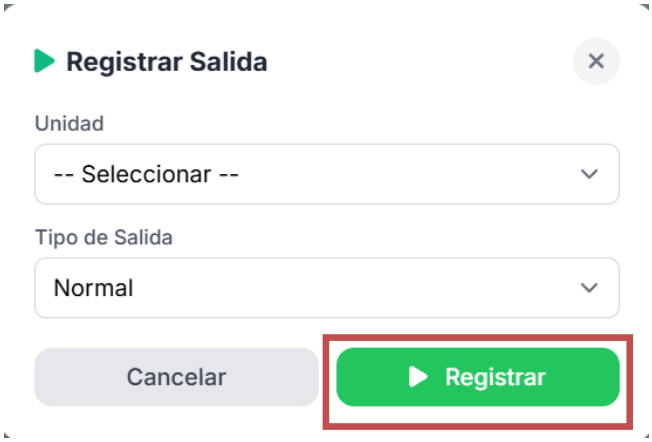
Figura 17: Pantalla de GPS en tiempo real



Registrar Salida.

En la misma sección de, nos encontramos con un botón que se llama “Registrar Salida”, al pulsar nos muestra:

Figura 18: Pantalla de registrar salida



Se puede seleccionar tipos de salida como: Normal y Unidad Llena según sea el caso. Para que guarde el sistema basta con aplastar en el botón que dice “Registrar”.

Sorteo Diario

La operación depende solamente de ese sorteo que se realiza a diario para ejecutar la acción diríjase a la parte donde dice “Sorteo Diario”

Se pueden crear los grupos de descanso que esos son los que se mantienen en un ciclo anual.

Se pueden volver a crear con la acción de “Recrear”, se elimina los grupos y los genera de nuevo con un mensaje de confirmación.

El sorteo se puede limpiar si desea rehacer los turnos y randomizar los días para que después de la semana de operación los grupos no descansen en el mismo día.

Figura 19: Pantalla de sorteo diario



Al ejecutar en el botón de “Ejecutar Sorteo”, el sistema asigna posiciones a los conductores automáticamente.



Nota: Las unidades que están en estado Mantenimiento”, no se asignan a ningún conductor durante el sorteo. Si un conductor tenía asignado una unidad que ahora está en mantenimiento, aparecerá sin unidad en su turno.

Justificativos

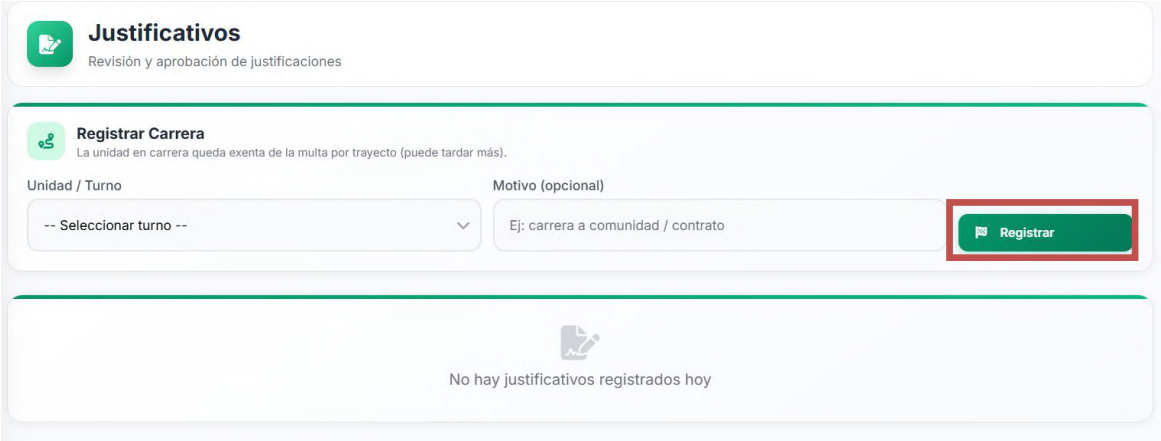
En la sección, veremos las solicitudes de justificaciones que un conductor a mandando en estado pendiente.

La tarjetera puede Aprobar y Rechazar.



