

Sistema Administrativo de Transporte Terrestre
SATT

Corporación Universitaria Remington.
Facultad de Ingenierías.
Ingeniería de Sistemas.

Diego Esteban Cardona Bedoya.
Gloria Amparo Lora Patiño.
Proyecto de grado.
2025.

TRABAJO DE GRADO

2

Agradecimientos

Infinitos agradecimientos a nuestro Dios y creador que nos ilumina cada día para construir herramientas útiles para otras personas. Por permitirnos tener espíritu de vida y salud para llevar a cabo el proyecto de grado.

También agradecemos a nuestros docentes que nos dieron la formación técnica para llevar a cabo un proyecto del mundo real. Al profesor Alfonso Morelo a quien le aprendí demasiado de arquitectura de software, a la profesora Marta Jiménez que nos hacía leer todos los módulos juiciosamente y sus evaluaciones eran encaminadas a saber si habíamos estudiado o no los temas pero que en las exposiciones se desarrollaba mucho esas habilidades blandas. Al profesor José Herney González Ardila, abogado que nos enseñó sobre la administración de personal, pero que en su clase lo bombardeábamos de preguntas de todo tipo porque al final el ingeniero debe formarse integralmente y nos dimos cuenta que no sabíamos nada de la parte legal. Gracias profe por su paciencia y resolernos tantas inquietudes en sus clases.

Agradecimientos para nuestra alma mater: Corporación Universitaria Uniremington por recibirnos, formarnos y evaluar de que estamos hechos.

También muchas gracias a la profe Gloria Amparo Lora Patiño, mi asesora de proyecto por ayudarme a construir una documentación mas clara y concisa. Reconozco que estuve por desfallecer con tanta exigencia de su parte, pero ahora veo que valió la pena el esfuerzo.

Al equipo de SOTRAGOLFLO LTDA la empresa que creyó en mí e invirtió en dinero, tiempo y espacio para que este proyecto fuera posible.

TRABAJO DE GRADO

3

Tabla de Contenidos

Contenido

Introducción	9
Marco Teórico.....	11
Introducción al transporte público en Colombia.....	11
Modalidades de transporte	12
Planteamiento del problema.....	14
Objetivos específicos	16
Metodología de Desarrollo	17
Fase de Análisis	17
Metodología de investigación.....	17
Requisitos Funcionales	27
Análisis Arquitectónico	35
Requisitos No Funcionales	35
Restricciones.....	37
Casos de Uso.....	38
Fase de diseño.....	128
Decisiones Arquitectónicas (ADR).....	128
Modelo de Arquitectura	132
Modelo de Datos.....	134
Wireframes.....	146
Fase de Desarrollo.....	149
Herramientas	149
Estrategia y Bibliotecas.....	150
Patrón Aplicado	152
Estructura Principal de Carpetas.....	164
Estrategia de Pruebas	167
Algunas Capturas de Pantalla	169
Métricas de código.....	174
Metodología de Implementación	179
Resultados y Discusión.....	186
Conclusiones.....	191
Referencias.....	194

TRABAJO DE GRADO

4

Lista de tablas

Tabla 1. Secciones Funcionales	27
Tabla 2. Requisitos Funcionales	28
Tabla 3. Drivers Arquitectónicos	35
Tabla 4. Requisitos no funcionales	36
Tabla 5. Tablas de Control de Acceso	134
Tabla 6. Tablas de control de Empleados	138
Tabla 7. Tablas de control de Vehículos.....	140
Tabla 8. Tablas de Revisiones y Logistica.....	142
Tabla 9. Tablas de Tesorería y Cartera	144
Tabla 10. Herramientas para el Desarrollo	149
Tabla 11. Bibliotecas usadas para el desarrollo	150
Tabla 12. Estructura de Carpetas de SATT.	165
Tabla 13. Segmentación de Red y Distribución de VLANS	184

TRABAJO DE GRADO

5

Lista de figuras

Figura 1. Casos de Uso de Control de Acceso.....	38
Figura 2. Casos de uso de control de empleados	44
Figura 3. Casos de uso de control de vehiculos	63
Figura 4. Casos de uso de Revisiones y Logística	93
Figura 5. Casos de uso de Tesorería y Cartera.....	106
Figura 6. Casos de uso de control administrativo	123
Figura 7. Diseño de arquitectura de SATT	132
Figura 8. Wireframe de formulario de pagos.....	146
Figura 9. Wireframe del formulario de despachos.....	148
Figura 10. Arquitectura tradicional API REST.....	152
Figura 11. Arquitectura Server Action de Nextjs	153
Figura 12. Arquitectura de Despliegue	179

TRABAJO DE GRADO

6

Lista de Ilustraciones

Ilustración 1. Foto de formato de autorización	18
Ilustración 2. Planilla de Control Interno de Recaudo	19
Ilustración 3. Copia de Comprobante de Ingreso.....	20
Ilustración 4. Archivo Físico Central.....	21
Ilustración 5. Archivadores Medianos Distribuidos	22
Ilustración 6. Plantilla en Blanco de Inspección Preoperacional	24
Ilustración 7. Ejemplar de Inspección Diligenciada	26
Ilustración 8. Modelo de Autenticación.....	134
Ilustración 9. Modelo para Auth.js	136
Ilustración 10. Modelo de Empleados	137
Ilustración 11. Modelo de Vehículos	139
Ilustración 12. Modelo de Revisión y Logística	142
Ilustración 13. Modelo de Tesorería y Cartera	144
Ilustración 14. Formulario de pagos de SATT.....	147
Ilustración 15. Formulario de despachos de SATT	148
Ilustración 16. Estructura de Árbol para flotillas.....	154
Ilustración 17. Metodo GET para obtener todas las flotillas	155
Ilustración 18. Método POST de Flotillas	156
Ilustración 19. Código del formulario de flotillas.....	158
Ilustración 20. Invocando un Server Action	160
Ilustración 21. Formulario de Despacho.....	160
Ilustración 22. Contenido de una Server Action.....	162
Ilustración 23. Esquema de Carpetas de SATT	164
Ilustración 24. Muestra de código de prueba	168
Ilustración 25. Resultado de test	169
Ilustración 26. Módulo de Empleados vista escritorio.....	170
Ilustración 27. Vista móvil del módulo de empleados.....	170
Ilustración 28. Menú de Acceso versión móvil.	171
Ilustración 29. Barra de acceso rápido (solo en versión escritorio).....	172
Ilustración 30. Formulario de empleados (versión de escritorio).	173
Ilustración 31. Formulario de empleados (versión móvil).....	173
Ilustración 32. Métricas por carpetas	174
Ilustración 33. Resumen de métricas de sonar.....	175
Ilustración 34. Revisión de una Issue de fiabilidad	176
Ilustración 35. Corrección de Issue de Sonar.....	176
Ilustración 36. Resultado del proceso de compilación.....	177
Ilustración 37. Issue jsx reportada por sonar	177
Ilustración 38. Entradas DNS estático en routerboard.....	180
Ilustración 39. Regla NAT de puerto 80 a Nginx.	181
Ilustración 40. Túneles Wireguard para conexión a sedes.....	182
Ilustración 41. ACL de firewall	183

TRABAJO DE GRADO

	7
Ilustración 42. Interfaces VLAN.....	184
Ilustración 43. Resultado de análisis de web-check.xyz.....	187
Ilustración 44. Resultado de análisis con PageSpeed Insights (escritorio).	188
Ilustración 45. Resultado de análisis con PageSpeed Insights (versión móvil).	189

TRABAJO DE GRADO

8

Resumen

El Sistema Administrativo de Transporte Terrestre es un proyecto que busca apoyar el proceso de las empresas de transporte público terrestre automotor en sus modalidades intermunicipal, urbano colectivo y mixto. Fue concebido gracias a los procesos de una transportadora en el Urabá que ofrece servicios en todas esas modalidades, permitiendo tener un sistema robusto que ofrece servicios operativos y administrativos de alto impacto.

Para lograrlo, se utilizó la ingeniería del software y la arquitectura de sistemas mediante una metodología tradicional en cascada. Con un equipo de desarrollo conformado por dos personas y tecnologías sin pago por el uso como lo es Next.js, MySQL sobre Linux Ubuntu LTS se logró implementar en el servidor de la empresa el aplicativo SATT. Un sistema WEB que responde a un dominio propio de la empresa desde cualquier parte del mundo con acceso a internet, con políticas de seguridad óptimas y la portabilidad suficiente para su uso en el día a día desde móviles, tabletas o computadoras de escritorio. Un trabajo de ingeniería satisfactorio que no se hubiera podido lograr sin el conocimiento adquirido en el proceso de formación.

Palabras clave

Sistema Administrativo de Transporte Terrestre; Transporte Público; Desarrollo Full Stack; Javascript Framework; Arquitectura de Software.

TRABAJO DE GRADO

9

Introducción

El sistema de transporte público en Urabá no ha sufrido procesos de innovación desde que iniciaron a operar en 1968. Siempre se ha contado con un mecanismo manual para gestionar la flota que presta el servicio. Después de casi cincuenta años se puede evidenciar que ha mantenido la misma metodología.

En la actualidad los sistemas de transporte han sufrido una clara saturación del mercado. Esto radica principalmente en que la industria automotriz cada vez desarrolla vehículos más eficientes y accesibles, lo que estimula a los empresarios a invertir en la industria. Como resultado se obtiene un amplio parque automotor que tiende a saturar las plataformas físicas de transporte como son las terminales.

Por su parte, con la alta competencia, la vastísima regulación del estado y la informalidad, la industria del transporte ha tenido que buscar las formas de optimizar su operación, con el horizonte claro de cumplir a cabalidad la regulación estatal a la vez que se mantiene rentable su operación en el tiempo sin detrimento aparente de la calidad del servicio.

Para lograr la optimización, es necesario contar con Sistema Administrativo de Transporte Terrestre, el cual pueda gestionar la operación e integrarla con los lineamientos administrativos y estatales a fin de que puedan ayudar al empresario a gestionar eficazmente la logística que antecede la liberación del servicio a la vez que cuida sus intereses económicos.

TRABAJO DE GRADO

10

En este proyecto abordaremos, desde la ingeniería del software y la arquitectura general de sistemas, como podemos crear un sistema que puedan utilizar las empresas del servicio público de transporte terrestre automotor.

TRABAJO DE GRADO

11

Marco Teórico

Introducción al transporte público en Colombia

El transporte público en Colombia tiene una connotación industrial pues su único objetivo es el de generar un servicio esencial y tener una contraprestación económica por ello. De ahí que según la ley 105 (Ministerio de Transporte, 1993):

El transporte público es una industria encaminada a garantizar la movilización de personas o cosas por medio de vehículos apropiados a cada una de las infraestructuras del sector, en condiciones de libertad de acceso, calidad y seguridad de los usuarios, sujeto a una contraprestación económica [...]

Como se puede inferir, existen elementos fundamentales en su definición, como lo son vehículos apropiados a cada una de las infraestructuras del sector; condiciones de libertad de acceso, calidad y seguridad. Por su parte, existen también principios derivados del carácter público, pues en su definición más simple el transporte busca garantizar la movilización de personas o cosas de un lugar a otro. Sin embargo, cuando se introduce el término público, se configura un carácter de regulación y vigilancia. Esto lo confirma el segundo principio de la misma ley (Ministerio de Transporte, 1993):

DEL CARACTER DE SERVICIO PUBLICO DEL TRANSPORTE. La operación del transporte público en Colombia es un servicio público bajo la regulación del Estado, quien ejercerá el control y la vigilancia necesarios para su adecuada prestación, en condiciones de calidad, oportunidad y seguridad.

TRABAJO DE GRADO

12

En síntesis, se puede definir el transporte público como aquella actividad que consiste en movilizar personas o cosas de un lugar a otro en condiciones de accesibilidad, calidad y seguridad sujeto a una contraprestación económica en un ambiente industrializado controlado y vigilado por el estado.

En Colombia el transporte adquiere un componente de protección estatal, esto gracias al principio de esencialidad, el cual obliga a entidades territoriales a garantizarlo como derecho fundamental. Bajo este principio muchas herramientas jurídicas fueron proporcionadas a los entes locales para que, bajo ciertos escenarios, fuera posible integrar uno u otro modelo de transporte y que permitiera progresivamente el desarrollo de sus comunidades.

A nivel nacional, la máxima autoridad encargada de la formulación y adopción de las políticas, planes, programas, proyectos y regulación económica en materia de transporte, tránsito e infraestructura en todas las modalidades posibles está en cabeza del Ministerio de Transporte. (Función Pública, 2015)

Modalidades de transporte

Para hablar de modalidades, la normatividad en la materia sugiere inicialmente ciertas clasificaciones importantes que se deben considerar. En su categoría más amplia se encuentra el transporte aéreo, fluvial, marítimo, férreo y terrestre. En este caso nos concentraremos en el transporte terrestre, denominado propiamente como Transporte Terrestre Automotor.

TRABAJO DE GRADO

13

Dentro del transporte terrestre automotor encontramos varias modalidades. Estas modalidades están determinadas por el ministerio de transporte y buscan clasificar los diferentes servicios al que los usuarios pueden tener acceso. Además, cada modalidad tiene una reglamentación exclusiva y unas reglas de juego; incluso pudiendo variar en las autoridades competentes. Por ejemplo, la modalidad de Transporte Terrestre Automotor de Pasajeros por Carretera es el transporte que hoy conocemos como transporte intermunicipal. Esta modalidad es dirigida directamente por el Ministerio de Transporte; mientras que el Transporte Terrestre Automotor Colectivo Metropolitano, Distrital, y Municipal de pasajeros es lo que se conoce coloquialmente como el transporte urbano; en este caso la autoridad encargada de su dirección es la autoridad municipal.

El transporte intermunicipal es el que sucede entre diferentes municipios; quiere decir que intervienen varias jurisdicciones en su recorrido y en ese caso tiene sentido que su dirección ocurra exclusivamente desde el ministerio de transporte, pues hablamos de una entidad de orden nacional. Si su dirección estuviera en cabeza de los alcaldes de cada municipio supondría un caos de autoridad y un posible conflicto de intereses. Es importante tener en cuenta que hay excepciones y casos especiales contemplados en la norma. Por ejemplo, en las zonas metropolitanas, en donde intervienen varios municipios en su conformación, corresponde para el transporte urbano que su autoridad resida en el municipio capital o nombrado como líder. En ese caso el transporte urbano podría atravesar varios municipios sin convertirse propiamente en una modalidad intermunicipal.

TRABAJO DE GRADO

14

Planteamiento del problema.

Las empresas de transporte en su gran mayoría cuando nacieron en la industria, utilizaron el modelo de negocio afiliador. Este modelo lo que busca es que las personas naturales que quieran invertir en el negocio, adquieran desde un vehículo y lo ponga a disposición de la empresa para que esta lo opere en sus rutas autorizadas y así obtener un beneficio por la prestación del servicio. Aún así, esta relación comercial empezó a tener múltiples variantes en cada unidad de negocio y según la cantidad de vehículos era mas o menos complicado administrarlos.

En esencia, el negocio funcionaba de una forma sencilla: la empresa se habilitaba en una modalidad¹, luego adquiría los permisos de operación sobre ciertas rutas. Según el estimativo de demanda del servicio y la rentabilidad, la empresa cobraba unos valores al propietario y éste último operaba la ruta y obtenía las ganancias por mover los pasajeros. Sin embargo, el propietario era quien ofrecía el momento de verdad a los usuarios, es decir, era el primer eslabón en la cadena comercial. Esto le daba cierta sensación de libertad y control ya que las empresas difícilmente podrían rastrear la demanda de las rutas.

Estas relaciones comerciales no eran tan reguladas por el estado, incluso que era prácticamente arbitrario. Cualquier persona podía simplemente conducir el vehículo, operar la ruta y hasta con cierto grado de embriaguez. Tampoco había mucha regulación en cuanto a jornadas laborales, pólizas de seguro y muchos otros ítems que hoy son fundamentales.

¹ Existen varias modalidades definidas por el ministerio de transporte. Ver Modalidades en el vocabulario.

TRABAJO DE GRADO

15

Esto llevó a que las empresas empezaran a colocar mucho control en sus operaciones; desde los procesos de selección de conductores, la vigilancia de su jornada laboral hasta la forma como se vigila el mantenimiento de las maquinas.

Las empresas de transporte al ser prestadoras de un servicio público esencial² deben estar vigiladas por el estado y esta vigilancia ocurre en las nombradas superintendencias, en este caso, superintendencia de puertos y transporte. Estas también emiten mecanismos de control que van en concordancia a lo reglado por el Ministerio de Transporte.

Con todo lo anterior, las empresas deben contar con mecanismos de control efectivos que permitan gestionar correctamente:

- El recaudo de los dineros pagados por cada vehículo.
- Los requisitos exigibles al vehículo y al conductor antes de su operación.
- El historial tanto del conductor, vehículo y propietario.

Estos controles no pueden simplemente llevarse en una hoja de Excel o un cuaderno por lo que se requiere de un sistema de información que permita, no solo resolver problemas del presente, sin sentar la base para resolver problemas del futuro.

² Que se considera irrenunciable, así como sucede con el acueducto, la energía eléctrica y el servicio de recolección de basuras.

TRABAJO DE GRADO

16

Objetivos

Objetivo General

Construir un sistema administrativo de transporte terrestre que permita el cumplimiento de la normatividad del sector transporte en la modalidad de servicio público terrestre automotor de pasajeros por carretera, colectivo municipal, distrital o metropolitano y mixto a la vez que optimiza la gestión administrativa y protección del patrimonio de las empresas de transporte.

Objetivos específicos

1. Extraer los requerimientos funcionales y expectativas no funcionales del cliente para determinar su necesidad, para posteriormente elegir una arquitectura orientada a satisfacer dichos requerimientos.
2. Diseñar un modelo de datos que se ajuste a las necesidades de requisitos.
3. Desarrollar un software que este acorde a la arquitectura elegida e implementarlo en el servidor de la empresa patrocinadora.
4. Realizar pruebas básicas de rendimiento.

TRABAJO DE GRADO

17

Metodología de Desarrollo

Para la realización de este proyecto se recurrió a la metodología de desarrollo tradicional en cascada. Para ello, se aplicaron todas las fases de análisis, diseño, desarrollo y despliegue en estricto orden. Esto con el fin de abarcar los requisitos fundamentales para el despliegue a producción y realizar una integración tecnológica de alto impacto como un esfuerzo inicialmente único.

Esto teniendo en cuenta la población objetivo, un gremio de transportadores que es un poco reacio al cambio y del cual se requieren esfuerzos importantes en capacitación. Por tal motivo el equipo decidió esta metodología, tratando de viabilizar un cambio de paradigma en la forma como la tecnología acompaña sus procesos en el día a día y cómo todos los procesos tienen un soporte digital.

Fase de Análisis

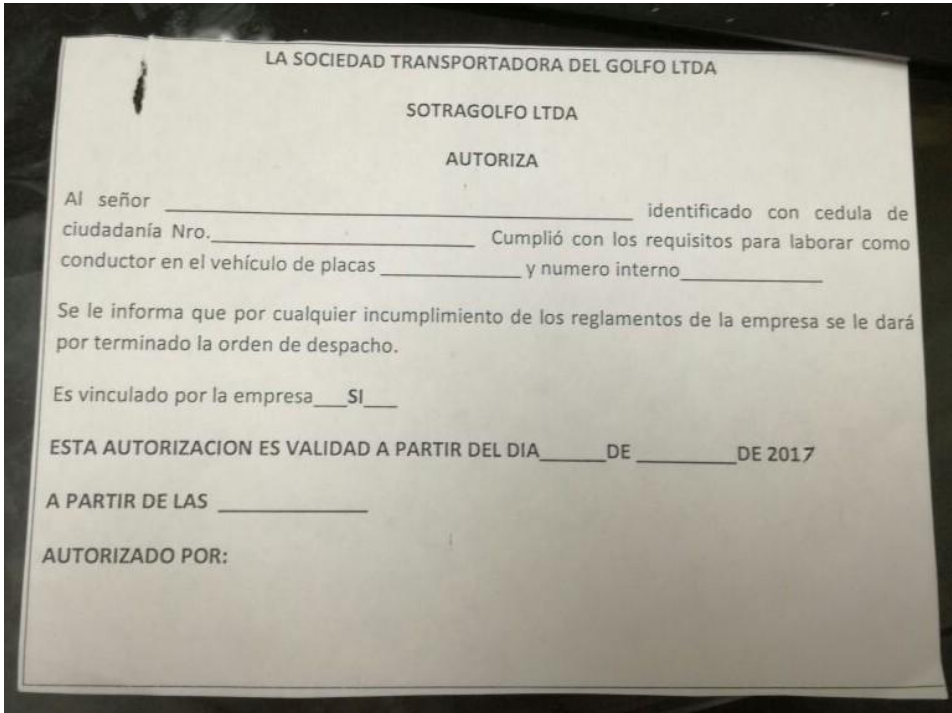
Metodología de investigación

Se usó una técnica aplicada, descriptiva y de tipo cualitativa ya que se observaron muy bien los procesos, conversando con cada líder involucrado y en general a todos los tipos de actores que intervienen en el organigrama.

Por su parte, también se recolectó información de formatos, reportes y plantillas existentes, que se contrastaron con las entrevistas de las personas que los producían y los usaban.