En esa sección también puede justificar una excepción de una salida de unidad de tipo “Carrera” con eso la unidad se exenta de cobros de multas y el tiempo de trayecto no se toma en cuenta.

Figura 20: Pantalla de justificativos

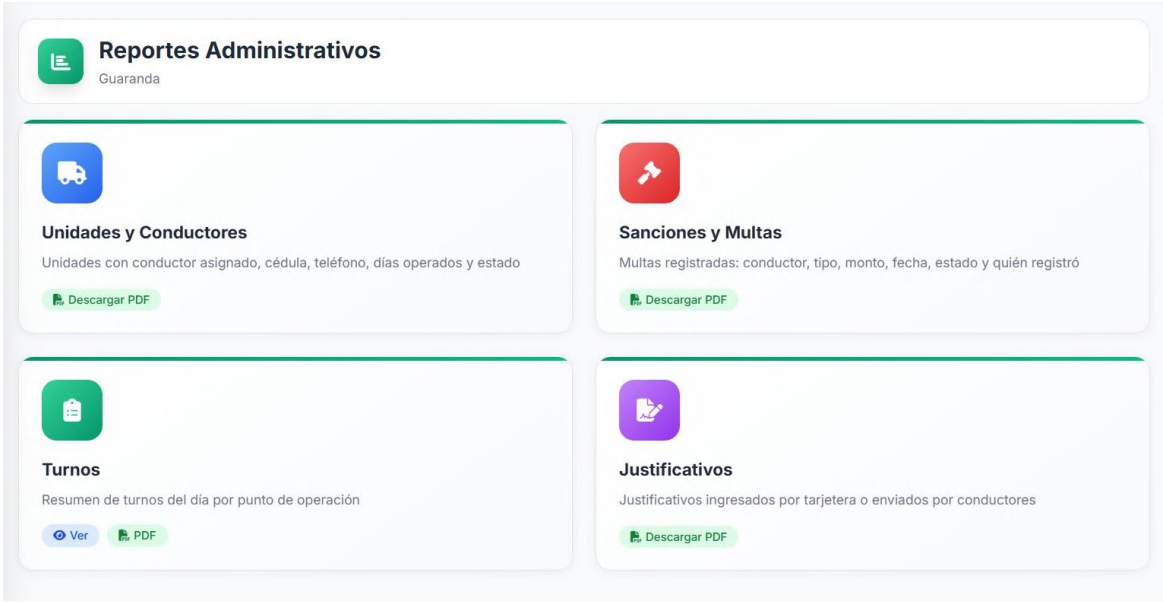


Reportes

Al igual como en gerente, la tarjetera puede crear sus reportes según las reglas de negocio lo establecen, se dirige a la sección de reportes y podemos exportar en PDF.

Figura 21: Pantalla Principal de reportes




Reportes Administrativos
Guaranda
Unidades y Conductores
Unidades con conductor asignado, cédula, teléfono, días operados y estado
Descargar PDF
Sanciones y Multas
Multas registradas: conductor, tipo, monto, fecha, estado y quién registró
Descargar PDF
Turnos
Resumen de turnos del día por punto de operación
Ver
PDF
Justificativos
Justificativos ingresados por tarjetera o enviados por conductores
Descargar PDF

Capítulo VI

Para Conductores

Pantalla de Inicio

- Al iniciar sesión con el rol de conductor, ve su pantalla principal con:
- Saludo personalizado: "Hola, [su nombre]"
- Fecha actual
- Su turno de hoy: Punto y unidad asignada
- Estado del GPS: Si está conectado o no
- Documentos: Matrícula y licencia vigentes.