TRABAJO DE GRADO

18



LA SOCIEDAD TRANSPORTADORA DEL GOLFO LTDA
SOTRAGOLFO LTDA
AUTORIZA

Al señor _____ identificado con cedula de ciudadanía Nro. _____ Cumplió con los requisitos para laborar como conductor en el vehículo de placas _____ y numero interno _____

Se le informa que por cualquier incumplimiento de los reglamentos de la empresa se le dará por terminado la orden de despacho.

Es vinculado por la empresa SI

ESTA AUTORIZACION ES VALIDAD A PARTIR DEL DIA _____ DE _____ DE 2017

A PARTIR DE LAS _____

AUTORIZADO POR:

Ilustración 1. Foto de formato de autorización

En la ilustración 1 se puede observar que los formatos de autorización se llenaban a mano y con ello un conductor demostraba que podía manejar cierto vehículo. El conductor conservaba la autorización física y la mostraba al despachador. Sin embargo, el formato podría fácilmente violarse o perderse.

Los recaudos también tenían un componente manual, en donde se escribía diariamente el valor de los conceptos recaudados para una fecha específica. Esto suponía un mayor esfuerzo para su cálculo y gestión a final de mes.

TRABAJO DE GRADO

Escuela a C. Seguridad y Administradores del Golfo Ltda.
S.T.S. 11215 70
Resolución No. 000004 Junio 23 de 1997, emitida del Centro

PLANILLA DE APORTES DE ADMINISTRACIONES, SEGURIDAD SOCIAL Y OTROS
FECHA: Apartadó 1 de 11 de 2008 2010

ADMINISTRACION Y PARQUEO	SEGURIDAD SOCIAL	OTROS
3001	\$	\$
3002	\$	\$
3003	\$	\$
3004	\$	\$
3005	\$	\$
3006	40.500 2000 \$ 34800	\$
3007	\$	\$
3008	\$	\$
3009	\$	\$
3010	19000 1000 \$ 11.600	\$
3011	\$	\$
3012	19000 1000 \$ 11.600	\$
3013	19000 1000 \$ 11.600	\$
3014	\$	\$
3015	\$	\$
3016	\$	\$
3017	\$	\$
3018	\$	\$
3019	\$	\$
3020	19000 1000 \$ 11.600	\$
3021	19000 1000 \$ 11.600	\$
3022	\$	\$
3023	19000 1000 \$ 11.600	\$
3024	19000 1000 \$ 11.600	\$ 5000
3025	\$	\$
3026	\$	\$
3027	19000 1000 \$ 11.600	\$ 2600
3028	19000 1000 \$ 11.600	\$
3029	19000 1000 \$ 8778	\$
3030	19000 1000 \$ 11.600	\$
3031	19000 1000 \$ 11.600	\$
3032	19000 1000 \$ 11.600	\$ 2000
3033	19000 1000 \$ 11.600	\$
3034	19000 1000 \$ 11.600	\$
3035	\$	\$
3036	\$	\$
VALOR ADMINISTRACION	\$ 325.500	558.078
VALOR SEGURIDAD SOCIAL	\$ 205.978	3000
VALOR PARQUEO	\$ 17.000	
VALOR OTROS	\$ 9.600	
VALOR TOTAL	\$ 558.078	561.078

VEHICULOS QUE NO TRABAJARON HOY _____

FUNCIÓNARIO _____

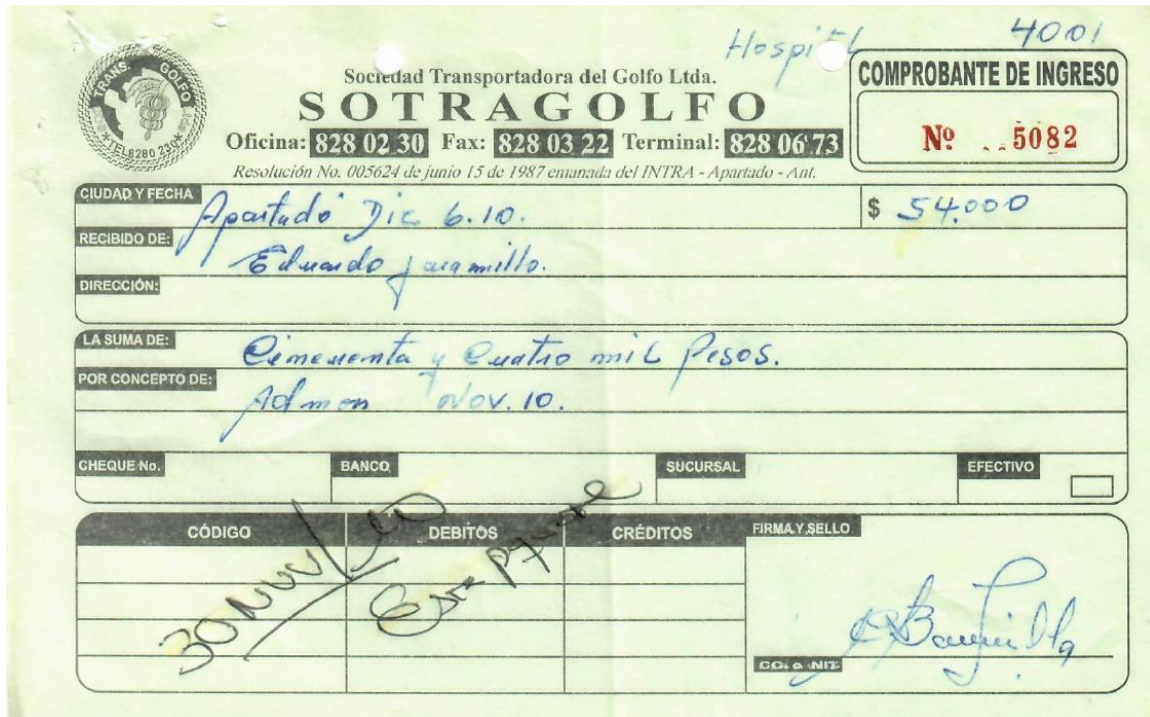
NOV 10

Ilustración 2. Planilla de Control Interno de Recaudo

Igualmente, se entregaba el comprobante al cliente y se generaba una copia espejo en posesión en la empresa para posteriores procesos de auditorías.

TRABAJO DE GRADO

20



Sociedad Transportadora del Golfo Ltda.
SOTRAGOLFO
Oficina: 828 02 30 Fax: 828 03 22 Terminal: 828 06 73
Resolución No. 005624 de junio 15 de 1987 emanada del INTRA - Apartado - Ant.

COMPROBANTE DE INGRESO
Nº 5082

CIUDAD Y FECHA: Apartado Dic 6.10. \$ 54.000

RECIBIDO DE: Eduardo Jacamillo.

DIRECCIÓN:

LA SUMA DE: Cimentación y Cuatro mil Pesos.

POR CONCEPTO DE: Adm en nov. 10.

CHEQUE No. BANCO SUCURSAL EFECTIVO

CÓDIGO	DEBITOS	CRÉDITOS	FIRMA Y SELLO
	30000 Eduardo Jacamillo		Eduardo Jacamillo

Ilustración 3. Copia de Comprobante de Ingreso

La empresa recaudaba los conceptos de Administración, Despacho, Seguridad Social y Fondo de Reposición. La administración y despacho le pertenecían a la empresa mientras que la seguridad social y fondo de reposición al conductor y vehículo respectivamente. Sin embargo, la empresa solo cobraba el despacho a los vehículos que trabajasen dicho día mientras que los demás se mantenían inamovibles. Aún así, se demostró que muchos recaudadores, aunque cobraban el despacho al conductor, no lo reportaban a la empresa justificando que el carro no había trabajado. Esto suponía pérdidas importantes para la empresa, requiriéndose mecanismos fiables de control.

También notamos que la empresa contaba con un archivo físico de mucha extensión, casi que se puede notar el espacio tan importante que se debía reservar para gestionar el archivo físico.



Ilustración 4. Archivo Físico Central

Existen también archivadores por toda la oficina que contienen información de todos los vehículos según la flota, requiriendo de esfuerzos y tiempos para encontrar los documentos de algún vehículo cuando estos son requeridos. Muchas veces, sin querer, estas carpetas eran movidas del archivador donde se suponen que debían estar, dificultando encontrarlas en procesos tan vitales como el aporte de documentos cuando existía un accidente de tránsito o cuando llegaban instituciones de vigilancia y control a realizar auditorías.

TRABAJO DE GRADO



Ilustración 5. Archivadores Medianos Distribuidos

TRABAJO DE GRADO

23

Los archivos asociados a los vehículos son de suma importancia para la operatividad; como analistas de la situación actual consideramos importante, en ese momento, digitalizar los documentos operativos ya que estos no siempre se mantenían actualizados en el archivo físico de la empresa.

El control de documentos como SOAT y Certificado de Revisión Técnico mecánica y de gases, se hacían en el patio operativo a través del coordinador de despacho, quien trataba de revisar que estos estuvieran al día. En algunos casos la empresa nos contó que las autoridades de tránsito encontraban vehículos circulando sin los documentos vigentes en los puestos de control y ni hablar de uno que otro accidente en donde se evidenció que no contaba con la revisión técnico mecánica y de gases al día.

Por normativa, la resolución 0315 de 2013 exige que los vehículos deben realizar una inspección preoperacional, es decir antes de realizarse el primer despacho del día. La empresa contaba con dos inspectores, quienes realizaban estas inspecciones en un formato impreso el cual estaba generando un volumen significativo de papel.

TRABAJO DE GRADO



INSPECCION PREOPERACIONAL



Placa: _____ Fecha ____/____/____ No Interno: _____

I	DOCUMENTACIÓN PORTADA	B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
	TALLER:				
	NIT Y/O C.C.				
1	SOAT			Vigente a la fecha	
2	Tarjeta de operación			Vigente a la fecha	
3	Licencia de conducción			Vigente a la fecha	
4	Certificación CDA			Vigente a la fecha	
5	Pólizas de responsabilidad civil			Vigente a la fecha	
6	Tarjeta de propiedad			Vigente a la fecha	
II	ASPECTOS DE SEGURIDAD	B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
	TALLER:				
	NIT Y/O C.C.				
1	Estado de llantas profundidad de labrado			2.5 mm profundidad sin irregularidades en la superficie	
2	Estado de llantas (presión)			Según la recomendación del fabricante. Ver Llanta.	
3	Luces de parqueo			Pastas, luces y señales en tablero en buen estado	
4	Luces de carretera			Con buen alcance y funcionando ambas farolas	
5	Direccionales			Pastas, luces y señales en tablero en buen estado	
6	Pastas de luces			Claras y sin rupturas	
7	Luces de frenos			Pastas, luces y señales en tablero en buen estado	
8	Luz de reversa			Pastas y luces en buen estado	
9	Pito			Con nivel de sonido aceptable	
10	Plumillas de parabrisas			Con buena ubicación y girando al ángulo correcto	
11	Extintor			Recarga último año y buena presión en manómetro	
12	Tacos			Por lo menos en buen estado	
13	Botiquín primeros auxilios			Elementos básicos	
14	Gato			Que funcione	
15	Llantas de repuesto			Para vehículos livianos 32 lbs. y 1 cm de labrado.	
16	Cruceña			Buen estado, sin puntos de soldadura extras	
17	Linterna			Pilas extras	
18	Salida de emergencia			Indicada con señales visibles	
III	ASPECTOS DE CALIDAD PARA PRESTAR EL SERVICIO	B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
	TALLER:				
	NIT Y/O C.C.				
1	Puerta de acceso			Buen funcionamiento, abra y cierre con facilidad	
2	Sillas			Sin resortes salidos, con espaldar y cojin de asiento	
3	Pasamanos			Limpios y bien sujetos al techo del bus	
4	Ventanas			Completas y fácil de abrir y cerrar sin elementos cortantes	
5	Pintura			Buen estado libre de grafitis y letreros	
6	Aseo en general			Limpio y buen olor	
7	Pisos			Limpios y buen estado sin elementos cortantes	
8	Estado de carrocería			Limpios y buen estado sin elementos cortantes	
IV	ASPECTOS MECANICOS	B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
	TALLER:				
	NIT Y/O C.C.				
1	Estado de batería			Con buen líquido de batería y bornes.	
2	Líquido de frenos			Dentro del nivel indicado por el fabricante	
3	Frenos			Con buena eficacia, sin tanto recorrido de bater	
4	Aceite de motor			Sin fugas	
5	Combustible			Sin fugas	
6	Dirección			Ajustada	

Representante Legal

Encargado de Inspección

Conductor

Ilustración 6. Plantilla en Blanco de Inspección Preoperacional

TRABAJO DE GRADO

25

Estas plantillas eran enviadas a la impresora laser y entregadas a los inspectores, quienes en el día podrían diligenciar aproximadamente 250 hojas de papel con el diligenciamiento de las inspecciones.

Esto traía múltiples problemas de control:

- Era difícil realizar la trazabilidad sobre un vehículo en particular, ya que encontrar las inspecciones de un día en cuestión requería mover mucho el archivo físico.
- Si los inspectores rotaban entre flotillas era difícil saber qué ítems estaban pendientes de corrección sin antes realizar una inspección y por ende los plazos se debían reestablecer por pérdida del historial, ya que ciertos ítems con novedad tenían plazos de corrección, sobre todo los relacionados con pintura, silletería y logos corporativos.
- En ocasiones las inspecciones se perdían por las lluvias fuertes que tomaban por sorpresa los inspectores, ya que se movilizaban en moto.
- Era difícil mantener en el tiempo las inspecciones ya que estas ocupaban mucho espacio en el archivo central y por orden de prioridad había que descartar más rápido las inspecciones.



TRABAJO DE GRADO



INSPECCION PREOPERACIONAL



Placa: JH1 513

Fecha 20/9/17

No Interno: 20004

I		DOCUMENTACIÓN PORTADA		B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
TALLER:							
NIT Y/O C.C.							
1	SOAT			✓		Vigente a la fecha	
2	Tarjeta de operación			✓		Vigente a la fecha	
3	Licencia de conducción			✓		Vigente a la fecha	
4	Certificación COA			✓		Vigente a la fecha	
5	Pólizas de responsabilidad civil			✓		Vigente a la fecha	
6	Tarjeta de propiedad			✓		Vigente a la fecha	
II		ASPECTOS DE SEGURIDAD		B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
TALLER:							
NIT Y/O C.C.							
0	Estado de llantas profundidad de labrado			✓		2.5 mm profundidad sin irregularidades en la superficie	
1	Estado de llantas (presión)			✓		Según la recomendación del fabricante. Ver llamas.	
3	Luces de parqueo			✓	✓	Pastillas, luces y señales en tablero en buen estado	
4	Luces de carretera			✓	✓	Con buen alcance y funcionando ambas farolas	
5	Dirigenciales			✓	✓	Pastillas, luces y señales en tablero en buen estado	
6	Patillos de luces			✓		Claras y sin rupturas	
7	Luces de frenos			✓		Pastillas, luces y señales en tablero en buen estado	
8	Luz de reversa			✓		Pastillas y luces en buen estado	
9	Piso			✓		Con nivel de sonido aceptable	
10	Pumillas de parabrisas			✓		Con buena ubicación y girando al ángulo correcto	
11	Extintor			✓		Recarga último año y buena presión en manómetro	
12	Tacos			✓		Por lo menos en buen estado	
13	Baliquin primeros auxilios			✓		Elementos básicos	
14	Cable			✓		Que funcione	
15	Llantas de repuesto			✓		Para vehículos livianos 32 lbs. y 1 cm. de la llanta.	
16	Cruceña			✓		Buen estado, sin puntos de soldadura extras	
17	Linterna			✓		Pilas extras	
18	Salida de emergencia			✓		Indicada con señales visibles	
III		ASPECTOS DE CALIDAD PARA PRESTAR EL SERVICIO		B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
TALLER:							
NIT Y/O C.C.							
1	Puerta de acceso			✓		Buen funcionamiento, abra y cierre con facilidad	
2	Sillas			✓		Sin resortes salidos, con espaldar y cojín de asiento	
3	Pasamanos			✓		Limpios y bien sujetos al techo del bus	
4	Ventanas			✓		Completas y fácil de abrir y cerrar sin elementos cortantes	
5	Finjura			✓		Buen estado libre de grafitis y letreros	
6	Aseo en general			✓		Limpio y buen olor	
7	Pisos			✓		Limpios y buen estado sin elementos cortantes	
8	Estado de carrocería			✓		Limpios y buen estado sin elementos cortantes	
IV		ASPECTOS MECANICOS		B	M	SITUACIÓN ESPERADA	PLAZO CORRECCIÓN
TALLER:							
NIT Y/O C.C.							
1	Estado de batería			✓		Con buen líquido de batería y bornes.	
2	Líquido de frenos			✓		Dentro del nivel indicado por el fabricante.	
3	Frenos			✓		Con buena eficacia, sin tanto recorrido de boster	
4	Aceite de motor			✓		Sin fugas	
5	Combustible			✓		Sin fugas	
6	Dirección			✓		Ajustada	

Representante Legal

Alcides Blanco
Encargado de Inspección

Alcides Blanco
Conductor

Ilustración 7. Ejemplar de Inspección Diligenciada

TRABAJO DE GRADO

27

Requisitos Funcionales

Se han definido primero unas secciones para agrupar los requisitos. Estas secciones son las siguientes:

Tabla 1. Secciones Funcionales

<i>Código</i>	<i>Sección</i>	<i>Finalidad</i>
SC01	Control de Acceso	Agrupar las funciones de control de gestión de usuarios, autenticación, permisos y control de recursos con los que cuenta el sistema.
SC02	Empleados	Definir la funcionalidad de manejo del talento humano. Funciones de creación de empleados y sus funciones segregadas de documentos, restricciones y demás que involucren el personal.
SC03	Vehículos	Encapsular toda la funcionalidad relacionada con la gestión directa y esencial del parque automotor. Registro de vehículos, documentos entre otros.
SC04	Revisiones y Logística	Agrupar funciones relacionadas con la liberación de servicio como son los despachos, alistamiento de vehículos y sus prerequisites.
SC05	Tesorería y Cartera	Define la funcionalidad relacionada con el tesoro: control de recaudos, manejo de taquillas, políticas de pago entre otros.
SC06	Administrativa	Agrupar funcionalidad administrativa y especial que solo podrán ejecutar usuarios con privilegios avanzados y que son de propagación transversal a otras secciones funcionales.

Con estas secciones se definieron los siguientes requisitos funcionales:

Tabla 2. Requisitos Funcionales

<i>Código</i>	<i>Sección</i>	<i>Requisito</i>
RQF01	SC01	<p>El sistema debe permitir la gestión de usuarios. Los usuarios deben poder darse de alta, de baja y editar, permitiendo también, la carga de una foto de perfil.</p> <p>Los usuarios iniciarán sesión con correo electrónico y contraseña.</p>
RQF02	SC01	<p>El control de acceso de usuarios será por medio del correo electrónico. Esto quiere decir que una vez sea dado de alta el usuario, el sistema debe enviar un enlace en donde puede crear su contraseña. También en cualquier momento el sistema debe permitir el restablecimiento de la contraseña bien sea por un administrador o el mismo usuario, utilizando el correo electrónico como medio fundamental para el control de contraseña.</p>
RQF03	SC01	<p>El sistema debe permitir establecer permisos a cada usuario. Para ello debe seleccionar el usuario en cuestión y posteriormente podrá asignarle módulos a este indicando además las acciones permitidas sobre dicho módulo. En cualquier momento un administrador puede agregar o quitar acciones al igual que módulos al usuario en cuestión.</p>
RQF04	SC02	<p>El sistema debe permitir la gestión de empleados. Los empleados deben contar con un cargo y estos cargos también deben poder gestionarse desde el sistema. También el sistema debe permitir la carga de una foto de perfil.</p> <p>Un empleado debe contar con la siguiente información:</p> <p>Tipo de documento, Documento, Cargo, Tipo de licencia (Opcional), Nombres y Apellidos, Genero, Fecha de nacimiento, Ciudad de residencia, Si fuma o no, Tipo de sangre o RH, Ciudad de nacimiento, Ciudad de residencia, Dirección, Teléfono, Teléfono alternativo (Opcional), Barrio, Deporte que practica, Correo electrónico.</p>

RQF05	SC02	El sistema dará la opción de gestionar autorizaciones a empleados. Estas autorizaciones son válidas cuanto el empleado es un conductor. En este caso, una autorización es la posibilidad de un conductor poder operar en uno o varios vehículos vinculados a la empresa. Una autorización requiere además del vehículo en cuestión al cual se autoriza, una fecha de inicio y de finalización, siendo la de finalización opcional. En el caso que no se proporcione fecha de finalización, la autorización quedará indefinida.
RQF06	SC02	El sistema debe poder permitir la revocatoria de TODAS las autorizaciones de un empleado. Para ello solo será necesario, aparte de seleccionar el empleado en cuestión, la fecha de finalización o retiro del empleado conductor. De esta forma todas las autorizaciones relacionadas al conductor que estén vigentes quedarán finalizadas con dicha fecha.
RQF07	SC02	El sistema debe dar la posibilidad al usuario de inactivar/activar una autorización para un empleado en cualquier momento sin que esto implique borrarla o crearla nuevamente.
RQF08	SC02	El sistema tendrá la opción de definir los tipos de documentos que podrán cargarse a los empleados. Estos tipos de documentos deberán contener la siguiente información: Nombre de la tipología, flag si es <i>requerido para operar</i> , un flag si es predefinido, un flag que establece si bloquea la operación si el documento es faltante, alerta de notificación, descripción (Opcional).
RQF09	SC02	El sistema debe poder cargar y eliminar documentos a los empleados. Los documentos serán soportados en formato PDF pudiéndose visualizar desde la misma interfaz de la aplicación. El sistema debe solicitar: Tipo de documento, archivo, fecha inicio de vigencia, fecha fin de la vigencia (si la tipología no está predefinida), observaciones (Opcional).
RQF10	SC02	El sistema debe poder permitir la gestión de historial del empleado. Este historial permite agregar los eventos disciplinarios que tengan los colaboradores en toda su vida laboral. Los datos que se deben almacenar son: Título del evento, fecha del evento, anexo – archivo (opcional), detalles (texto largo).

TRABAJO DE GRADO

30

RQF11	SC02	<p>El sistema debe poder permitir la gestión de restricciones a los empleados. Las restricciones son controles médicos que deben vigilarse por el personal SST a fin de que ejecute su labor correctamente. Una restricción puede ser, por ejemplo: conducir con lentes. Esto permite al liberador del servicio verificar que el operario cumple con los equipos médicos recetados u otros controles de salud. La información solicitada deberá ser la siguiente:</p> <p>Nombre de la restricción, fecha de control, flag para informar en el despacho, observaciones.</p>
RQF12	SC02	<p>El sistema debe ofrecer una opción para extender las jornadas de trabajo de los empleados. Estos deben tener por norma un día de descanso por cada 6 días de trabajo continuo. Sin embargo, en ciertos casos es necesario que los conductores se extiendan uno o varios días. En este caso el sistema debe permitir el registro de esta extensión y permitir el despacho en caso de tener la extensión.</p>
RQF13	SC03	<p>El sistema debe gestionar los propietarios, estas son entidades poseedoras de los vehículos afiliados a la empresa. Su información soportada es:</p> <p>Imagen de perfil (opcional), tipo de persona: Natural o jurídica, tipo de documento de identificación, documento de identificación, razón social para jurídicas, nombre y apellidos para naturales, genero (si es natural), ciudad de residencia, barrio, dirección (opcional), teléfono, fecha de Nacimiento, correo electrónico.</p>
RQF14	SC03	<p>El sistema debe gestionar información de diferentes empresas de transporte, ya que los vehículos, por medio de colaboración empresarial, pueden ser operados por cualquiera de las empresas del convenio. La información que gestionará será básica:</p> <p>Razón social, nit, teléfono, dirección, correo electrónico.</p>
RQF15	SC03	<p>El sistema debe poder gestionar la información de los vehículos; estos son fundamentales para la organización. La información soportada es la siguiente:</p> <p>Cuatro fotos: frontal, trasera, lateral derecha, lateral izquierda, tipo de vehículo (se debe crear maestro), ciudad de matrícula, propietario principal, marca (se debe crear maestro), placa, modelo, color, línea, capacidad de pasajeros, cilindraje, tipo de combustible, tipo de carrocería, serie, No. De motor, No. Chasis, No. Licencia.</p> <p>Igualmente, en cualquier momento el vehículo puede inactivarse/activarse.</p>

TRABAJO DE GRADO

31

RQF16	SC03	<p>El sistema debe gestionar las flotillas. Estas son agrupaciones internas que tiene la empresa y de la cual aplica diferentes políticas de recaudo, valores y demás gestiones propias del negocio. La flotilla debe abarcar datos como:</p> <p>Nombre de la flotilla, fatiga máxima: Es un numero el cual establece cuantos días continuos permite trabajar los conductores despachados en las rutas de esa flotilla, descripción.</p>
RQF17	SC03	<p>El sistema tendrá la opción de definir los tipos de documentos que podrán cargarse a los vehículos. Estos tipos de documentos deberán contener la siguiente información:</p> <p>Nombre de la tipología, un flag si es requerido para operar, un flag si es predefinido, un flag que establece si bloquea la operación si el documento es faltante, alerta de notificación, descripción (Opcional).</p>
RQF18	SC03	<p>El sistema debe permitir la gestión de cupos. Los cupos son cuentas que se asocian a los vehículos según su lugar de operación y sobre la cual se registran tanto los despachos como los pagos. Los cupos tienen un numero interno de modo que los datos que deberá solicitar son la flotilla de trabajo y el numero interno.</p> <p>El sistema registrará la fecha de activación cuando se crea el cupo y la fecha de finalización cuando se elimina. No pueden existir dos cupos iguales asociados a un vehículo existente y tampoco puede haber dos placas diferentes asociadas a un mismo número interno activos a la vez.</p>
RQF19	SC03	<p>El sistema debe poder cargar y eliminar documentos a los vehículos. Los documentos serán soportados en formato PDF pudiéndose visualizar desde la misma interfaz de la aplicación. El sistema debe solicitar:</p> <p>Tipo de documento (los definidos para vehículos), archivo, fecha inicio de vigencia, fecha fin de la vigencia (si la tipología no está predefinida), observaciones (Opcional)</p>
RQF20	SC03	<p>El sistema debe poder permitir la gestión de historial del vehículo. Este historial permite agregar los eventos que le sucedan a los diferentes vehículos como accidentes y otros. Los datos que se deben almacenar son:</p> <p>Título del evento, fecha del evento, anexo – archivo (opcional), detalles (texto largo).</p>

TRABAJO DE GRADO

32

RQF21	SC03	El sistema debe permitir registrar los cambios de propiedad, es decir, cuando un vehículo es vendido a otro propietario se hace necesario hacer un traspaso de propiedad. En este caso el sistema tendrá una sección en donde se pueden registrar los diferentes traspasos manteniendo la trazabilidad. Basta con seleccionar el propietario, la fecha en que es efectivo el traspaso (fecha de matrícula) y observaciones opcionales.
RQF22	SC03	La aplicación debe poder registrar la vinculación de un vehículo. La vinculación es la acción de incluir un automotor al parque de la empresa, otorgándole tarjeta de operación. Basta con indicar la transportadora en donde se vincula, la modalidad, la fecha de vinculación y algunas notas opcionales.
RQF23	SC04	El sistema debe permitir establecer plantillas de revisión a los vehículos. Las revisiones son inspecciones técnicas y preoperacionales que se hacen a los automotores. Cada plantilla tiene que ser dirigida a un tipo de vehículo en donde se definen sus elementos de revisión. Los elementos de revisión, por usabilidad, deben tener instrucciones de revisión, características de un resultado positivo o esperado y características de un resultado posiblemente negativo para permitir a los inspectores guiarse en caso de dudas. Además, debe establecerse un flag de criticidad, el cual en caso de ser verdadero y resultado negativo el diagnostico, el sistema debe impedir la operación de dicho vehículo mientras un inspector constata la novedad y supera la restricción con otra inspección positiva. Estas plantillas se deben clasificar en dos grupos: Preoperacionales o Revisiones. Las preoperacionales son más rápidas y simples, en cambios las revisiones son más técnicas y ofrecen plazos de corrección.
RQF24	SC04	El sistema debe tener la opción de realizar las revisiones a los automotores. Para ello debe seleccionar el vehículo y con la información del tipo de vehículo cargarse la plantilla diseñada para dicha tipología. El usuario podrá navegar entre cada ítem de revisión indicando si este elemento se encuentra o no en óptimas condiciones. En caso de no estar en óptimas condiciones, el sistema debe dar la posibilidad de cargar una foto como evidencia y establecer una fecha de plazo para corrección. La siguiente información es relevante: Vehículo al que se le hace la revisión, Conductor presente al momento de hacer la revisión, fecha de vencimiento.

TRABAJO DE GRADO

33

RQF25	SC04	<p>La aplicación deberá poder realizar inspecciones preoperacionales a los vehículos. Estas inspecciones anteceden a la operación y emiten un concepto sobre su estado optimo visible para operar. Estas inspecciones deben ser realizadas por los conductores y serán cargadas las plantillas del tipo “preoperacional”. Estas inspecciones se registran por cada usuario con sesión activa en el sistema y el módulo permitido. El sistema solo pedirá la elección del vehículo sobre el cual se va a hacer la inspección preoperacional asumiendo que es un conductor con la sesión activa, de modo que listará los vehículos autorizados para dicho usuario/conductor. Este módulo será lo más intuitivo posible, permitiendo indicar elemento por elemento su estado bueno o malo con posibilidad de carga de imagen de evidencia cuando este tiene un reporte negativo.</p>
RQF26	SC04	<p>El sistema debe gestionar la información de rutas con las que cuenta la empresa. Para esto debe gestionar la información concerniente a: flotilla, origen, destino, kilometraje de la ruta, cliente (necesario para rutas especiales), notas o comentarios.</p>
RQF27	SC04	<p>El sistema debe poder permitir registrar los despachos que se hacen a en las diferentes flotillas y rutas con los que cuenta la empresa. Para ello deben validarse una serie de requisitos antes de generar la tarjeta de despacho (autorización de salida). El sistema mostrará una opción para elegir:</p> <p>Fecha y hora de despacho, vehículo a ser despachado, conductor, ruta. notas opcionales.</p> <p>Además, el sistema debe tener un mecanismo para mostrar las restricciones marcadas como “informar en el despacho” para que el operador las exija al momento de liberar el despacho.</p> <p>Después de crearse el despacho el sistema debe generar un imprimible relacionando el código de despacho, fecha y hora de despacho, vehículo, ruta, conductor y coordinador de despacho.</p>
RQF28	SC05	<p>El sistema debe contar con la opción de gestionar las taquillas de recaudo. Estas taquillas son necesarias para el proceso de recaudo y cobros a los propietarios en los patios de operación. La información a gestionar será:</p> <p>Nombre de la taquilla, IP Address, DDNS (Opcional), descripción (opcional).</p>

TRABAJO DE GRADO

34

RQF29	SC05	<p>La aplicación debe permitir gestionar los conceptos de recaudo. Estos conceptos son necesarios para establecer los ítems de cobro, para esto requiere gestionar la siguiente información:</p> <p>Nombre del concepto, Flag inamovible, Flag uso en cartera, observaciones (opcionales).</p>
RQF30	SC05	<p>El sistema debe permitir la configuración de los conceptos de recaudo con tarifas diferenciadas según la flotilla. Para lograr esto se debe elegir la flotilla, el año en que aplica y la tarifa establecida para dicho concepto.</p>
RQF31	SC05	<p>La aplicación debe permitir la creación de plantillas de pago para cada vehículo en cuestión. En esta plantilla se establecen los conceptos que debe pagar diariamente.</p>
RQF32	SC05	<p>El sistema debe permitir el registro de pagos. Los pagos se crean seleccionando el vehículo, el tipo de pago (principal o secundario), variante de pago (simple o múltiple), un flag de mantenimiento para descuentos, un flag de tesorería para funciones avanzadas de alto nivel, la fecha de cuota (inicial y final), el vehículo (cupó), notas opcionales; esto cargará la plantilla de pagos y el sistema permitirá la opción de agregar más conceptos si es requerido en el pago.</p> <p>Al crearse el pago el sistema debe generar el imprimible del recibo para entregar al cliente.</p> <p>El sistema debe permitir solicitar la anulación. Esta anulación es requerida si el recibo ha sido impreso.</p>
RQF33	SC05	<p>La aplicación debe ofrecer una sección de cuadro de caja en donde los recaudadores pueden ver el dinero que han recaudado por fecha y terminal. En el caso de los administradores podrán ver todos los registros de todos los usuarios y caja. Además, el cuadro de caja tendrá un sistema de códigos para indicar que un usuario administrador/tesorero le ha recibido el recaudo.</p>
RQF34	SC06	<p>El sistema tendrá un sistema de supervisión a los documentos cargados a vehículos y empleados. Para ello el sistema notificará a los usuarios aprobadores que un nuevo documento fue cargado, luego éste podrá verificar que existe congruencia entre los datos del archivo con los ingresados por el usuario que lo cargó. De esta manera, podrá aceptar o rechazar la carga de dicho documento.</p>

RQF35	SC06	El sistema tendrá una sección en donde se podrán aprobar o rechazar las solicitudes de anulación de pagos.
-------	------	--

Análisis Arquitectónico

Dentro de los requisitos no funcionales del sistema, detectamos que los dispositivos móviles eran fundamentales y que eran estos en gran medida los que tenían que ejecutar ciertos puntos de función del sistema. Así pudimos llegar a la siguiente tabla:

Tabla 3. Drivers Arquitectónicos

<i>Característica</i>	<i>Nivel de Importancia (1 a 5)</i>
Funcionalidad	5
Fiabilidad	5
Usabilidad	5
Eficiencia de desempeño	3
Mantenibilidad	4
Portabilidad	4
Compatibilidad	2
Seguridad	5

Requisitos No Funcionales

Tomando como base los drivers arquitectónicos, se definieron los requisitos no funcionales esperados del sistema como muestra en la siguiente tabla:

TRABAJO DE GRADO

36

Tabla 4. Requisitos no funcionales

<i>Código</i>	<i>Requisito</i>
RQNF01	El sistema debe contener módulos que se ejecuten bien en dispositivos móviles, por tal motivo debe contar con capacidades <i>response design</i> en la mayoría de sus módulos. Esto permite ser compatible también con Tablet y computadores de escritorio de múltiples tamaños.
RQNF02	El sistema debe ser altamente fiable, con características de integridad que permitan confiar sus valores y cálculos, ya que se requieren abarcar funcionalidades específicas de control del tesoro. Esto requiere operaciones centralizadas con aislamiento de transacciones a nivel de bases de datos.
RQNF03	El sistema debe ser eficiente, con características de rendimientos que permitan la fluidez de sus operaciones. Una transacción no debe superar los cuatro (4) segundos de espera.
RQNF04	El sistema debe poder contar con arquitecturas que le permitan integrarse con otros sistemas a través de APIs, ya que en el futuro deben poder conectarse con aplicaciones contables y otros sistemas de la Supertransporte como parte del proceso de vigilancia y control que esta ejerce sobre las empresas de transporte.
RQNF05	El sistema debe contar con mecanismos de protección al acceso no autorizado a la información sensible. Por ello debe contar con políticas de autenticación robustas con credenciales de acceso (correo y contraseña) manteniendo siempre un esquema de autorización para las transacciones.
RQNF06	El sistema debe ser mantenible en el tiempo. Por lo tanto, debe contar con un esquema que permita establecer las versiones de las dependencias y la actualización controlada a través del tiempo.
RQNF07	El sistema debe contar con interfaz intuitiva que permita al usuario ejecutar las tareas con naturalidad sin que se vuelva complicado su manejo. Esto debido a la población objetiva la cual no es muy inclinada al mundo digital.
RQNF08	El sistema debe cumplir la funcionalidad esperada en los requisitos funcionales adaptándose al proceso de negocio de la empresa de transporte.

TRABAJO DE GRADO

37

Restricciones

- El sistema debe ser programado en tecnologías libres que permitan el uso sin licencia de pago del entorno de desarrollo.
- Se debe aprovechar el servidor físico con el que cuenta la compañía. HP Proliant G9.

Casos de Uso

Control de Acceso

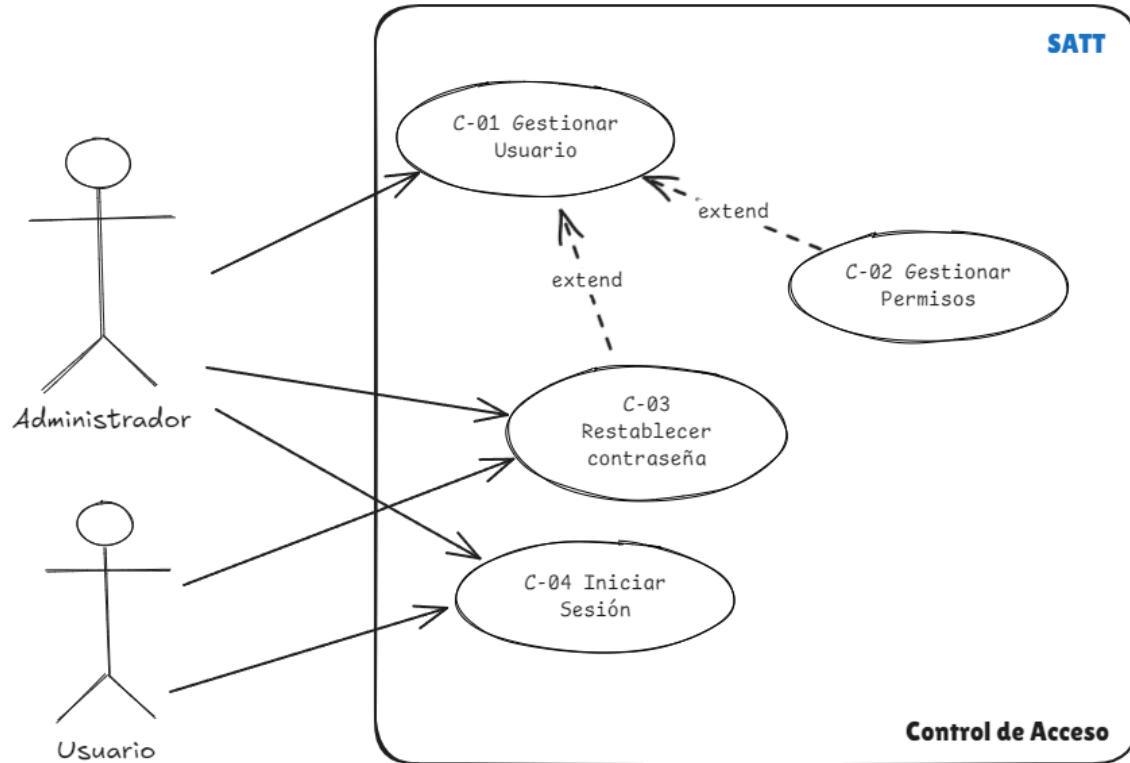


Figura 1. Casos de Uso de Control de Acceso

C-01 Gestionar Usuario

La gestión de usuarios permite al administrador de usuarios crear, editar, inactivar y dar de baja a los diferentes usuarios del sistema.

Escenarios

1. Crear Usuario

1. El administrador del sistema selecciona la opción de crear nuevo.
2. El sistema presenta un formulario en donde puede anexar la foto y llenar los datos requeridos los cuales son: foto de perfil, tipo de documento y número de documento, nombres, apellidos, fecha de nacimiento y correo electrónico.
3. El administrador confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al administrador el resultado exitoso del proceso.
5. El sistema envía por correo electrónico al nuevo usuario las instrucciones para configurar la contraseña.

1.1 Precondiciones:

- El correo electrónico no puede estar registrado en la lista de usuarios ya que este es único y utilizado para el inicio de sesión.

2. Editar Usuario

1. El administrador selecciona el usuario que desea editar, indicando al sistema el registro y dando la opción disponible que permite editar.
2. El sistema carga la información del usuario en un formulario y permite la edición.
3. El administrador modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Precondiciones:

TRABAJO DE GRADO

40

- Si el administrador cambia el correo electrónico, el nuevo correo electrónico tampoco debe existir en el sistema ya que es un campo único y es necesario para iniciar sesión.

3. Eliminar o dar de baja a un usuario

1. El administrador selecciona el usuario que desea editar, indicando al sistema el registro y dando la opción disponible que permite eliminar o dar de baja.
2. El sistema realiza una confirmación de seguridad.
3. El administrador confirma.
4. El sistema le da de baja al usuario.

C-02 Gestionar Permisos

Un administrador puede gestionar los permisos de los usuarios. Esto simplemente es que pueda agregar recursos o módulos a los usuarios, así como las acciones que pueden hacer sobre ellos.

Escenarios

1. Agregar Recurso / Módulo

1. El administrador selecciona el usuario sobre el cual va a trabajar y luego busca la opción de permisos. Posteriormente ejecuta la opción de nuevo permiso.
2. El sistema carga los recursos o módulos que no están asignados al usuario.
3. El administrador selección a el recurso que desea agregar.

TRABAJO DE GRADO

41

4. El sistema muestra las acciones que puede marcar sobre dicho módulo como creación, edición, listar, leer y otras opciones especiales según aplique.
5. El administrador marca las acciones que desea y confirma al sistema.
6. El sistema aplica la configuración y retroalimenta el estado al administrador.

2. Editar acciones de un Recurso / Módulo

1. El administrador selecciona el usuario sobre el cual va a trabajar y luego busca la opción de permisos.
2. El sistema carga los recursos que actualmente tiene el usuario agregado permitiendo la opción de modificar acciones.
3. El administrador selecciona sobre el comando que le permite editar las acciones.
4. El sistema muestra las acciones que puede marcar sobre dicho módulo como creación, edición, listar, leer y otras opciones especiales según aplique, marcando las que tiene asignadas actualmente.
5. El administrador marca/desmarca las acciones que desea y confirma al sistema.
6. El sistema aplica la configuración y retroalimenta el estado al administrador.

3. Eliminar un Recurso / Módulo

1. El administrador selecciona el usuario sobre el cual va a trabajar y luego busca la opción de permisos.
2. El sistema carga los recursos que actualmente tiene el usuario agregado permitiendo la opción de eliminar.
3. El administrador selecciona sobre el comando que le permite eliminar el recurso.

TRABAJO DE GRADO

42

4. El sistema confirma al administrador sobre la acción de eliminar.
5. El administrador confirma al sistema.
6. El sistema elimina el recurso y sus acciones retroalimentando el estado al administrador.