Figura 22: Pantalla del conductor

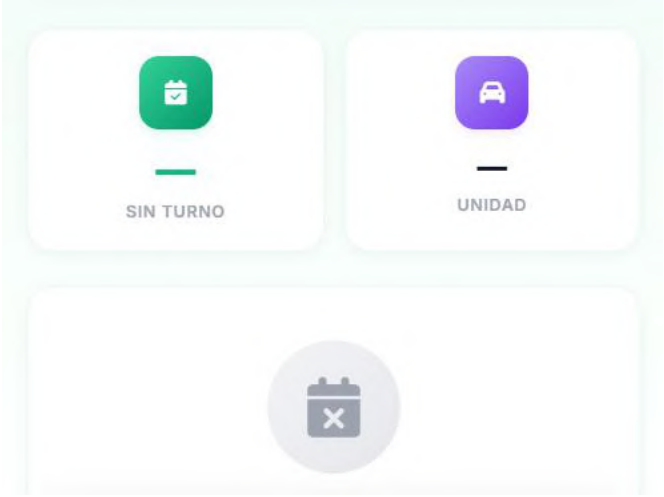


Mi turno del Día

En la pantalla de inicio podemos ver la posición que se le asignó en el sorteo que se hizo previamente, la unidad que está usando para la operación y el punto de operación (Guaranda - Salinas).

Y un botón para “Confirmar Presencia” cuando llegue a la parada donde se va a dar el inicio de la operación.

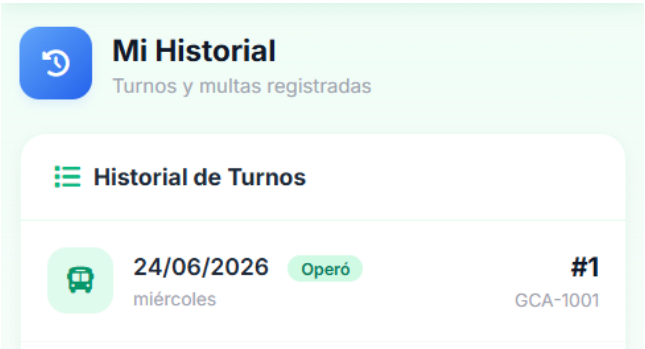
Figura 23: Pantalla de visualización de turnos



Historial y Mis Multas

Para ver las multas que tiene el conductor se dirige a la parte de “Historial” nos encontramos en una lista donde detalla los turnos que ha operado en días previos y debajo se encuentra la parte de “Multas” se detallará todas las multas que tenga hasta la fecha, pendientes, anuladas o cobradas.

Figura 24: Pantalla del historial turnos y multas





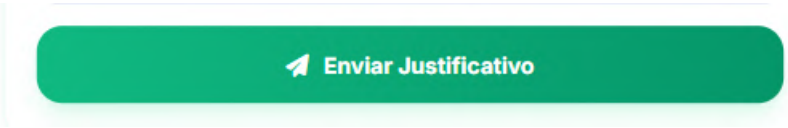
Justificativos

Si desea que el sistema o la tarjetera no le bloquee su turno o sancione, el conductor puede mandar una solicitud de justificación para su respectiva revisión, primeramente, se dirige a la sección de “Justificativos”

Figura 25: Pantalla de justificativo (Conductor)



En el formulario se llena con la información necesaria para que se vaya a revisión luego se envía con el botón de “Enviar Justificativos”, se envía a la tarjetera y ella lo aceptará o rechazará.



Notificaciones

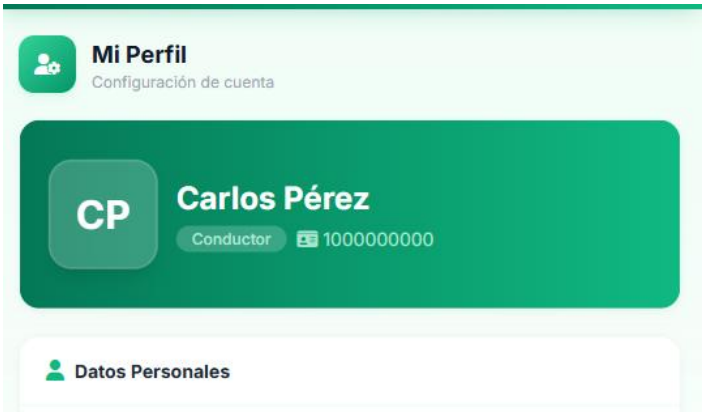
Para ver las notificaciones que se le ha enviado al conductor se puede dirigir a “Notificaciones” donde verá una lista de las notificaciones que se han generado ya sea por el sistema o por el gerente manualmente.

Al presionar una notificación esta se marca como leída.