C-03 Reestablecer Contraseña

Los usuarios en cualquier momento pueden reestablecer su contraseña por olvido. Esta función puede ser invocada desde diferentes partes del sistema.

Escenarios

1. Flujo Principal

1. Desde la opción de inicio de sesión se puede invocar la acción de “olvide mi contraseña”.
2. Al darle clic el sistema muestra al sistema un formulario en donde podrá escribir el correo electrónico del usuario que ha olvidado la contraseña.
3. El sistema confirma al usuario que se ha enviado un correo con las instrucciones de recuperación.

2. Flujo Secundario

1. Un usuario con privilegios elevados podrá seleccionar un usuario en el sistema e invocar la acción de reestablecer su contraseña.
2. El sistema confirma al usuario que se ha enviado un correo con las instrucciones de recuperación.

2.1 Precondiciones:

- El usuario debe existir en el sistema y poderse encontrar con el correo electrónico.

C-04 Iniciar Sesión

Los usuarios para poder interactuar con el sistema deberán iniciar sesión con un correo electrónico y una contraseña.

Escenarios

Flujo Principal

1. El usuario entra a la URL del sistema.
2. El sistema muestra el logo de la compañía y el formulario para iniciar sesión el cual consta de correo electrónico y contraseña.
3. El usuario ingresa sus credenciales y acciona el botón de iniciar sesión.
4. El sistema verifica las credenciales y da paso a los recursos del sistema que le han sido asignados como usuario.

Precondiciones:

- El usuario debe existir en el sistema y poderse encontrar con el correo electrónico.
- El usuario no puede estar bloqueado o sin contraseña asignada.
- El usuario no puede estar dado de baja.

Empleados

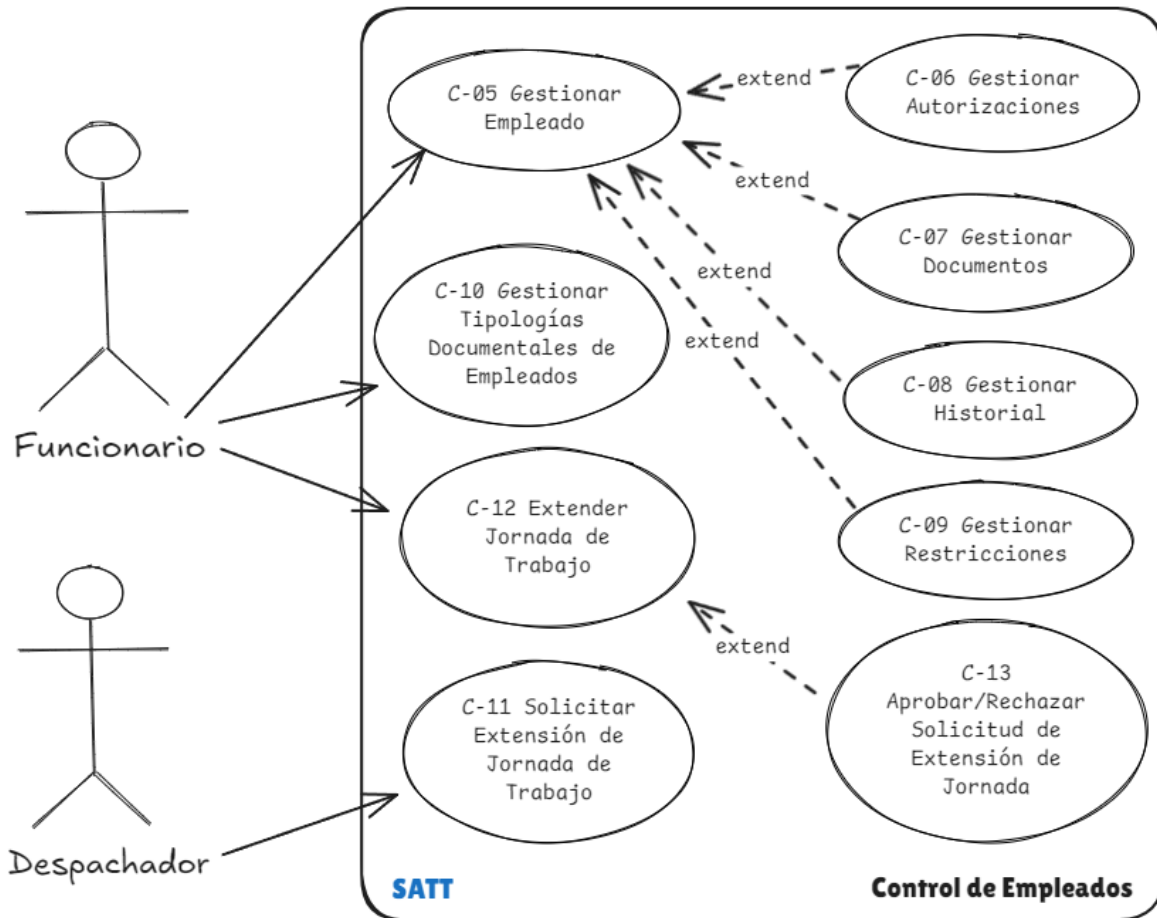


Figura 2. Casos de uso de control de empleados

C-05 Gestionar Empleado

Escenarios

1. Crear Empleado

1. El funcionario ingresa al módulo de empleados y selecciona la opción de crear o nuevo.

TRABAJO DE GRADO

45

2. El sistema presenta un formulario en donde puede anexar la foto y llenar los datos requeridos.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otro empleado con correo o tipo de documento y documento iguales.

1.2 Postcondiciones:

- El empleado debe quedar listado en el sistema.

2. Editar Empleado

1. El funcionario selecciona el empleado que desea editar, indicando al sistema el registro y dando la opción disponible que permite editar.
2. El sistema carga la información del empleado en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Precondiciones:

- Si el funcionario cambia el correo electrónico, el tipo de documento y documento, los nuevos datos no deben existir en el sistema para otro empleado.

3. Eliminar o dar de baja a un empleado

TRABAJO DE GRADO

46

1. El funcionario selecciona el empleado que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar o dar de baja.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema le da de baja al empleado.

3.1 Postcondiciones:

- El empleado debe quitarse de la lista.

C-06 Gestionar Autorizaciones

Las autorizaciones son las que permiten a los empleados conductores, operar ciertos vehículos, por lo tanto, es importante gestionar estas autorizaciones en el sistema para las funciones de logística.

Escenarios

1. Crear Autorización

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de autorizaciones.
2. El sistema lista las autorizaciones actuales del empleado y muestra la opción de crear o agregar nueva(s) autorización(es).
3. El funcionario acciona el botón de crear nueva(s) autorización(es).
4. El sistema muestra un formulario en donde puede elegir los diferentes vehículos que va autorizar para el empleado.

TRABAJO DE GRADO

47

5. El funcionario selecciona los diferentes vehículos y llena los demás datos requeridos de fecha de inicio, fecha final (si aplica) y observaciones. Aplica la opción de guardar.
6. El sistema crea las autorizaciones y retroalimenta al funcionario el resultado del proceso.

1.1 Precondiciones:

- No puede existir dos autorizaciones sobre un mismo vehículo.

1.2 Postcondiciones:

- La autorización creada queda listada en pantalla.

2. Editar Autorización

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de autorizaciones.
2. El sistema lista las autorizaciones actuales del empleado y muestra la opción de editar una autorización.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargado; permitiendo editar las fechas de vigencia y las observaciones.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- La autorización debe estar creada.

3. Eliminar Autorización

- El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de autorizaciones.
- El sistema lista las autorizaciones actuales del empleado y muestra la opción de eliminar una autorización.
- El funcionario ejecuta el comando de eliminación de la autorización.
- El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
- El funcionario confirma la eliminación.
- El sistema realiza la revocatoria de la autorización y retroalimenta el estado del proceso.

3.1 Precondiciones:

- La autorización debe estar creada en el sistema y en estado válido.

4. Inactivar / Activar una autorización

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de autorizaciones.
2. El sistema lista las autorizaciones actuales del empleado y muestra la opción de activar/desactivar una autorización.
3. El funcionario ejecuta sobre el control que permite activar/inactivar la autorización.
4. El sistema realiza el cambio de estado a la autorización y retroalimenta al usuario sobre el resultado de la transacción.

TRABAJO DE GRADO

49

4.1 Postcondiciones:

- Si el estado inicial era activo, ahora quedará inactiva.
- Si el estado inicial era inactivo, ahora quedará activa.

C-07 Gestionar Documentos

Los documentos son necesarios para mantener un archivo organizado directamente del empleado.

Escenarios

1. Cargar nuevo documento

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de documentos.
2. El sistema lista los documentos actuales del empleado y muestra la opción de agregar un nuevo documento.
3. El funcionario acciona el botón de crear un nuevo documento.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como el archivo, el tipo de documento, fecha de inicio de vigencia, fecha de vencimiento (si aplica) y observaciones opcionales.
5. El funcionario carga el archivo, diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema carga el documento en el sistema e informa al funcionario sobre el resultado exitoso del proceso.

TRABAJO DE GRADO

50

1.2 Precondiciones:

- El tipo de documento a cargar no existe en el sistema, esta vencido o próximo a vencerse.
- El archivo a cargar debe ser PDF o formato imagen.
- La tipología documental debe estar creada en el sistema.

1.2 Postcondiciones:

- El documento creado debe listarse en pantalla.

2. Editar Documento

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de documentos.
2. El sistema lista los documentos actuales del empleado y muestra la opción de editar en cada registro.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargado; permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- El documento debe estar cargado.

3. Eliminar Documento

TRABAJO DE GRADO

51

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de documentos.
2. El sistema lista los documentos actuales del empleado y muestra la opción de eliminar en cada registro.
3. El funcionario ejecuta el comando de eliminación de la autorización.
4. El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
5. El funcionario confirma la eliminación.
6. El sistema realiza la eliminación y retroalimenta el estado del proceso.

3.1 Postcondiciones:

- El documento desaparece de la lista de documentos del empleado.

4. Visualizar un documento

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de documentos.
2. El sistema lista los documentos actuales del empleado y muestra la opción de visualizar en cada registro.
3. El funcionario ejecuta sobre el control que permite visualizar un documento.
4. El sistema muestra, en una pantalla embebida (siempre que el dispositivo lo permita) el contenido del documento.

C-08 Gestionar Historial

Permite almacenar los eventos importantes que tenga el empleado para su gestión de hoja de vida laboral. El historial se crea a partir de eventos y cada registro es un evento.

Escenarios

1. Crear un nuevo evento de historial

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de historial.
2. El sistema lista los eventos actuales del empleado y muestra la opción de agregar uno nuevo.
3. El funcionario acciona el botón de crear un nuevo evento.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como el título del evento, la fecha en que sucede, un anexo opcional y unos detalles como texto largo.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema crea el evento e informa al funcionario sobre el resultado exitoso del proceso.

1.1 Precondiciones:

- El empleado debe estar creado.

1.2 Postcondiciones:

- El evento creado debe listarse en pantalla.

2. Editar Evento

TRABAJO DE GRADO

53

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de historial.
2. El sistema lista los eventos actuales del empleado y muestra la opción de agregar uno nuevo.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargado; permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- El evento debe estar creado previamente.

3. Eliminar Evento

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de historial.
2. El sistema lista los eventos actuales del empleado y muestra la opción de eliminar en cada registro.
3. El funcionario ejecuta el comando de eliminación del evento.
4. El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
5. El funcionario confirma la eliminación.

TRABAJO DE GRADO

54

6. El sistema realiza la eliminación y retroalimenta el estado del proceso.

3.1 Postcondiciones:

El evento desaparece de la lista de eventos del empleado.

C-09 Gestionar Restricciones

Permite realizar operaciones con las restricciones de los empleados, los cuales dan referencia sobre los cuidados médicos y ocupacionales que deben tener para ejecutar su labor. Permite a los funcionarios realizar seguimiento a estas restricciones y hacerlas cumplir fielmente.

Escenarios

1. Crear Restricción

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de restricciones.
2. El sistema lista las restricciones actuales del empleado y muestra la opción de agregar nueva.
3. El funcionario acciona el botón de crear.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como el nombre de la restricción, la fecha de control (o vencimiento), un interruptor para indicar si informa o no al área de despacho o si esta se vuelve visible para ellos y unos detalles técnicos.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.

TRABAJO DE GRADO

55

6. El sistema crea la restricción e informa sobre el resultado exitoso del proceso.

1.1 Precondiciones:

- El empleado debe estar creado.
- La fecha de control siempre debe ser superior a la actual.

1.2 Postcondiciones:

- La restricción quedará listada en pantalla.

2. Editar Restricción

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de restricciones.
2. El sistema lista las restricciones actuales del empleado y muestra la opción de editar en cada registro.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargado; permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- La restricción debe estar creado previamente.

3. Eliminar Restricción

TRABAJO DE GRADO

56

1. El funcionario ingresa al módulo de empleados, selecciona el empleado y entra a la subsección de restricciones.
2. El sistema lista las restricciones actuales del empleado y muestra la opción de eliminar en cada registro.
3. El funcionario ejecuta el comando de eliminación.
4. El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
5. El funcionario confirma la eliminación.
6. El sistema realiza la eliminación y retroalimenta el estado del proceso.

3.1 Postcondiciones:

- La restricción queda listada en pantalla con la leyenda de “superada”.

C-10 Gestionar Tipologías Documentales de Empleados

Es una gestión casi de tipo maestro que determina la configuración y tipologías documentales permitidas en las cargas de documentos de los empleados.

Escenarios

1. Crear Tipología

1. El funcionario ingresa al módulo de las tipologías documentales para empleados.
2. El sistema lista las tipologías documentales creadas (en caso de existir).
3. El funcionario acciona el botón de crear o nueva.

TRABAJO DE GRADO

57

4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como el nombre de la tipología, configuraciones de requerido para operar, predefinido, días de vigencia, bloquear operación si falta, alerta de notificación en días y una descripción opcional.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema crea la tipología e informa sobre el resultado exitoso del proceso.

1.1 Precondiciones:

- El nombre de la tipología no puede existir previamente.

1.2 Restricciones:

- Si la tipología es configurada como “Requerida para operar” se podrá elegir entre bloquear o no la operación si el documento, o topología, es faltante.
- En caso de que la tipología no sea marcada como “Requerida para operar”, la función “Bloquear operación si falta” no podrá configurarse o estará deshabilitada.
- Si la tipología es marcada como “Predefinida”, obligatoriamente se debe indicar cuantos días de vigencia tiene la tipología, en caso contrario este campo (Días de vigencia) es opcional.

1.3 Postcondiciones:

- La tipología quedará listada en pantalla.

2. Editar Tipología

1. El funcionario ingresa al módulo de las tipologías documentales para empleados.

TRABAJO DE GRADO

58

2. El sistema lista las tipologías documentales creadas con la opción de editar en cada una.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargado; permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- La tipología debe estar creada previamente.
- Los nuevos datos de la topología no pueden coincidir con otra tipología diferente a la que se encuentra en edición.

2.2 Restricciones:

- Si la tipología es configurada como “Requerida para operar” se podrá elegir entre bloquear o no la operación si el documento, o topología, es faltante.
- En caso de que la tipología no sea marcada como “Requerida para operar”, la función “Bloquear operación si falta” no podrá configurarse o estará deshabilitada.
- Si la tipología es marcada como “Predefinida”, obligatoriamente se debe indicar cuantos días de vigencia tiene la tipología, en caso contrario este campo (Días de vigencia) es opcional.

3. Eliminar Tipología

1. El funcionario ingresa al módulo de las tipologías documentales para empleados.
2. El sistema lista las tipologías documentales creadas con la opción de eliminar en cada registro.
3. El funcionario ejecuta el comando de eliminación.
4. El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
5. El funcionario confirma la eliminación.
6. El sistema realiza la eliminación lógica y retroalimenta el estado del proceso.

3.1 Postcondiciones:

- La tipología queda eliminada de la lista en pantalla. Sin embargo, si la tipología tiene asociados documentos ya cargados, esta no podrá eliminarse físicamente por relación de integridad y tendrá que marcarse para borrado lógico.

C-11 Solicitar Extensión de Jornadas de Trabajo

El sistema internamente vigila que los conductores no excedan ciertos días continuos de trabajo, esto se hace para evitar la fatiga y posibles accidentes por no descansar lo suficiente. Sin embargo, en ciertas ocasiones es necesaria la extensión de uno o varios días y postergar dicho descanso. Los coordinadores de despacho (despachadores) podrán solicitar extender la jornada de un conductor.

Escenarios

Flujo Principal

TRABAJO DE GRADO

60

1. El usuario una vez seleccionado el conductor, el sistema según el contexto permitirá solicitar la extensión de jornada del conductor (empleado) seleccionado.
2. El sistema carga un formulario en donde se puede visualizar la fecha de descanso próxima, el control de fecha de descanso postergada o solicitada y un campo para justificar dicha solicitud.
3. El despachador diligencia los campos requeridos de fecha y justificación e invoca la opción de solicitar.
4. El sistema guarda la solicitud y notifica a los funcionarios que tengan permisos elevados para aprobar o rechazar dichas solicitudes. Luego el sistema notifica al despachador sobre el resultado del proceso.

Precondiciones:

- El conductor debe estar creado en el sistema.

Restricciones:

- La fecha de descanso extendida debe ser mayor a la fecha de descanso programada por el sistema.

C-12 Extender Jornada de Trabajo

Los usuarios con privilegios avanzados contarán con una sección donde podrán aprobar, rechazar o crear sus propias extensiones de jornada de trabajo para los conductores.

Escenarios

Flujo Principal

1. El funcionario ingresa al módulo de extensiones de jornada.
2. El sistema muestra las ultimas solicitudes y ofrece la opción de crear una nueva extensión de jornada.
3. El usuario pulsa el botón de crear o nueva.
4. El sistema presenta un formulario en donde el funcionario puede elegir el conductor, la fecha de su nuevo descanso y la justificación de la extensión.
5. El usuario diligencia los datos y acciona el botón de guardado.
6. El sistema registra la extensión de jornada y notifica el resultado de la transacción al funcionario.

Precondiciones:

- El conductor debe estar creado en el sistema.

Restricciones:

- La fecha de descanso extendida debe ser mayor a la fecha de descanso programada por el sistema.

Postcondiciones:

- La extensión de jornada se ve listada en pantalla en estado “Aprobado”.

C-13 Aprobar / Rechazar Solicitud de Extensión de Jornada

Los usuarios con privilegios avanzados contarán con una sección donde podrán aprobar, rechazar solicitudes de despachadores para extensión de jornada de un conductor.

Escenarios

Flujo Principal

1. El funcionario ingresa al módulo de extensiones de jornada.
2. El sistema muestra las últimas solicitudes y ofrece la opción de gestionar cada registro.
3. El funcionario pulsa sobre el botón de aprobar o rechazar solicitud.
4. El sistema cambia el estado de la solicitud de aprobado o rechazado según el caso y retroalimenta al usuario sobre el resultado de la operación.

Precondiciones:

- Debe existir la solicitud en el sistema creada por un usuario de menor privilegio como el despachador.

Postcondiciones:

La solicitud muestra el estado del funcionario (aprobado o rechazado) eliminando los botones de gestión sobre el registro gestionado.

Vehículos

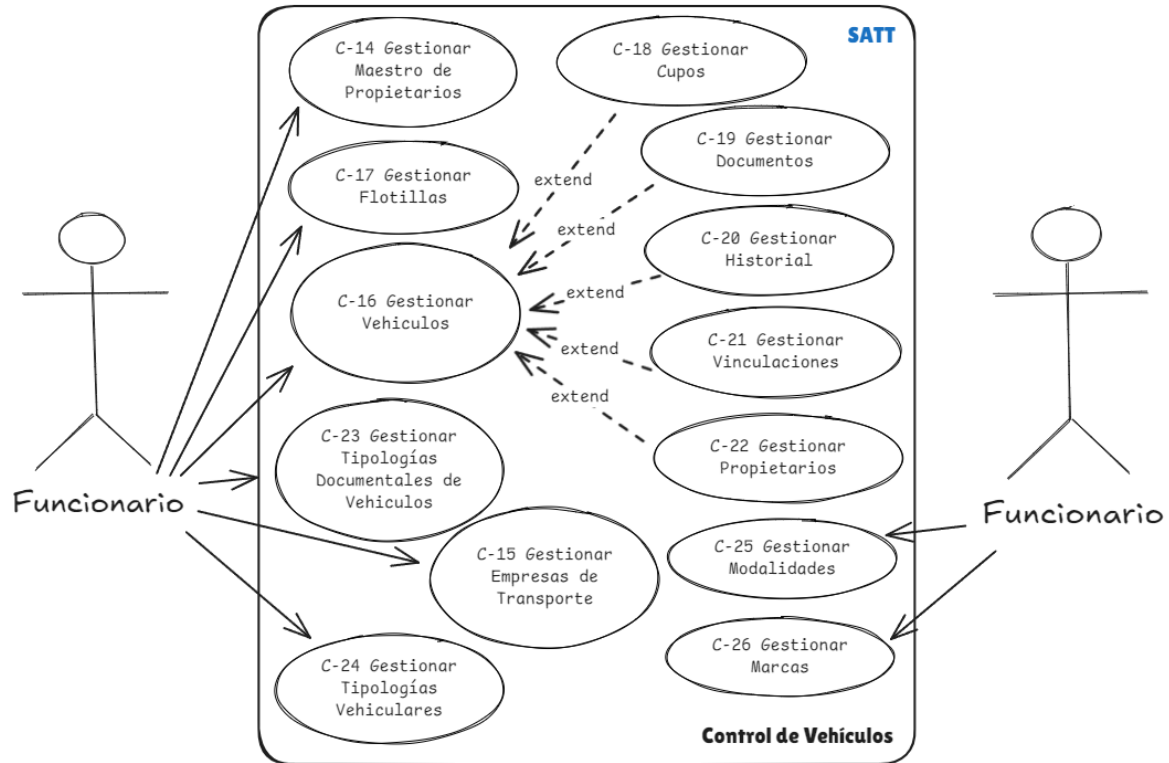


Figura 3. Casos de uso de control de vehiculos

C-14 Gestionar Maestro de Propietarios

Los propietarios son personas naturales y jurídicas que se deben gestionar en un maestro para luego asociarlos a los vehículos.

Escenarios

1. Crear Propietario

TRABAJO DE GRADO

64

1. El funcionario ingresa al módulo de maestro de propietarios y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede anexar la foto y llenar los datos requeridos.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.2 Precondiciones:

- No puede existir otro propietario con correo o tipo de documento y documento iguales.

1.2 Postcondiciones:

- El propietario debe quedar listado en el sistema.

2. Editar Propietario

1. El funcionario selecciona el propietario que desea editar, indicando al sistema el registro y dando la opción disponible que permite editar.
2. El sistema carga la información del propietario en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Precondiciones:

TRABAJO DE GRADO

65

- Si el funcionario cambia el correo electrónico, el tipo de documento y documento, los nuevos datos no deben existir en el sistema para otro propietario.

3. Eliminar o dar de baja a un propietario

1. El funcionario selecciona el propietario que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar o dar de baja.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema le da de baja al propietario.

3.1 Precondiciones:

- El propietario debe existir y no estar asociado a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

- El propietario debe quitarse de la lista.

C-15 Gestionar Empresas de Transporte

Las empresas de transporte son maestros que se deben de gestionar para asociar a los vehículos cuando estos son vinculados.

Escenarios

1. Crear Empresa de Transporte

TRABAJO DE GRADO

66

1. El funcionario ingresa al módulo de maestro de empresas de transporte y selecciona la opción de crear o nueva.
2. El sistema presenta un formulario en donde puede diligenciar los datos requeridos.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otra empresa con tipo de documento y documento iguales.

1.2 Postcondiciones:

- La empresa debe quedar listada en el sistema.

2. Editar Empresa de Transporte

1. El funcionario selecciona el propietario que desea editar, indicando al sistema el registro y dando la opción disponible que permite editar.
2. El sistema carga la información de la empresa en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Precondiciones:

- Si el funcionario cambia el tipo de documento y documento, los nuevos datos no deben existir en el sistema para otra empresa de transporte.

TRABAJO DE GRADO

67

3. Eliminar o dar de baja a una empresa de transporte

1. El funcionario selecciona la empresa que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar o dar de baja.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema le da de baja a la empresa de transporte.

3.1 Precondiciones:

- La empresa debe existir y no estar asociada a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

- La empresa de transporte debe quitarse de la lista.

C-16 Gestionar Vehículos

Los vehículos son la columna vertebral del proceso de negocio. Es fundamental su gestión para el proceso corporativo.

Escenarios

1. Crear Vehículo

1. El funcionario ingresa al módulo de vehículos y selecciona la opción de crear o nuevo.

TRABAJO DE GRADO

68

2. El sistema presenta un formulario en donde puede diligenciar los datos requeridos como son: 4 fotos, tipo de vehículo, ciudad de matrícula, propietario, marca, placa, modelo, color, línea, capacidad de pasajeros, cilindraje, combustible, tipo de carrocería, seria, numero de motor, numero de chasis y numero de licencia.
3. El funcionario diligencia los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otro vehículo con la misma placa.
- Debe existir marcas, propietarios y tipologías de vehículos creadas en el sistema.

1.2 Postcondiciones:

- El vehículo debe quedar listado en el sistema.

2. Editar Vehículo

1. El funcionario selecciona el vehículo que desea editar, indicando al sistema el registro y dando la opción disponible que permite editar.
2. El sistema carga la información del vehículo en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- El sistema no permite editar la placa.

3. Eliminar un Vehículo

TRABAJO DE GRADO

69

1. El funcionario selecciona el vehículo que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema elimina el vehículo junto con sus imágenes.

3.1 Precondiciones:

- El vehículo no puede estar asociado a otros módulos como despachos, revisiones o pagos.

3.2 Postcondiciones:

- El vehículo debe quitarse de la lista en pantalla.

C-17 Gestionar Flotillas

Las flotas son necesarias para clasificar internamente la empresa en centros de trabajo y despacho.

Escenarios

1. Crear Flotilla

1. El funcionario ingresa al módulo de flotillas y selecciona la opción de crear o nuevo.

TRABAJO DE GRADO

70

2. El sistema presenta un formulario en donde puede diligenciar los datos requeridos como son: nombre, el tiempo en días de vigilancia de fatiga y descripción (opcional).
3. El funcionario diligencia los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otra flotilla con el mismo nombre.

1.2 Postcondiciones:

- La flotilla queda listada en el sistema.

2. Editar Flotilla

1. El funcionario selecciona la flotilla que desea editar, indicando al sistema el registro y dando la opción disponible que permite editar.
2. El sistema carga la información de la flotilla en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

3. Eliminar una Flotilla

1. El funcionario selecciona la flotilla que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.

TRABAJO DE GRADO

71

3. El funcionario confirma.
4. El sistema elimina la flotilla y retroalimenta al funcionario sobre la transacción.

3.1 Precondiciones:

- La flotilla no puede estar asociado a otros módulos como vehículos.

3.2 Postcondiciones:

- La flotilla debe quitarse de la lista en pantalla.

C-18 Gestionar Cupos

Los cupos son cuentas con números internos en donde se pueden hacer los recaudos y los despachos de los vehículos.

Escenarios

1. Crear un nuevo cupo a un vehículo

- 1 El funcionario ingresa al módulo de vehículos, luego ingresa a la subsección de cupos y selecciona la opción de crear o nuevo.
- 2 El sistema presenta un formulario en donde puede diligenciar los datos requeridos como son: selección de una flota y un numero interno.
- 3 El funcionario diligencia los datos y selecciona el botón guardar.
- 4 El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

TRABAJO DE GRADO

72

- El vehículo debe estar creado en el sistema.
- La flotilla debe estar creada en el sistema.

1.2 Restricciones:

- Solo se permite un cupo por cada flotilla.
- El numero interno no puede estar activo previamente ya que solo se permite un numero único activo.

1.3 Postcondiciones:

- El cupo queda listado en el sistema.

2. Editar Cupo

1. El funcionario ingresa al módulo de vehículos, selecciona un vehículo y luego ingresa a la subsección de cupos.
2. El sistema lista los cupos disponibles con la opción de editar en cada uno de ellos.
3. El funcionario selecciona el cupo que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
4. El sistema carga la información del cupo en un formulario y permite la edición.
5. El funcionario modifica los datos necesarios y aplica la opción de guardar.
6. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- El numero interno no será modificable, solo podrá modificar la flotilla.
- Solo se permite un cupo por cada flotilla.

TRABAJO DE GRADO

73

- El numero interno no puede estar activo previamente ya que solo se permite un numero único activo.

3. Eliminar Cupo

1. El funcionario ingresa al módulo de vehículos, selecciona un vehículo y luego ingresa a la subsección de cupos.
2. El sistema lista los cupos disponibles con la opción de editar en cada uno de ellos.
3. El funcionario selecciona el cupo que desea eliminar, indicando al sistema el registro y ejecutando la opción disponible que permite eliminar.
4. El sistema realiza una confirmación de seguridad.
5. El funcionario confirma.
6. El sistema elimina el cupo y retroalimenta al funcionario sobre la transacción.

3.1 Precondiciones:

- El cupo debe estar activo o con fecha de terminación indefinida.

3.2 Postcondiciones:

- El cupo queda listado en pantalla, pero marcado con fecha de terminación del momento en que se hace la eliminación.

C-19 Gestionar Documentos

Los documentos de los vehículos permiten sustentar la operación y llevar un archivo digital del parque automotor de la empresa.

Escenarios

1. Cargar Documento

- 1 El funcionario ingresa al módulo de vehículos, luego ingresa a la subsección de documentos y selecciona la opción de crear o nuevo.
- 2 El sistema presenta un formulario en donde puede diligenciar los datos requeridos como son: la tipología documental, el archivo, fecha de inicio de vigencia, fecha vencimiento si aplica y observaciones opcionales.
- 3 El funcionario diligencia los datos y selecciona el botón guardar.
- 4 El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- El vehículo debe estar creado en el sistema.
- La tipología documental debe estar creada en el sistema.
- El tipo de documento a cargar no existe en el sistema, esta vencido o próximo a vencerse.
- El archivo a cargar debe ser PDF o formato imagen.

1.2 Postcondiciones:

- El documento queda listado en el sistema.

2. Editar Documento

1. El funcionario ingresa al módulo de vehículos, selecciona un vehículo y luego ingresa a la subsección de cupos.

TRABAJO DE GRADO

75

2. El sistema lista los cupos disponibles con la opción de editar en cada uno de ellos.
3. El funcionario selecciona el cupo que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
4. El sistema carga la información del cupo en un formulario y permite la edición.
5. El funcionario modifica los datos necesarios y aplica la opción de guardar.
6. El sistema retroalimenta el estado del proceso.

3. Eliminar Documento

1. El funcionario ingresa al módulo de vehículos, selecciona un vehículo y luego ingresa a la subsección de documentos.
2. El sistema lista los documentos disponibles con la opción de eliminar en cada uno de ellos.
3. El funcionario selecciona el documento que desea eliminar, indicando al sistema el registro y ejecutando la opción disponible que permite eliminar.
4. El sistema realiza una confirmación de seguridad.
5. El funcionario confirma.
6. El sistema elimina el documento y retroalimenta al funcionario sobre la transacción.

3.1 Precondiciones:

- El documento debe estar creado en el sistema.

3.2 Postcondiciones:

- El documento se elimina del listado en pantalla.

4. Visualizar un documento

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de documentos.
2. El sistema lista los documentos actuales del vehículo y muestra la opción de visualizar en cada registro.
3. El funcionario ejecuta sobre el control que permite visualizar un documento.
4. El sistema muestra, en una pantalla embebida (siempre que el dispositivo lo permita) el contenido del documento.

C-20 Gestionar Historial

Permite almacenar los eventos importantes que tenga el vehículo. El historial se crea a partir de eventos y cada registro es un evento.

Escenarios

1. Crear un nuevo evento de historial

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de historial.
2. El sistema lista los eventos actuales del vehículo y muestra la opción de agregar uno nuevo.
3. El funcionario acciona el botón de crear un nuevo evento.

TRABAJO DE GRADO

77

4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como el título del evento, la fecha en que sucede, un anexo opcional y unos detalles como texto largo.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema crea el evento e informa al funcionario sobre el resultado exitoso del proceso.

1.1 Precondiciones:

- El vehículo debe estar creado.

1.2 Postcondiciones:

- El evento creado debe listarse en pantalla.

2. Editar Evento

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de historial.
2. El sistema lista los eventos actuales del vehículo y muestra la opción de agregar uno nuevo.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargada, permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- El evento debe estar creado previamente.

3. Eliminar Evento

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de historial.
2. El sistema lista los eventos actuales del vehículo y muestra la opción de eliminar en cada registro.
3. El funcionario ejecuta el comando de eliminación del evento.
4. El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
5. El funcionario confirma la eliminación.
6. El sistema realiza la eliminación y retroalimenta el estado del proceso.

3.1 Postcondiciones:

El evento desaparece de la lista de eventos del empleado.

C-21 Gestionar Vinculaciones

Todos los vehículos de servicio público deben estar vinculados a una empresa transportadora para poder operar, eso lo determina la tarjeta de operación. En el sistema es fundamental controlar la vinculación de los vehículos.

Escenarios

TRABAJO DE GRADO

79

1. Vincular un nuevo vehículo

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de vinculaciones.
2. El sistema lista las vinculaciones actuales del vehículo (si existen) y muestra la opción de agregar una nueva.
3. El funcionario acciona el botón de crear una nueva.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como la transportadora, la modalidad, la fecha de vinculación y notas opcionales.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema crea la vinculación e informa al funcionario sobre el resultado exitoso del proceso.

1.1 Precondiciones:

- El vehículo debe estar creado.
- El vehículo debe estar desvinculado o sin vinculación activa.

1.2 Postcondiciones:

- La vinculación se muestra en pantalla.

2. Editar Vinculación

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de vinculaciones.
2. El sistema lista las vinculaciones actuales del vehículo y muestra la opción de editar en una vinculación activa.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.

TRABAJO DE GRADO

80

4. El sistema muestra un formulario con la información precargada, permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- La vinculación debe estar creada y activa.

3. Desvincular

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de vinculaciones.
2. El sistema lista las vinculaciones actuales del vehículo y muestra la opción de desvincular en cada registro.
3. El funcionario ejecuta el comando de desvinculación.
4. El sistema muestra al funcionario un formulario en donde puede escoger la fecha de desvinculación.
5. El funcionario ingresa los datos y confirma la desvinculación.
6. El sistema realiza la desvinculación y retroalimenta el estado del proceso.

3.1 Restricciones:

- La fecha de desvinculación debe ser mayor a la fecha de vinculación.

3.2 Postcondiciones:

TRABAJO DE GRADO

81

- El registro se muestra en pantalla agregando la fecha de desvinculación proporcionada por el funcionario.
- Los controles de edición son ocultados o deshabilitados.

C-22 Gestionar Propietarios

Durante todo el proceso corporativo un vehículo puede ser comprado y vendido en diferentes ocasiones, es por tanto que controlar la trazabilidad de los propietarios es fundamental para el proceso administrativo de la empresa.

Escenarios

1. Asociar un propietario inicial

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de propietarios.
2. El sistema muestra la opción de agregar uno nuevo.
3. El funcionario acciona el botón de asociar un propietario.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como seleccionar el propietario, la fecha de matrícula y las observaciones.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema asocia el propietario e informa al funcionario sobre el resultado exitoso del proceso.

1.2 Precondiciones:

- El vehículo debe estar creado.

TRABAJO DE GRADO

82

- El propietario debe estar creado.
- No puede existir propietarios asociados previamente.

1.2 Postcondiciones:

- El propietario asociado debe listarse en pantalla.

2. Registrar un traspaso de propiedad

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de propietarios.
2. El sistema lista los propietarios asociados y muestra la opción de agregar uno nuevo.
3. El funcionario acciona el botón de asociar un propietario.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como seleccionar el propietario, la fecha de matrícula y las observaciones.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema asocia el propietario e informa al funcionario sobre el resultado exitoso del proceso.

2.1 Precondiciones:

- El vehículo debe estar creado.
- El propietario debe estar creado.
- Debe existir al menos un propietario activo registrado.

2.2 Postcondiciones:

- El propietario asociado debe listarse en pantalla.

TRABAJO DE GRADO

83

- El propietario anterior mostrará la fecha final de matrícula un día antes de la fecha inicial de matrícula del nuevo propietario,

3. Editar Propiedad

1. El funcionario ingresa al módulo de vehículos, selecciona el vehículo y entra a la subsección de propietarios.
2. El sistema lista los propietarios actuales del vehículo y muestra la opción de editar sobre el propietario activo.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargada, permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

3.1 Precondiciones:

- El propietario debe estar asociado previamente.

3.2 Restricciones:

- El propietario en sí no es editable. Solo podrán editarse la fecha de matrícula y las observaciones.

C-23 Gestionar Tipologías Documentales de Vehículos

TRABAJO DE GRADO

84

Es una gestión casi de tipo maestro que determina la configuración y tipologías documentales permitidas en las cargas de documentos de los vehículos.

Escenarios

1. Crear Tipología

1. El funcionario ingresa al módulo de las tipologías documentales para vehículos.
2. El sistema lista las tipologías documentales creadas (en caso de existir).
3. El funcionario acciona el botón de crear o nueva.
4. El sistema muestra un formulario en donde podrá ingresar la información necesaria como el nombre de la tipología, configuraciones de requerido para operar, predefinido, días de vigencia, bloquear operación si falta, alerta de notificación en días y una descripción opcional.
5. El funcionario diligencia la información solicitada y ejecuta la función de guardado.
6. El sistema crea la tipología e informa sobre el resultado exitoso del proceso.

1.1 Precondiciones:

- El nombre de la tipología no puede existir previamente.

1.2 Restricciones:

- Si la tipología es configurada como “Requerida para operar” se podrá elegir entre bloquear o no la operación si el documento, o topología, es faltante.
- En caso de que la tipología no sea marcada como “Requerida para operar”, la función “Bloquear operación si falta” no podrá configurarse o estará deshabilitada.

TRABAJO DE GRADO

85

- Si la tipología es marcada como “Predefinida”, obligatoriamente se debe indicar cuantos días de vigencia tiene la tipología, en caso contrario este campo (Días de vigencia) es opcional.

1.3 Postcondiciones:

- La tipología quedará listada en pantalla.

2. Editar Tipología

1. El funcionario ingresa al módulo de las tipologías documentales para vehículos.
2. El sistema lista las tipologías documentales creadas con la opción de editar en cada una.
3. El funcionario acciona el botón de editar sobre el registro que desea modificar.
4. El sistema muestra un formulario con la información precargado; permitiendo editar sus campos.
5. El funcionario realiza los ajustes pertinentes y ejecuta la opción de guardar.
6. El sistema modifica los datos y retroalimenta al funcionario el resultado del proceso.

2.1 Precondiciones:

- La tipología debe estar creada previamente.
- Los nuevos datos de la topología no pueden coincidir con otra tipología diferente a la que se encuentra en edición.

2.2 Restricciones:

TRABAJO DE GRADO

86

- Si la tipología es configurada como “Requerida para operar” se podrá elegir entre bloquear o no la operación si el documento, o topología, es faltante.
- En caso de que la tipología no sea marcada como “Requerida para operar”, la función “Bloquear operación si falta” no podrá configurarse o estará deshabilitada.
- Si la tipología es marcada como “Predefinida”, obligatoriamente se debe indicar cuantos días de vigencia tiene la tipología, en caso contrario este campo (Días de vigencia) es opcional.

3. Eliminar Tipología

1. El funcionario ingresa al módulo de las tipologías documentales para vehículos.
2. El sistema lista las tipologías documentales creadas con la opción de eliminar en cada registro.
3. El funcionario ejecuta el comando de eliminación.
4. El sistema solicita confirmación por parte del funcionario para proceder con la eliminación.
5. El funcionario confirma la eliminación.
6. El sistema realiza la eliminación lógica y retroalimenta el estado del proceso.

3.1 Postcondiciones:

- La tipología queda eliminada de la lista en pantalla. Sin embargo, si la tipología tiene asociados documentos ya cargados, esta no podrá eliminarse físicamente por relación de integridad y tendrá que marcarse para borrado lógico.

C-24 Gestionar Tipologías Vehiculares

Las tipologías vehiculares son consideradas maestros, necesarios para crear los vehículos. El sistema permitirá la gestión de este maestro.

Escenarios

1. Crear Tipología Vehicular

1. El funcionario ingresa al módulo de maestro de tipos de vehículos y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede ingresar el nombre de la tipología, el tipo de licencia que se requiere para manejar el tipo de vehículo y una descripción opcional.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otro tipo de vehículo con el mismo nombre.

1.2 Postcondiciones:

- La tipología debe quedar listada en el sistema.

2. Editar Tipología

TRABAJO DE GRADO

88

1. El funcionario selecciona la tipología que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información de la tipología en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia el nombre de la tipología, este no debe coincidir con el nombre de otra tipología previamente guardada.

3. Eliminar tipología

1. El funcionario selecciona el registro que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema le da de baja al propietario.

3.1 Precondiciones:

- La tipología debe existir y no estar asociado a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

- La tipología debe quitarse de la lista.

Las modalidades están definidas por el Ministerio de Transporte, es importante tener claro cada vehículo en que modalidad se encuentra vinculado. Por tal motivo se requiere manejar un maestro de modalidades para que la empresa adapte su realidad conforme a la norma.

Escenarios

1. Crear Modalidad

1. El funcionario ingresa al módulo de maestro de modalidades y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede ingresar el nombre de la modalidad y una descripción opcional.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.2 Precondiciones:

- No puede existir otra modalidad con el mismo nombre.

1.2 Postcondiciones:

- La modalidad debe quedar listada en el sistema.

2. Editar Modalidad

TRABAJO DE GRADO

90

1. El funcionario selecciona la modalidad que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información de la modalidad en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia el nombre de la tipología, este no debe coincidir con el nombre de otra tipología previamente guardada.

3. Eliminar Modalidad

1. El funcionario selecciona la modalidad que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema elimina la modalidad y retroalimenta al funcionario sobre el estado de la transacción.

3.1 Precondiciones:

- La modalidad debe existir y no estar asociado a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

La modalidad debe quitarse de la lista.