Perfil y cambio de contraseña

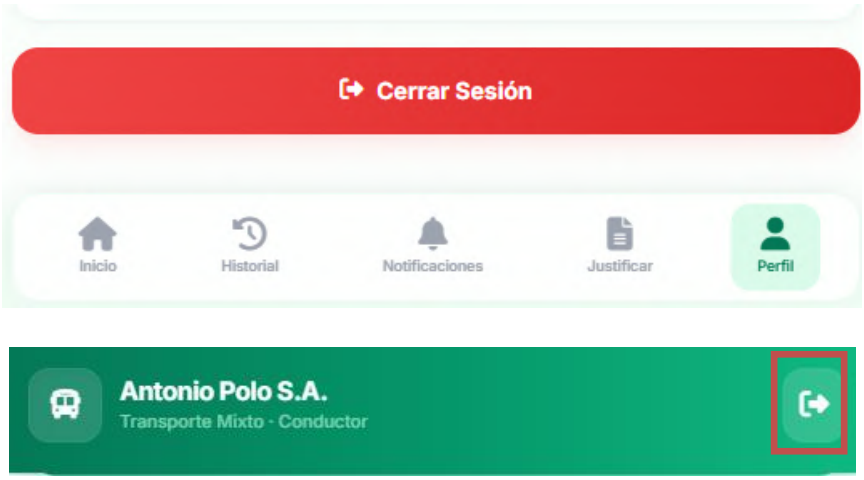
En esta sección se pueden ver los datos personales del conductor donde puede editar algún campo si en caso cambia la información a futuro.



También hay la sección de poder cambiar su contraseña con controles para asegurar integridad. Para terminar esa acción solamente le da al botón de "Actualizar Contraseña".



Y podemos hacer el cierre de sesión de dos maneras una es en la sección de “Perfil” y otra con un botón que hace la acción rápidamente.



Capítulo VII

Seguridad

Para la protección de la cuenta se le sugiere lo siguiente:

- **No comparta su contraseña** con nadie, ni siquiera con compañeros de trabajo
- **Cierre sesión siempre** al terminar de usar el sistema, especialmente en dispositivos compartidos
- **Cambie su contraseña periódicamente** por seguridad

Sesión cerrada por inactividad

Para más seguridad el sistema verifica si el usuario permanece más de 30 minutos inactivo sin hacer alguna acción. Al volver al sitio pedirá que inicie sesión de nuevo.

Restablecimiento de la contraseña.

Si por razones usted olvida o pierde su contraseña:

- No puede recuperarla por correo ni por SMS.
- Debe pedir al gerente que le restablezca la contraseña.
- El gerente le brindará una contraseña temporal.
- Al iniciar sesión con esa contraseña, el sistema le pedirá obligatoriamente que cree una nueva contraseña.

Contraseñas Temporales

La primera vez que use el sistema se le dará una contraseña temporal que ha generado el gerente al hacer un nuevo registro de un usuario.

También debe recibir la contraseña temporal cuando se le restablece la clave de acceso.

Se debe cambiar obligatoriamente la contraseña a lo que inicia sesión.

ANEXO I.

Glosario de términos

Ciclo: Período anual de operación. Cada año se crea uno nuevo.

Ciclo activo: El ciclo que está vigente y en uso actualmente.

Conductor: Usuario del sistema que conduce la unidad.

Dashboard: Pantalla principal o panel de control.

Geocerca: Zona geográfica virtual. Cuando el conductor entra en esa zona el sistema lo detecta.

GPS: Sistema de posicionamiento global que permite saber dónde está cada unidad.

Grupo de Descanso: Conductores que no operan ese día.

Justificativo: Explicación escrita que un conductor presenta cuando no pudo cumplir con su turno o llegó tarde.

Multa: Sanción económica que se aplica a un conductor por retraso o falta.

Puntero: Los dos primeros conductores en la lista de turnos. Pueden operar todo el día; la sanción de 9:00 am aplica solo por no presentarse.

Salida: Registro de cuando un conductor inicia un viaje.

Sorteo: Asignación de posiciones a conductores para operación diaria.

Tarjetera: Persona encargada de la operación del día en los puntos de operación.