C-26 Gestionar Marcas

Las marcas son un maestro que permite asociar a diferentes vehículos.

Escenarios

1. Crear Marca

1. El funcionario ingresa al módulo de maestro de marcas y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede ingresar el nombre de la marca y una descripción opcional.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otra marca con el mismo nombre.

1.2 Postcondiciones:

- La marca debe quedar listada en el sistema.

2. Editar Marca

1. El funcionario selecciona la marca que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información de la marca en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.

TRABAJO DE GRADO

92

4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia el nombre de la marca, este no debe coincidir con el nombre de otra marca previamente guardada.

3. Eliminar Marca

1. El funcionario selecciona la marca que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema elimina la marca y retroalimenta al funcionario sobre el estado de la transacción.

3.1 Precondiciones:

- La marca debe existir y no estar asociado a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

- La marca debe quitarse de la lista.

Revisiones y Logística

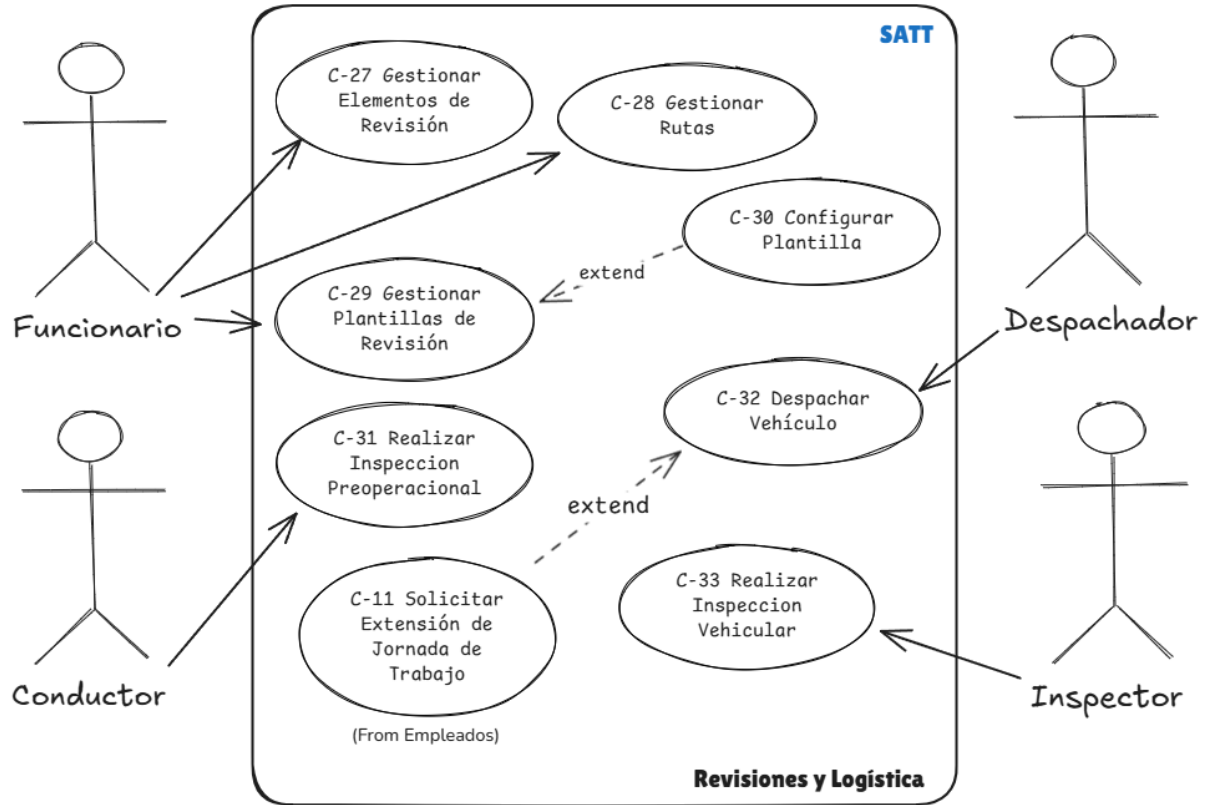


Figura 4. Casos de uso de Revisiones y Logística

C-27 Gestionar Elementos de Revisión

Este maestro permite elegir los ítems que los inspectores deben revisar en los vehículos.

Escenarios

1. Crear Ítem

TRABAJO DE GRADO

94

1. El funcionario ingresa al módulo de maestro de elementos de revisión y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede ingresar el nombre del ítem, un flag para indicar si el ítem es crítico, la inspección que se debe hacer, resultado esperado y resultado negativo.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otro ítem con el mismo nombre.

1.2 Postcondiciones:

- El ítem debe quedar listado en el sistema.

2. Editar Ítem

1. El funcionario selecciona el ítem que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información del ítem en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia el nombre del ítem, este no debe coincidir con el nombre de otra ítem previamente guardado.

3. Eliminar Ítem

1. El funcionario selecciona el ítem que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema elimina la marca y retroalimenta al funcionario sobre el estado de la transacción.

3.1 Precondiciones:

- El ítem debe existir y no estar asociado a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

- La marca debe quitarse de la lista.

C-28 Gestionar Rutas

Este maestro permite mantener el inventario de rutas que opera la empresa.

Escenarios

1. Crear Ruta

1. El funcionario ingresa al módulo de maestro de rutas y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede seleccionar la flotilla que opera la ruta, ingresar el origen, destino, kilometraje, cliente o población objetivo y notas opcionales.

TRABAJO DE GRADO

96

3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otra ruta con el mismo par origen destino.

1.2 Postcondiciones:

- La ruta queda listada en el sistema.

2. Editar Ruta

1. El funcionario selecciona la ruta que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información de la ruta en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia el par origen destino, este no debe coincidir con otro par de ruta previamente guardada.

3. Eliminar Ruta

1. El funcionario selecciona el ítem que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.

C-29 Gestionar Plantilla de Revisión

Las plantillas de revisión permiten agrupar los elementos de revisión según la calidad de revisión (inspección o preoperacional) y la tipología vehicular.

Escenarios

1. Crear Plantilla

1. El funcionario ingresa al módulo de plantillas de revisión y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede seleccionar el tipo de vehículo, el tipo de plantilla (Supervisión o Preoperacional) e ingresar una descripción opcional.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- El tipo de plantilla para el tipo de vehículo no ha sido creado previamente.

1.2 Postcondiciones:

- La plantilla queda listada en el sistema.

2. Editar Plantilla

TRABAJO DE GRADO

98

1. El funcionario selecciona la plantilla que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información de la plantilla en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia el tipo de plantilla y el tipo de vehículo, esta combinación no debe coincidir con otra plantilla previamente guardada.

3. Eliminar Plantilla

1. El funcionario selecciona la plantilla que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma la eliminación.
4. El sistema elimina la plantilla y retroalimenta al funcionario sobre el estado de la transacción.

3.1 Precondiciones:

- La plantilla debe existir y no estar asociado a ningún otro registro como inspecciones realizadas (por llave foránea).

3.2 Postcondiciones:

- Si la plantilla ya fue usada para realizar revisiones el sistema no la quita de la lista, pero elimina su configuración de ítems.

- Si la plantilla no fue usada para realizar revisiones la plantilla se quita de la lista en pantalla.

4. Activar / Inactivar Plantilla

1. El funcionario selecciona la plantilla que desea activar o desactivar, indicando al sistema el registro y dando la opción disponible que permite activar / desactivar.
2. El sistema cambia de estado alternando entre activo / inactivo según el caso y retroalimenta el cambio al funcionario.

4.1 Precondiciones:

- La plantilla debe existir.

4.2 Postcondiciones:

- Si la plantilla esta activa, cambiará a estado inactiva.
- Si la plantilla esta inactiva, cambiará a estado activa.

C-30 Configurar Plantilla

El sistema permitirá la configuración de plantillas. Esta configuración consiste en agregar o quitar ítems asociados a su funcionalidad.

Escenarios

1. Agregar Ítems

TRABAJO DE GRADO

100

1. El funcionario ingresa al módulo de la plantilla y accede a la subsección de configuración.
2. El sistema hace un listado de los ítems con los que cuenta la plantilla (si los tiene) y mostrando un comando o botón para asociar un nuevo ítem.
3. El funcionario acciona sobre el botón o comando que permite asociar un nuevo ítem.
4. El sistema muestra un formulario en donde puede seleccionar el ítem.
5. El funcionario confirma los datos y selecciona el botón guardar.
6. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- El ítem no está asociado a la plantilla.

1.2 Postcondiciones:

- EL ítem queda listado en pantalla y asociado a la plantilla.

2. Retirar Ítem

1. El funcionario ingresa al módulo de la plantilla y accede a la subsección de configuración.
2. El sistema hace un listado de los ítems con los que cuenta la plantilla (si los tiene) y mostrando un comando o botón para retirar el ítem listado.
3. El funcionario acciona el botón para retirar el ítem.
4. El sistema solicita confirmación al funcionario.

TRABAJO DE GRADO

101

5. El funcionario confirma el retiro.
6. El sistema retira el ítem de la plantilla retroalimentando al funcionario sobre el resultado de la transacción.

2.1 Precondiciones:

- El ítem está asociado a la plantilla.

2.2 Postcondiciones:

- EL ítem es retirado del listado en pantalla.

C-31 Realizar Inspección Preoperacional

Los conductores que tengan acceso al sistema deberán realizar la inspección preoperacional a sus vehículos autorizados antes de salir en operación.

Escenarios

1. Flujo Principal

1. El conductor ingresa a la opción de preoperacional, luego acciona el comando de crear una nueva.
2. El sistema muestra un formulario al usuario en donde pueda seleccionar el vehículo al cual le va a realizar la inspección.
3. El conductor selecciona el vehículo y confirma el formulario.
4. El sistema crea la inspección y le permite le muestra al conductor todos los ítems que debe revisar, permitiendo ejecutar tres comandos: uno para aprobar, otro para rechazar y otro para visualizar la ayuda escrita.

TRABAJO DE GRADO

102

5. El conductor inicia el proceso de inspección y marca en cada ítem el resultado de la revisión.

1.1 Precondiciones:

- El conductor debe contar con al menos un vehículo autorizado.

1.2 Postcondiciones:

- La inspección queda listada en pantalla con el estado finalizado.

C-32 Despachar Vehículo

Es el proceso de liberación de servicio. El despachador autoriza a operar un vehículo con su conductor sobre una ruta seleccionada.

Escenarios

1. Flujo Principal

1. El despachador ingresa al modulo de despachos.
2. El sistema muestra los últimos despachos realizados (si los hubiere) junto con la opción de realizar un nuevo despacho.
3. El despachador acciona el botón de nuevo despacho.
4. El sistema muestra un formulario en donde se puede elegir la fecha, las flotillas que con las que va a trabajar.
5. El despachador selecciona una o varias flotillas.
6. El sistema carga los vehículos asociados a esas flotillas en el listado de vehículos.
7. El despachador selecciona un vehículo.
8. El sistema carga los conductores asociados a dicho vehículo.

TRABAJO DE GRADO

103

9. El despachador selecciona el conductor que se encuentra en ese momento y selecciona la ruta.
10. El sistema realiza en ese preciso momento una validación al vehículo, y al conductor y muestra información acerca de posibles problemas encontrados en validaciones del vehículo, validaciones del conductor, restricciones del conductor y fecha de descanso próxima; dando la opción de poder invocar una solicitud de extensión de jornada.
11. El despachador valida que todo se encuentre en orden, escribe notas si las hubiere en el campo de notas y pincha sobre el botón de “despachar”.
12. El sistema registra el despacho y permite la opción de impresión de la tarjeta de despacho.
13. El despachador ejecuta el proceso de impresión.

1.1 Precondiciones:

- El conductor debe estar registrado en el sistema.
- El vehículo debe estar registrado en el sistema y asociado al conductor que se va a despachar.
- Debe existir al menos una ruta asociada a la flotilla que pertenece el vehículo.

1.2 Restricciones:

- El vehículo y el conductor deben contar con los documentos marcados como “requerido para operar”.
- Si existen tipologías marcadas como “Permite operar si falta” a pesar de que sean “requeridos para operar” el sistema no restringe el despacho si este documento **nunca** ha sido cargado.
- Los documentos no pueden estar vencidos si el documento tiene vencimiento.
- El conductor no puede estar bloqueado por los días de fatiga máxima.
- El conductor y el vehículo deben estar en estado activo.

1.3 Postcondiciones:

- El despacho se muestra en pantalla como parte de los últimos despachos realizados.

C-33 Realizar Inspección Vehicular

Los inspectores pueden realizar revisiones mas profundas a los vehículos, anticipando fallas potenciales y creando reglas de cumplimiento que tienen fechas de vencimiento.

Escenarios

1. Flujo Principal

1. El inspector ingresa a la opción de revisiones, luego acciona el comando de crear una nueva.
2. El sistema muestra un formulario en donde puede seleccionar el vehículo al cual le va a realizar la inspección, el conductor asociado presente en la inspección y una fecha de vencimiento de la inspección.
3. El inspector diligencias los datos y acciona el botón de crear revisión.
4. El sistema crea la inspección y le muestra al inspector todos los ítems que debe revisar, permitiendo ejecutar tres comandos: uno para aprobar, otro para rechazar y otro para visualizar la ayuda escrita.
5. El inspector inicia el proceso de inspección y marca en cada ítem el resultado de la revisión.

1.1 Precondiciones:

- El vehículo esta registrado en el sistema.

TRABAJO DE GRADO

105

- El conductor esta registrado en el sistema y asociado al vehículo que se va a inspeccionar.
- Existe una plantilla configurada para el tipo de vehículo que se va a inspeccionar.

1.2 Postcondiciones:

- La revisión queda listada en pantalla.

Tesorería y Cartera

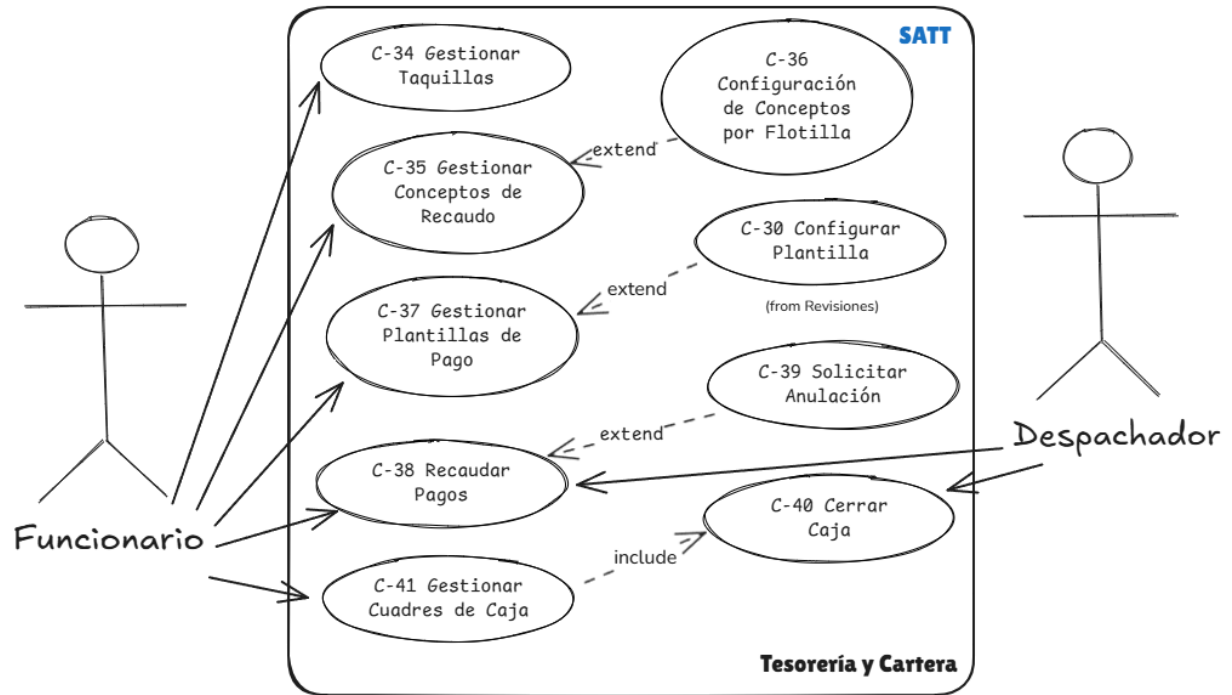


Figura 5. Casos de uso de Tesorería y Cartera

C-34 Gestionar Taquillas

Las taquillas son las estaciones de recaudo que están habilitadas para recibir dinero. Es fundamental para el negocio gestionarlas como maestro.

Escenarios

1. Crear Taquilla

1. El funcionario ingresa al modulo de taquillas, luego acciona el comando de crear nueva.
2. El sistema muestra un formulario en donde puede ingresar el nombre de la taquilla, la IP address, un DDNS opcional y una descripción también opcional.

TRABAJO DE GRADO

107

3. El funcionario diligencia los datos y acciona el botón de crear taquilla.
4. El sistema guarda y los datos y retroalimenta el resultado de la transacción al funcionario.

1.1 Precondiciones:

- El nombre de la taquilla no está registrado previamente

1.2 Postcondiciones:

- La taquilla creada se lista en pantalla

2. Editar Taquilla

1. El funcionario ingresa al módulo de taquillas
2. El sistema muestra el listado de taquillas registradas colocando en cada registro la opción de editar.
3. El funcionario selecciona el registro que desea editar y acciona el botón de edición.
4. El sistema muestra un formulario con la información precargada que el funcionario puede editar.
5. El funcionario realiza los cambios necesarios y acciona el botón de guardado.
6. El sistema modifica los datos retroalimentando al funcionario sobre el resultado de la transacción.

2.1 Precondiciones:

- La taquilla está registrada previamente.

2.2 Restricciones:

- Si el funcionario cambia el nombre de la taquilla, el nuevo nombre no debe coincidir con el de otra taquilla previamente guardada.

3. Eliminar Taquilla

1. El funcionario ingresa al módulo de taquillas
2. El sistema muestra el listado de taquillas registradas colocando en cada registro la opción de eliminar.
3. El funcionario selecciona el registro que desea eliminar y acciona el botón de eliminación.
4. El sistema solicita confirmación por parte del funcionario.
5. El funcionario confirma la eliminación.
6. El sistema realiza el borrado y retroalimenta al funcionario sobre el resultado de la transacción.

3.1 Precondiciones:

- La taquilla debe existir y no estar asociada a otros recursos como pagos (por restricciones de clave foránea).

3.2 Postcondiciones:

- La taquilla se elimina del listado en pantalla.

4. Inactivar / Activar Taquilla

1. El funcionario selecciona la taquilla que desea activar o desactivar, indicando al sistema el registro y dando la opción disponible que permite activar / desactivar.
2. El sistema cambia de estado alternando entre activo / inactivo según el caso y retroalimenta el cambio al funcionario.

4.1 Precondiciones:

- La taquilla debe existir.

4.2 Postcondiciones:

- Si la taquilla esta activa, cambiará a estado inactiva.

- Si la taquilla esta inactiva, cambiará a estado activa.

C-35 Gestionar Conceptos de Recaudo

Este maestro permite gestionar los ítems que serán usados para los pagos.

Escenarios

1. Crear Ítem

1. El funcionario ingresa al módulo de maestro de conceptos y selecciona la opción de crear o nuevo.
2. El sistema presenta un formulario en donde puede ingresar el nombre del ítem, un flag para indicar si el ítem es inamovible, un flag para indicar si se usará para llevar control de cartera y observaciones opcionales.
3. El funcionario diligencia los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- No puede existir otro ítem con el mismo nombre.

1.2 Postcondiciones:

- El ítem debe quedar listado en el sistema.

2. Editar Ítem

1. El funcionario selecciona el ítem que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información del ítem en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

TRABAJO DE GRADO

110

2.1 Restricciones:

- Si el funcionario cambia el nombre del ítem, este no debe coincidir con el nombre de otra ítem previamente guardado.

3. Eliminar Ítem

1. El funcionario selecciona el ítem que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema elimina la marca y retroalimenta al funcionario sobre el estado de la transacción.

3.1 Precondiciones:

- El ítem debe existir y no estar asociado a ningún otro registro (por llave foránea).

3.2 Postcondiciones:

- El ítem debe quitarse de la lista.

C-36 Configuración de Conceptos por Flotilla

Los conceptos pueden tener tarifas diferentes por cada flotilla, el sistema permitirá indicar estos valores a través del mismo módulo de conceptos.

1. Preconfigurar Concepto

1. El funcionario ingresa al módulo de maestro de conceptos.
2. El sistema lista los conceptos registrados agregando a cada concepto la opción de preconfigurar tarifas base.

TRABAJO DE GRADO

111

3. El funcionario acciona sobre el comando de configuración de cualquier concepto que desee.
4. El sistema ingresa a una subsección en donde podrá ver la configuración tarifaria según las flotas, si estas están definidas y presentará la opción de crear una nueva tarifa base.
5. El funcionario acciona sobre el botón de crear nueva tarifa base.
6. El sistema presenta un formulario en donde puede seleccionar la flotilla, luego ingresar el año que aplica y la tarifa.
7. El funcionario diligencia los datos y selecciona el botón crear tarifa.
8. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- El concepto debe estar creado.

1.2 Postcondiciones:

- La tarifa base queda listada en pantalla junto con las demás si las hubiere.

1.3. Restricciones:

- Solo es posible registrar una tarifa base por flota y año.

2. Editar Tarifa

1. El funcionario selecciona el ítem que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información del ítem en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

TRABAJO DE GRADO

112

- Si el funcionario cambia la flotilla y el año, este no debe coincidir con otro ítem idéntico previamente guardado.

3. Eliminar Tarifa

1. El funcionario selecciona el ítem que desea eliminar, indicando al sistema el registro y dando la opción disponible que permite eliminar.
2. El sistema realiza una confirmación de seguridad.
3. El funcionario confirma.
4. El sistema elimina la marca y retroalimenta al funcionario sobre el estado de la transacción.

3.1 Precondiciones:

- El ítem debe existir.

3.2 Postcondiciones:

- El ítem debe quitarse de la lista.

C-37 Gestionar Plantillas de Pago

Los vehículos antes de ser despachados deben pagar el dinero de administración y otros ítems, es fundamental para el negocio definir en plantillas lo que debe pagar cada vehículo al momento de iniciar el día laboral.

Escenarios

1. Crear Plantilla

1. El funcionario ingresa al módulo de plantillas de pago y selecciona la opción de crear o nuevo.

TRABAJO DE GRADO

113

2. El sistema presenta un formulario en donde puede seleccionar el vehículo.
3. El funcionario confirma los datos y selecciona el botón guardar.
4. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.

1.1 Precondiciones:

- El vehículo debe estar registrado y con un cupo asignado.
- El vehículo no debe tener una plantilla registrada previamente.

1.2 Postcondiciones:

- La plantilla queda listada en el sistema.

2. Editar Plantilla

1. El funcionario selecciona la plantilla que desea editar, indicando al sistema el registro y ejecutando la opción disponible que permite editar.
2. El sistema carga la información de la plantilla en un formulario y permite la edición.
3. El funcionario modifica los datos necesarios y aplica la opción de guardar.
4. El sistema retroalimenta el estado del proceso.

2.1 Restricciones:

- Si el funcionario cambia vehículo, éste no debe tener una plantilla previamente creada.

C-38 Recaudar Pagos

Los propietarios deben pagar unos valores administrativos, a través de sus vehículos, derivados de la relación contractual con la empresa. Es muy importante que el

TRABAJO DE GRADO

114

sistema permita recaudar estos valores de forma transparente y controlada para todas las partes.

Escenarios

1. Recaudar Pago Primario Normal (de un día)

1. El funcionario ingresa al módulo de pagos.
2. El sistema intenta identificar la estación basándose en su IP y posteriormente la coloca en una leyenda visible al funcionario
3. El funcionario selecciona la opción de crear o nuevo.
4. El sistema presenta un formulario en donde el funcionario puede seleccionar el tipo de pago (Primario o Secundario), variante Normal, fecha de cuota, pago de mantenimiento y el cupo del vehículo.
5. El sistema carga la plantilla para el vehículo en el año en que se encuentra seleccionada la fecha de cuota, mostrando los conceptos que debe pagar en pantalla y mostrando controles de edición y eliminación de conceptos según su configuración (flag inamovible). El sistema también presenta la opción de agregar mas conceptos con su valor de forma libre, calculando siempre el valor total de los conceptos agregados y unas notas opcionales.
6. El funcionario realiza los ajustes que considere (opcionalmente) y selecciona el botón crear pago.
7. El sistema crea el pago, notifica al funcionario el resultado exitoso del proceso y permite al funcionario imprimir el recibo del pago si este lo desea.

1.1 Precondiciones:

- El vehículo debe estar registrado y con un cupo asignado.
- El vehículo debe tener una plantilla configurada.
- El día de recaudo debe estar abierto.

1.2 Restricciones:

- El vehículo no puede tener un pago principal en la fecha seleccionada, solo se permite un pago de este tipo por fecha.
- Si el pago es marcado como *mantenimiento* no puede existir otro pago así en la misma semana.
- Si el pago es marcado como *mantenimiento* el sistema debe verificar que dicho vehículo no haya sido despachado en dicha fecha.

1.3 Excepciones:

- Si el funcionario agrega un concepto libre y este concepto se encuentra en la plantilla, no listará un concepto duplicado, sino que el valor se sumará al indicado en la plantilla.
- Si el pago es marcado como *mantenimiento* se debe eliminar los conceptos descontados por mantenimiento de la carga de plantilla.

1.4 Postcondiciones:

- El pago queda en la lista de pagos recientes con opciones adicionales que se pueden invocar.

2. Crear un pago Primario Múltiple (varios días)

1. El funcionario ingresa al módulo de pagos.
2. El sistema intenta identificar la estación basándose en su IP y posteriormente la coloca en una leyenda visible al funcionario
3. El funcionario selecciona la opción de crear o nuevo.

TRABAJO DE GRADO

116

4. El sistema presenta un formulario en donde el funcionario puede seleccionar el tipo de pago (Primario), variante Múltiple, fecha de cuota de inicio y fecha de cuota final y el cupo del vehículo.
5. El sistema carga la plantilla para el vehículo en el año en que se encuentra seleccionada la fecha de cuota, mostrando los conceptos con las sumas de los valores según el rango de fecha seleccionado, que debe pagar en pantalla y mostrando controles de edición y eliminación de conceptos según su configuración (flag inamovible). El sistema también presenta la opción de agregar más conceptos con su valor de forma libre, calculando siempre el valor total de los conceptos agregados y unas notas opcionales.
6. El funcionario realiza los ajustes que considere (opcionalmente) y selecciona el botón crear pago.
7. El sistema crea el pago, notifica al funcionario el resultado exitoso del proceso y permite al funcionario imprimir el recibo del pago si este lo desea.

2.1 Precondiciones:

- El vehículo debe estar registrado y con un cupo asignado.
- El vehículo debe tener una plantilla configurada.
- El día de recaudo debe estar abierto.

2.2 Restricciones:

- El pago múltiple debe contener un rango de fechas de un mismo mes.
- En los pagos múltiple no se permite manipular el flag de mantenimiento el cual debe estar en false u OFF.

2.3 Postcondiciones:

- El pago queda en la lista de pagos recientes con opciones adicionales que se pueden invocar.

2. Crear un pago Secundario

1. El despachador ingresa al módulo de pagos.
2. El sistema intenta identificar la estación basándose en su IP y posteriormente la coloca en una leyenda visible al funcionario
3. El funcionario selecciona la opción de crear o nuevo.
4. El sistema presenta un formulario en donde el despachador puede seleccionar el tipo de pago (Secundario), fecha de cuota, el cupo del vehículo y un control para agregar conceptos con su valor de forma libre, calculando siempre el valor total de los conceptos agregados y unas notas opcionales
5. El funcionario agrega los conceptos manualmente y selecciona el botón crear pago.
6. El sistema crea el pago, notifica al despachador el resultado exitoso del proceso y permite al funcionario imprimir el recibo del pago si este lo desea.

2.1 Precondiciones:

- El vehículo debe estar registrado y con un cupo asignado.
- El día de recaudo debe estar abierto.

2.2 Restricciones:

- En los pagos secundarios no se permite controlar la variante de pago ni el descuento por mantenimiento.

2.3 Postcondiciones:

- El pago queda en la lista de pagos recientes con opciones adicionales que se pueden invocar.

3. Eliminar Pago

1. El funcionario ingresa al módulo de pagos.

TRABAJO DE GRADO

118

2. El sistema intenta identificar la estación basándose en su IP y posteriormente la coloca en una leyenda visible al funcionario, luego lista los pagos más recientes con la opción de eliminar si su estado lo permite.
3. El funcionario selecciona el pago y acciona el botón de eliminar.
4. El sistema solicita al funcionario confirmación.
5. El funcionario confirma.
6. El sistema realiza la anulación del pago.

3.1 Precondiciones:

- Debe existir el pago en estado creado.
- El pago no fue impreso.

3.2 Restricciones:

- Solo se permite borrar si el usuario que solicita el borrado es el mismo que lo creó.

3.3 Postcondiciones:

- El pago queda en la lista de pagos recientes con el estado *anulado*.

C-39 Solicitar Anulación de Pago

Cuando los pagos son impresos, no se puede anular de forma simple, sino que un funcionario de rango mayor debe revisar y proceder con la anulación. Es por eso que el funcionario recaudador o despachador debe solicitar la anulación, a través del sistema, para estos casos.

Escenarios

1. Flujo Principal

TRABAJO DE GRADO

119

1. El funcionario ingresa al módulo de pagos.
2. El sistema detecta la estación a través de la IP y la muestra en una leyenda visible al funcionario y lista los pagos recientes con la opción de *anular* si el pago está en estado impreso o expedido.
3. El funcionario acciona el botón o comando para anular pago.
4. El sistema muestra un formulario al funcionario en donde puede escribir el motivo de la anulación.
5. El funcionario ingresa las explicaciones a las que haya lugar y acciona el botón de solicitar.
6. El sistema guarda la información y notifica al funcionario el resultado exitoso del proceso.
7. El sistema notifica al funcionario de nivel superior (que tenga el recurso asignado) sobre una nueva anulación de pago.

1.1 Precondiciones:

- El pago esta creado en el sistema en estado impreso o expedido.

1.2 Postcondiciones:

- El pago queda listado en pantalla, pero con estado “Pendiente de Anulación”.

C-40 Cerrar Caja

Los despachadores podrán revisar en cualquier momento el estado de su recaudo. Esto les indica con cuánto dinero debe contar su caja para su posterior entrega a la oficina central.

Escenarios

1. Flujo Principal

1. El despachador ingresa al módulo de cuadro de caja.
2. El sistema hace un listado de sus cuadros más recientes por fecha y con la opción de imprimir en cada cuadro.
3. El despachador acciona el botón de imprimir sobre el cuadro que le interesa cerrar.
4. El sistema genera la impresión del cuadro con un token generado y cierra el día para ese usuario y estación.

1.1 Precondiciones:

- Debe existir el cuadro que desea cerrar.

1.2 Postcondiciones:

- La fecha queda cerrada para el usuario, quedando listada en pantalla, pero con la leyenda “cerrada”.
- El sistema no permitirá el ingreso de más pagos en la fecha y estación del cuadro impreso.

C-41 Gestionar Cuadros de Caja

Los funcionarios con permisos más elevados que el despachador, podrán revisar las cajas de todos los recaudadores. Esto les permitirá llevar un control de los dineros que estén soportados en pagos pero que no hayan entrado a las arcas de las empresas. Por su parte, los despachadores podrán tener un control sobre las entregas de dinero que hacen en las oficinas principales. Permitiendo la transparencia y la sana cultura dentro de la empresa.

Escenarios

1. Reimprimir Cuadre

1. El funcionario ingresa al módulo de cuadro de caja.
2. El sistema hace un listado de todos los cuadros de caja de los despachadores más recientes por fecha y con la opción de imprimir en cada cuadro.
3. El funcionario acciona el botón de imprimir sobre el cuadro que le interesa revisar.
4. El sistema genera la impresión del cuadro.

1.1 Precondiciones:

- Debe existir el cuadro que desea reimprimir.

1.2 Postcondiciones:

- No se altera nada en base de datos.

2. Recaudar Cuadre de Caja

1. El funcionario ingresa al módulo de cuadro de caja.
2. El sistema hace un listado de todos los cuadros de caja de los despachadores más recientes por fecha y con la opción de recaudar en cada cuadro.
3. El funcionario acciona el botón de recaudar sobre el cuadro de interés.
4. El sistema muestra un formulario en donde solicita el token que debe llevar el cuadro impreso.
5. El funcionario ingresa el token impreso en el cuadro y acciona el botón recaudar.
6. El sistema valida el token y hace el ingreso de recaudo.

2.1 Precondiciones:

- El cuadro debe existir impreso y cerrado.
- El cuadro no ha sido recaudado previamente.

2.2 Postcondiciones:

- El sistema muestra en la lista el cuadro recaudado agregándole el estado de “recaudado por” y el nombre del funcionario quien hizo el recaudo además de la fecha en que se hizo.

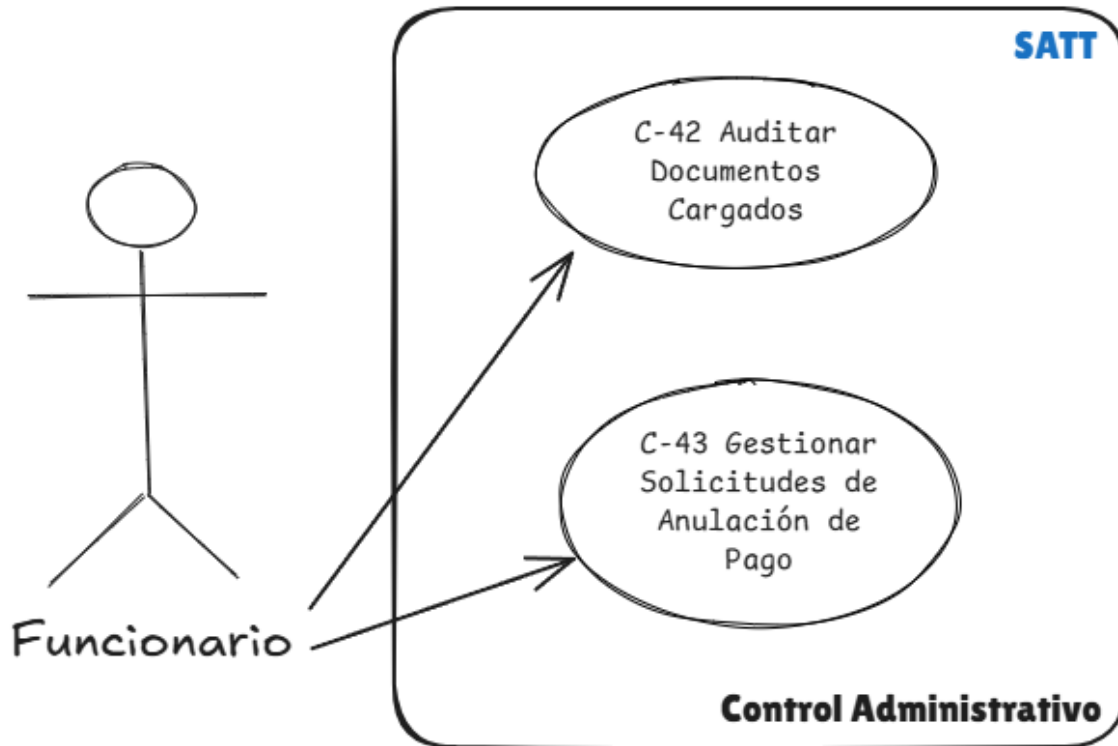


Figura 6. Casos de uso de control administrativo

C-42 Auditar Documentos Cargados

Los funcionarios con ciertos permisos elevados pueden realizar auditoria a los documentos que otros funcionarios cargan tanto a vehículos como a empleados. Es fundamental para la empresa contar con esta funcionalidad para mantener una segunda capa de control.

Escenarios

1. Aprobar Documento

TRABAJO DE GRADO

124

1. Un funcionario carga un documento al sistema. Bien sea para un vehículo o un empleado.
2. El sistema notifica a los usuarios aprobadores sobre un documento nuevo cargado.
3. El funcionario podrá revisar la notificación y entrar al modulo de aprobaciones en donde podrá revisar los últimos documentos pendientes por auditoria. Luego selecciona el documento que va auditar dando clic en el comando correspondiente de revisión.
4. El sistema cargará en pantalla el documento junto con los datos diligenciados por parte del funcionario responsable de la carga.
5. El funcionario revisa que los datos coincidan: fecha de inicio de vigencia, sujeto, fecha de vencimiento y otros datos adicionales que pueda tener el documento.
6. El funcionario acciona la opción de aprobar.
7. El sistema retroalimenta el resultado de la transacción.

1.1 Precondiciones:

- El documento debe estar cargado sin auditar.

1.2 Postcondiciones:

- El documento queda en estado de aprobado interno en el sistema sin ningún otro cambio aparente para el funcionario que carga.

2. Rechazar Documento

1. Un funcionario carga un documento al sistema. Bien sea para un vehículo o un empleado.
2. El sistema notifica a los usuarios aprobadores sobre un documento nuevo cargado.
3. El funcionario podrá revisar la notificación y entrar al módulo de aprobaciones en donde podrá revisar los últimos documentos pendientes por auditoria. Luego selecciona el documento que va auditar dando clic en el comando correspondiente de revisión.

TRABAJO DE GRADO

125

4. El sistema cargará en pantalla el documento junto con los datos diligenciados por parte del funcionario responsable de la carga.
5. El funcionario revisa que los datos coincidan: fecha de inicio de vigencia, sujeto, fecha de vencimiento y otros datos adicionales que pueda tener el documento.
6. El funcionario acciona la opción de rechazar.
7. El sistema solicita ingresar el motivo del rechazo.
8. El funcionario ingresa el motivo del rechazo y lo confirma.
9. El sistema notifica al usuario que cargó el documento sobre la anomalía y da de baja al documento cargado.
10. El sistema retroalimenta el resultado de la transacción.

2.1 Precondiciones:

- El documento debe estar cargado sin auditar.

2.2 Postcondiciones:

- El documento queda en estado de rechazado, eliminándolo de la lista de documentos cargados bien sea del vehículo o del empleado.

C-43 Gestionar Solicitudes de Anulación de Pago

Los funcionario o despachadores pueden cometer errores al generar un pago, esto debería poderse corregir. Sin embargo, debe tener un control y un segundo agente que revise el motivo de la anulación y si esta procede. Por norma el sistema no permitirá anular y aprobar la anulación sin que intervengan dos actores.

Escenarios

TRABAJO DE GRADO

126

Flujo Principal

1. El sistema notifica al funcionario(s) aprobador(es) sobre una solicitud por parte de otro actor.
2. El funcionario ingresa a la notificación y verifica la solicitud.
3. El sistema realiza una lista de las solicitudes pendientes de gestión y colocando en cada registro un botón para aprobar y otro para rechazar.
4. El funcionario ejecuta cualquiera de las dos opciones.
5. El sistema gestiona la solicitud y notifica sobre el resultado de la transacción al funcionario aprobador.

Precondiciones:

- Existe una solicitud de anulación de pago pendiente de aprobación

Postcondiciones:

- Si la solicitud fue aprobada el pago quedará en estado *Anulado*.
- Si la solicitud fue rechazada el pago quedará en estado *Expedido*.

Descripción de los Actores

Administrador

Es la persona que tiene todos los permisos y recursos del sistema. Básicamente tiene conocimientos avanzados para dar soporte al sistema y solucionar cualquier inconveniente que esta pueda generar sobre su funcionamiento. Además, administra los usuarios y permisos de todos los usuarios.

Funcionario

TRABAJO DE GRADO

127

Representa cualquier empleado de la empresa que tenga un rol específico dentro de la función administrativa. Por ejemplo, puede ser un contador, un profesional SST o cualquier otro profesional, tecnólogo o técnico contratado por la empresa para una labor específica de área.

Despachador

Es un funcionario con un cargo muy bien definido y un protagonismo fundamental en el área de negocio. Su función es liberar el servicio de transporte.

Conductor

Es el eslabón más operativo de la empresa. Son las personas que guían los vehículos y ejecutan la labor misional más importante: transportar personas.

TRABAJO DE GRADO

128

Fase de diseño

Decisiones Arquitectónicas (ADR)

ADR-01: Arquitectura Modular Basada en Capas (Next.js)

Contexto

EL sistema administrativo de transporte terrestre pose necesidad de rendimiento, adaptación a dispositivos móviles, facilidad de escalado, mantenibilidad, fiabilidad y seguridad de alto nivel. Satisfacer requisitos no funcionales RQNF02, RQNF03, RQNF04, RQNF05, RQNF06.

Alternativas

- **Laravel + Blade:** Separación poco clara entre backend y frontend. Limitaciones para usabilidad avanzada en ciertas interfaces.
- **Microservicios:** Complejidad avanzada para crear orquestadores y no adaptable en la restricción del cliente que era aprovechar el servidor físico de la compañía.

Decisión

Utilizar una arquitectura en capas implementada con el framework Next.js versión 14.

Esta estructura permite adaptar estrategias de diferentes patrones de diseño. Cuenta con una capa de Presentación bien definida por componentes de React.js. Una capa de

TRABAJO DE GRADO

129

lógica de negocio que usa rutas API internas y Server Actions que procesan la información con una integración excepcional y una capa de datos que está gestionada por Prisma ORM quien abstrae y define perfectamente las entidades del modelo de datos, logrando un uso fácil, seguro y fiable a través del tipado correcto de los campos, relaciones y restricciones de integridad en las tablas de MySQL.

En conjunto se puede implementar un patrón de diseño cliente servidor con alta carga en el servidor para permitir a los clientes tener pocos recursos computacionales.

Consecuencias Positivas

- ✓ Una separación clara de responsabilidades facilitando la mantenibilidad.
- ✓ Un solo lenguaje de programación (Typescript – Javascript) para todo el conjunto fullstack lo que permite tener curvas de aprendizaje controladas.
- ✓ Buena comunidad y componentes en el frontend que puedan ser integrados.
- ✓ Buena documentación para el aprendizaje.
- ✓ De código abierto y cero pagos por licenciamiento del SDK.
- ✓ Autenticación robusta y relativamente fácil de implementar.