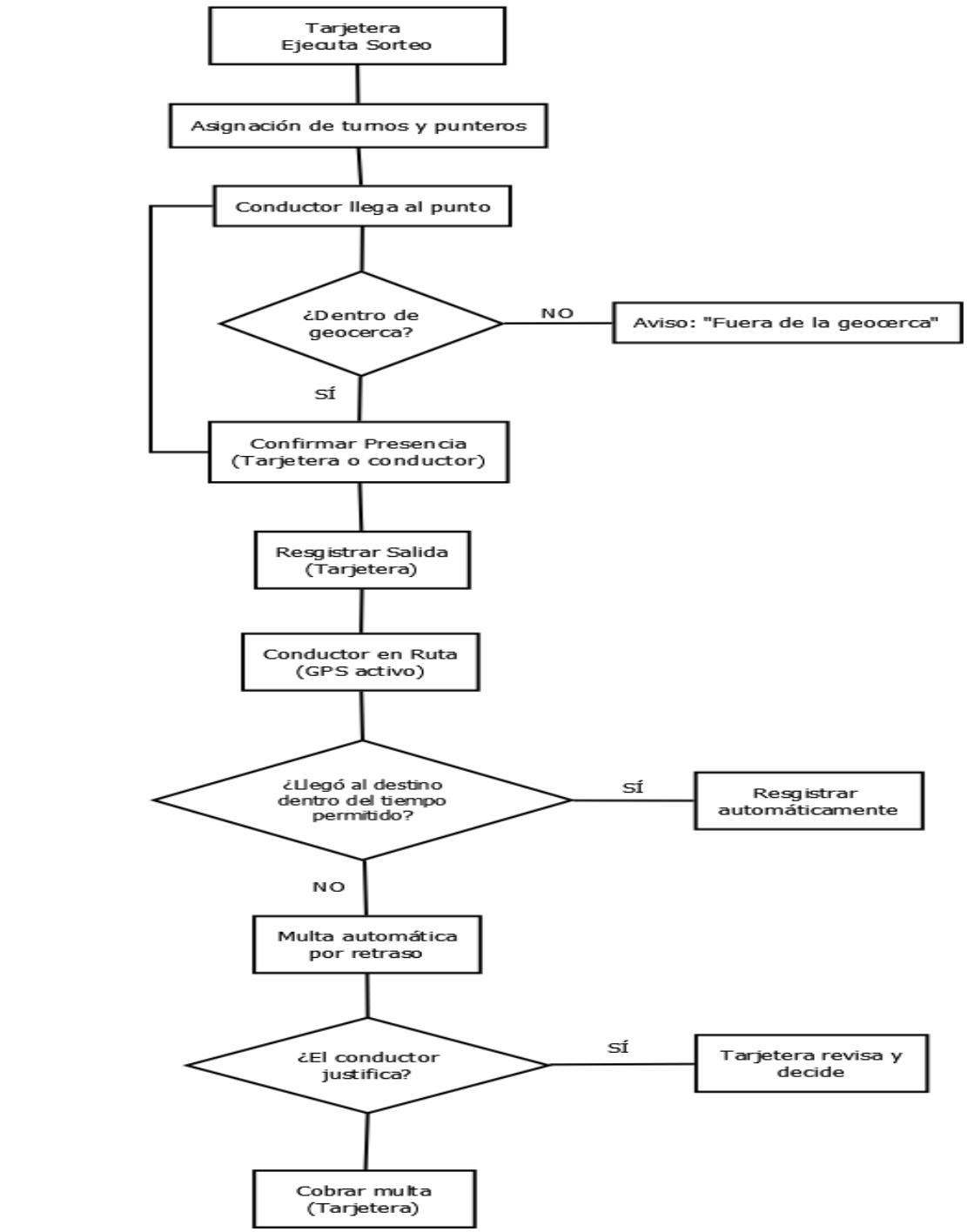
Turno: Asigna a cada conductor a una unidad y posición para un día en específico

Unidad: Cada camioneta de transporte que tiene la compañía.

ANEXO II.

Diagrama de Flujo de la Operación

Figura 26: Diagrama de flujo de operación



Anexo 6.

Certificación de Anti-plagio

ING. DARWIN CARRION BUENAÑO EN CALIDAD DE DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR,

CERTIFICA

Que el trabajo de integración curricular denominado "**SISTEMA WEB PARA LA GESTIÓN OPERATIVA Y CONTROL AUTOMATIZADO DEL ARRIBO Y DESPACHO DE UNIDADES MEDIANTE GEOLOCALIZACIÓN EN PUNTOS ESTABLECIDOS EN LA COMPAÑÍA DE TRANSPORTE ANTOÑO POLO S.A**", presentado por los Srs. Renan Marcial Sigcha Punina y Steven Alexander Agualongo Morocho estudiantes de la **carrera de Software** pasó el análisis de coincidencia no accidental en la herramienta COMPILATIO, reflejando un **porcentaje de similitud del 7%**, como se puede evidenciar en el documento adjunto.

Guaranda, 30 de junio del 2026

Atentamente,

Darwin Paul Carrión Buenaño
Director

Certificado de análisis
Compilatio Magister+ | UNIVERSIDAD ESTATAL DE BOLIVAR (UEB)

TESIS SIGCHA AGUALONGO

ID 4991e5e1333eedcb396c2f46456d94e65d616e6c

7%
Textos sospechosos

Nombre del fichero TESIS SIGCHA AGUALONGO.txt
Tamaño del archivo original : 15,27 MB
Número de palabras : 28 822

Depositante : DARWIN PAUL CARRION BUENAVO
Fecha de depósito : 29 de junio de 2026
Tipo de carga : interface
fecha de fin de análisis : 29 de junio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes — 7%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA — 7%

Textos estilísticamente próximos a un texto generado por una IA.
Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.



Idiomas no reconocidos — 13%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.
Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.



No incluido en el porcentaje de textos sospechosos :

Textos entre comillas — 1%

Pasajes entre comillas, a menudo indicativos de una cita.

Fuentes de similitudes (sección 2/2)

Similitudes 7%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Fuente principal detectada

N°		Descripciones	Similitudes	Ubicaciones
1		usgn.espe.edu.ec usgn.espe.edu.ec/wp-content/uploads/2020/07/LEY-INGENIOS.pdf	3%	
2		sayce.com.ec sayce.com.ec/wp-content/uploads/2025/04/CODIGO-ORGANICO-DE-LA-EC...	3%	
3		repositorio.uca.edu.ar repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarr...	1%	
4		Documento de otro usuario #a54ac7 Viene de de otro grupo	<1%	
5		Código Orgánico de la Economía Social de los Conocimientos... www.ecolex.org/details/legislation/codigo-organico-de-la-economia-social...	<1%	
6		¿ Qué medidas deben tomar los Prestadores de Servicios de... www.ecucert.gob.ec/que-medidas-deben-tomar-los-prestadores-de-servi...	<1%	
7		Datos generales sobre la propiedad intelectual en la legislación... es.slideshare.net/slideshow/datos-generales-sobre-la-propiedad-intelect...	<1%	
8		Artículo 35 UE Reglamento general de protección de datos.... www.privacy-regulation.eu/es/35.htm	<1%	
9		Sistema de capacitación jurídica para el mejoramiento de las... repositorio.upse.edu.ec/jspui/bitstream/46000/590/3/TESIS%20AB.%20jO...	<1%	
11		portal.compraspublicas.gob.ec portal.compraspublicas.gob.ec/sercop/wp-content/uploads/2018/12/serc...	<1%	

Fuente mencionada (sin similitudes detectadas)

N°	Descripciones
1	https://maps.googleapis.com/maps/api/staticmap?center={$lat
2	https://www.google.com/maps/embed/v1/view?center={$lat
3	https://www.google.com/maps/embed/v1/directions?origin={$originLat
4	https://doi.org/10.5281/zenodo.15194897
5	https://dspace.ups.edu.ec/bitstream/123456789/31488/1/UPS-CT012398.pdf

N°	Descripciones
6	http://hdl.handle.net/10757/657828
7	https://repositorio.utmachala.edu.ec/server/api/core/bitstreams/819e60b3-c1a7-4ae8-9b60-4ed8c959f1d8...
8	https://www.campusmvp.es/recursos/post/Desar
9	https://repositorio.puce.edu.ec/server/api/core/bitstreams/8f5d1ef4-5436-4384-8648-078da3830d1f/conte...

Anexo 7.

Evidencia de conversaciones con director, pares académicos y personal de la compañía