Consecuencias Negativas

- Cierta curva de aprendizaje para entender la filosofía de server components y SSR.
- Las API Routes tienen ciertas limitaciones frente a controlador más dedicados en framework tradicionales.
- Auth.js puede contener cierta magia no controlada.
- En algún momento Vercel podría cobrar por el uso del framework.
- Versiones futuras del framework podrían ser difícilmente actualizables por romper la compatibilidad con esta versión.

ADR-02: Selección de MySQL como sistema gestor de base de datos

Contexto

El sistema SATT requiere almacenar información relacionada en bases de datos. Estas bases de datos deben tener altos estándares de integridad y permitir una gestión fiable con rendimiento sobresaliente. Esto en cumplimiento de requisitos no funcionales RQFN02, RQFN03.

Alternativas

- **PostgreSQL.** Muy robusto pero el equipo no contaba con conocimientos necesarios. Esto supondría una curva de aprendizaje adicional al framework.
- **SQLite.** Muy simple, alta portabilidad, pero con poca capacidad para gestionar concurrencia, transacciones y restricciones de integridad. Fiabilidad cuestionable para este tipo de aplicaciones.
- **MariaDB.** Buena alternativa; con menos madurez y poca compatibilidad con Prisma ORM.

Decisión

Usar MySQL, porque el equipo cuenta con experiencia en su uso, en su programación y las herramientas necesarias para el diseño, despliegue y mantenimiento de modelos de datos. Buena concurrencia, fuertes relaciones de integridad y restricciones a través de su motor InnoDB. Además, es compatible con Prisma ORM y es fácilmente instalable en servidores propios.

Consecuencias Positivas

TRABAJO DE GRADO

131

- ✓ Historial y trayectoria en el mundo de la tecnología al ser un motor usado desde antaño con buenos estándares de calidad.
- ✓ Curva de aprendizaje nula al contar con experiencia y conocimientos entre el equipo de desarrollo.
- ✓ Rendimiento y madurez sobresalientes en entornos de carga importantes sin ser extremos.
- ✓ Herramientas fantásticas como MySQL Workbench para gestionar los servidores, usuarios y actividades de DBA avanzados con facilidad inigualable.
- ✓ Comunidad y documentación amplia.

Consecuencias Negativas

- Proyecto comprado por Oracle que en su mayoría de productos exige pago de licencias. Podría en el futuro volverlo de código cerrado y de pago.
- Menor robustez comparado con PostgreSQL para entornos de mucha carga y ultra avanzados.

Modelo de Arquitectura

Teniendo en cuenta los drivers arquitectónicos, la tecnología debe permitir portabilidad, con un rendimiento aceptable, mantenible, fiable y altos estándares de seguridad. Esto se puede lograr, entre varias opciones disponibles, con el framework NextJS; un framework que provee buen SEO, es escalable y con una plataforma en constante crecimiento. Además, nos permite usar diferentes estrategias de renderizado según la necesidad y el contexto. SSR (Server Side Rendering), SSG (Static Site Generation) y CSR (Client Side Rendering).

En el apartado de seguridad, contamos también con un ecosistema robusto gracias a AuthJS, tecnología que se puede implementar como artefacto integrable a la aplicación con políticas que permiten la autenticación y la autorización. Además, se puede escalar, en el futuro, con mecanismos de autenticación OAuth y MFA.

La siguiente figura da un panorama de la arquitectura de aplicación:

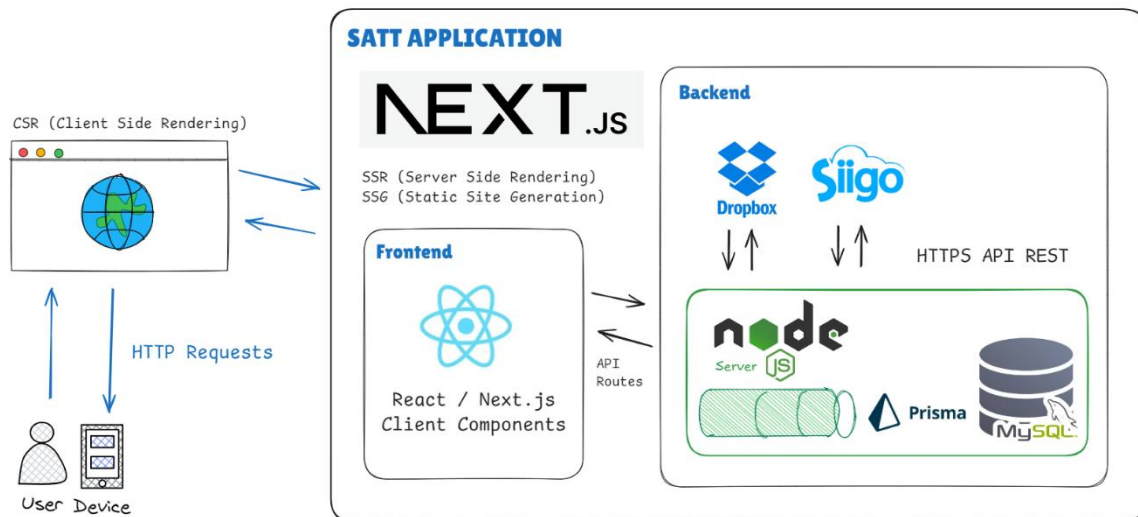


Figura 7. Diseño de arquitectura de SATT

En la imagen anterior se puede observar que la aplicación estará desarrollada en el framework Next.js en su versión 14. Por su parte, el backend que corre principalmente sobre el Event Loop de Node.js tendrá puntos de entrada, al estilo API, que solo podrán ser consumidas por componentes cliente de React. El framework está optimizado para renderizar inicialmente bajo la filosofía SSR (Server Side Rendering), sin embargo, en muchas partes de la interfaz se requiere hidratación y funcionalidad javascript, por lo que se le indican al framework con la directiva “use client” para que sea el cliente (navegador) quien controle estos eventos y funcionalidad. A pesar de ello, todo esto es totalmente transparente para el usuario quien su experiencia de uso será la mejor posible.

El backend estará conectado al servidor de MySQL a través de Prisma ORM el cual permite versatilidad de desarrollo mapeando con tipado de sus tablas, ya que se usará TypeScript – JavaScript como lenguaje único para toda la aplicación.

Adicionalmente, la tecnología permite conectarse a otros servicios a través de API al ecosistema (npm) contar con un abanico extensos de utilidades, y en este caso en particular, contaremos con Axios como cliente de peticiones en todo el proyecto.

Modelo de Datos

Control de Acceso

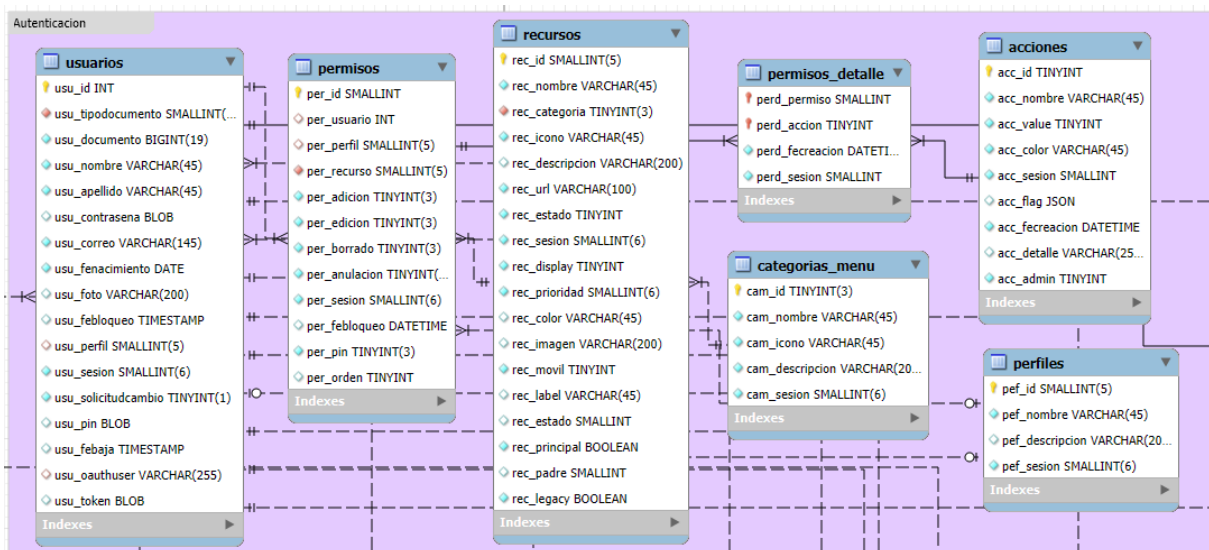


Ilustración 8. Modelo de Autenticación

Este modelo es el encargado de gestionar los usuarios y los permisos asociados a cada usuario. Está conformado por las siguientes tablas:

Tabla 5. Tablas de Control de Acceso

Tabla	Relacionada con	Finalidad
Acciones	Permisos Detalle (1: N)	Es un maestro que permite definir cada acción sobre un recurso, como lo es leer, crear, actualizar, entre otros.
Categorías Menú	Recursos (1: N)	Maestro que permite dividir el menú en categorías para dar seccionamiento a los recursos del sistema.

TRABAJO DE GRADO

135

Recursos	Permisos (1: N) Categorías Menú (1: N)	Son los diferentes módulos o recursos que pueden asignarse a un usuario. Esta tabla es controlada por el desarrollador, ya que cualquier funcionalidad asignable a un usuario deberá agregarse a esta tabla. Contiene las URL internas para acceder a los módulos ya desarrollados.
Permisos	Usuarios (N: 1) Recursos (N: 1)	Los permisos definen la relación entre un usuario y un recurso. Cada que se asigna un recurso a un usuario debe existir un registro indicándolo en esta tabla.
Permisos Detalle	Permisos (N: 1) Acciones (N: 1)	Es una tabla intermedia utilizada para normalizar la relación muchos a muchos entre Acciones y Permisos.
Usuarios	Permisos (1: N)	Son cada uno de los actores que tienen acceso al sistema. Tiene información básica que será usada por el sistema para identificarlo durante el proceso de autenticación y autorización.

TRABAJO DE GRADO

137

Control de Empleados

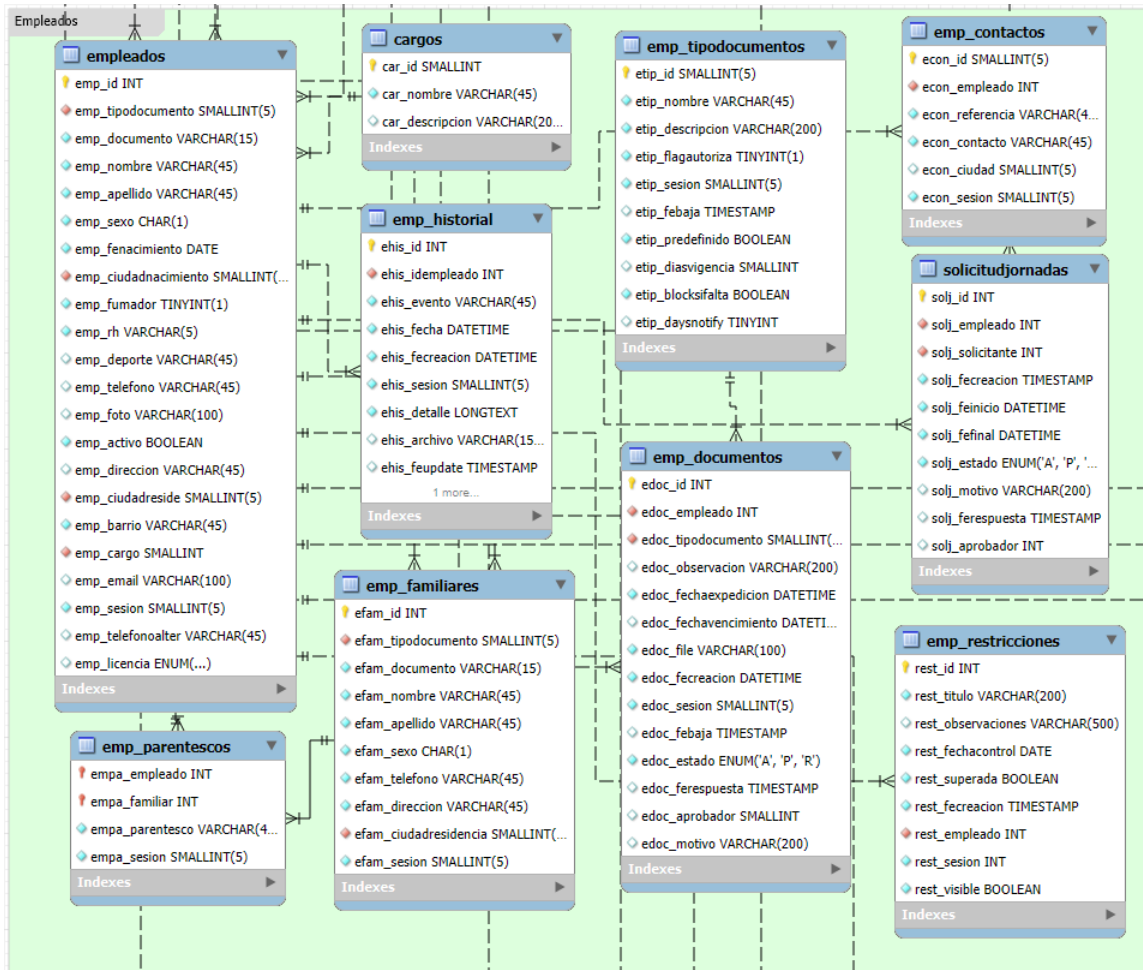


Ilustración 10. Modelo de Empleados

Este modelo, como su nombre lo indica, configura el ecosistema necesario para la funcionalidad de control del personal de la compañía. Está compuesto por las siguientes tablas:

TRABAJO DE GRADO

138

Tabla 6. Tablas de control de Empleados

<i>Tabla</i>	<i>Relacionada con</i>	<i>Finalidad</i>
Cargos	Empleados (1: N)	Es un maestro que permite definir los cargos de los colaboradores.
Empleados	emp_historial (1: N) emp_documentos (1: N) emp_restricciones (1: N) solicitudjornadas (1: N) autorizaciones (1: N) revisiones (1: N)	Permite almacenar los diferentes empleados con los que cuenta la compañía. Es una entidad relacionada en muchas mas tablas dada su importancia para el proceso operativo del negocio.
Emp_tipodocumentos	Emp_documentos (1: N)	Es un maestro que permite definir las tipologías documentales que se podrán cargar a los documentos de empleados.
Emp_documentos	Emp_tipodocumentos (N: 1) Empleados (N: 1)	En esta tabla se almacenan los documentos cargados a cada empleado asociando su tipología y demás metadatos como la fecha de vencimiento.
Emp_restricciones	Empleados (N: 1)	Permite almacenar las restricciones que se impongan al empleado durante su vida laboral.

TRABAJO DE GRADO

139

Control de Vehículos

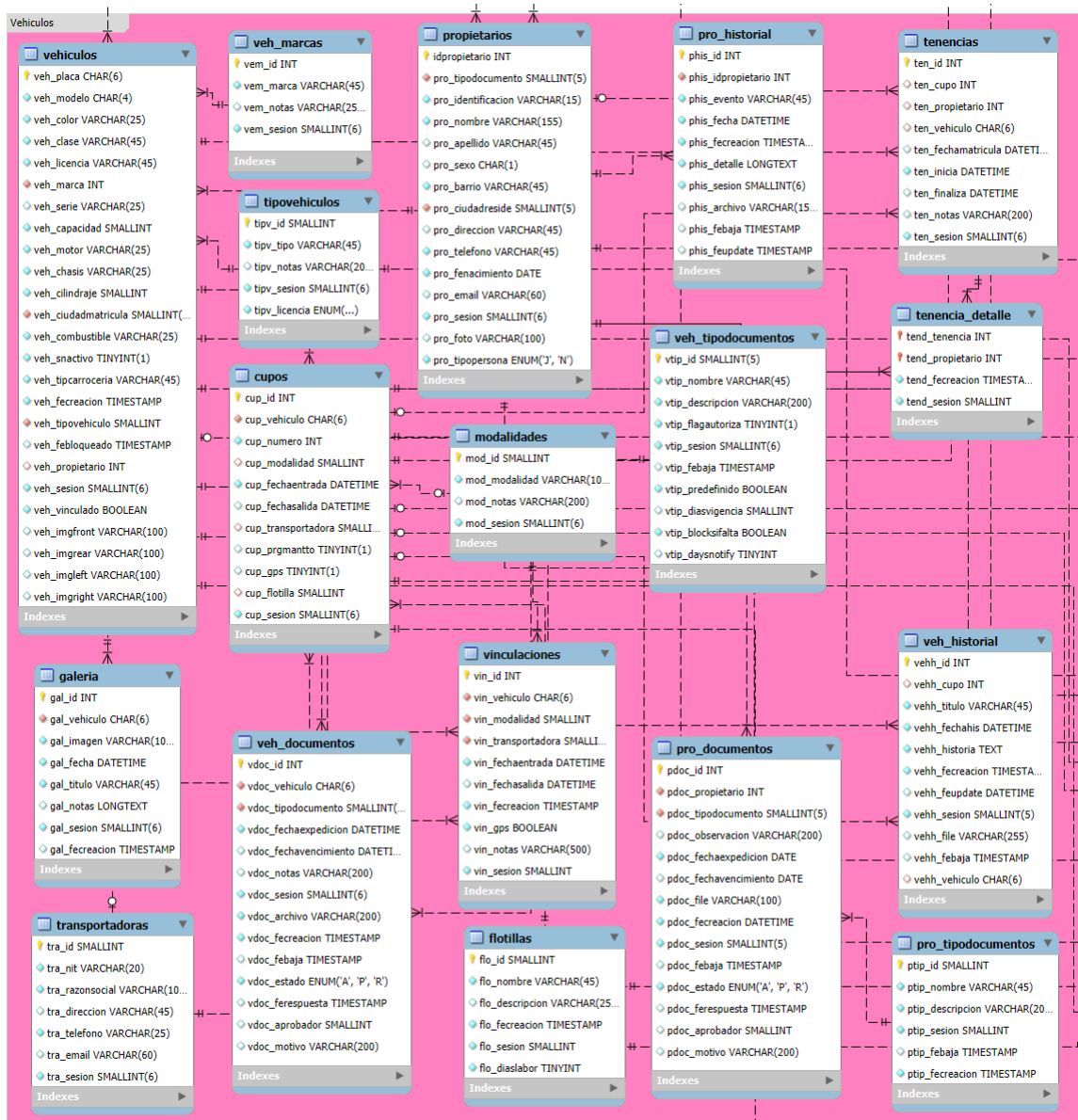


Ilustración 11. Modelo de Vehículos

En esta sección se abordaron las tablas fundamentales para la gestión del ecosistema de vehículos. Está conformado por las siguientes tablas:

TRABAJO DE GRADO

140

Tabla 7. Tablas de control de Vehículos.

Tabla	Relacionada con	Finalidad
Transportadoras	Vinculaciones (1: N)	Es el maestro que alberga todas las empresas de transporte en donde se tenga una relación comercial.
Vehículos	Propietarios (N: 1) Marcas (N: 1) tipovehiculos (N: 1) vinculaciones (1: N) cupos (1: N) revisiones (1: N) despachos (1: N) veh_documentos (1: N) autorizaciones (1: N) tenencias (1: N) veh_historial (1: N)	En esta tabla se almacenan todos los automotores con sus características técnicas como tipo de combustible, no de motor, chasis, capacidad de pasajeros cilindraje y en esencia lo que refleja una tarjeta de propiedad.
Tipovehiculos	Vehiculos (1: N)	En esta tabla se almacenan todas las tipologías vehiculares avaladas por el ministerio de transporte como es: Camioneta, Buseta, Microbus, etc.
Flotillas	Cupos (1: N)	En esta tabla se almacenarán las flotillas con las que cuenta la empresa como, por ejemplo: San José de Apartadó.
Veh_marcas	Vehículos (1: N)	Maestro que alberga las marcas de los vehículos. Necesario por políticas de normalización.
Propietarios	Vehículos (1: N)	Contiene la información básica de los propietarios; quienes no son actores iniciales del sistema, pero sí del modelo de negocio.

TRABAJO DE GRADO

141

Tenencias	Vehiculos (1: N) Propietarios (1: N)	Esta tabla permite registrar el historial de propietarios para un vehículo. Gracias a los campos de fecha se puede armar la historia de cuando un vehículo es traspasado entre los diferentes propietarios en toda la historia dentro de la compañía.
Modalidades	Vinculaciones (1: N)	Es un maestro que clasifica las modalidades como, por ejemplo, urbano, intermunicipal, mixto, entre otros.
Vinculaciones	Modalidades (N: 1) vehículos (N: 1) Transportadora (N: 1)	En esta tabla se almacena toda la información de vinculación de un vehículo. Permite conocer su estado y si ha tenido más de una vinculación en tiempos diferentes, aunque siendo este caso un poco raro, pero podría ser posible.
Veh_tipodocuments	Veh_documentos (1: N)	Maestro que define las tipologías documentales que se pueden cargar a un vehículo.
Veh_documentos	Vehiculos (N: 1) Veh_tipodocuments (N: 1)	En esta tabla se pueden almacenar los metadatos de todos los documentos pertenecientes a un vehículo.

Revisiones y Logística

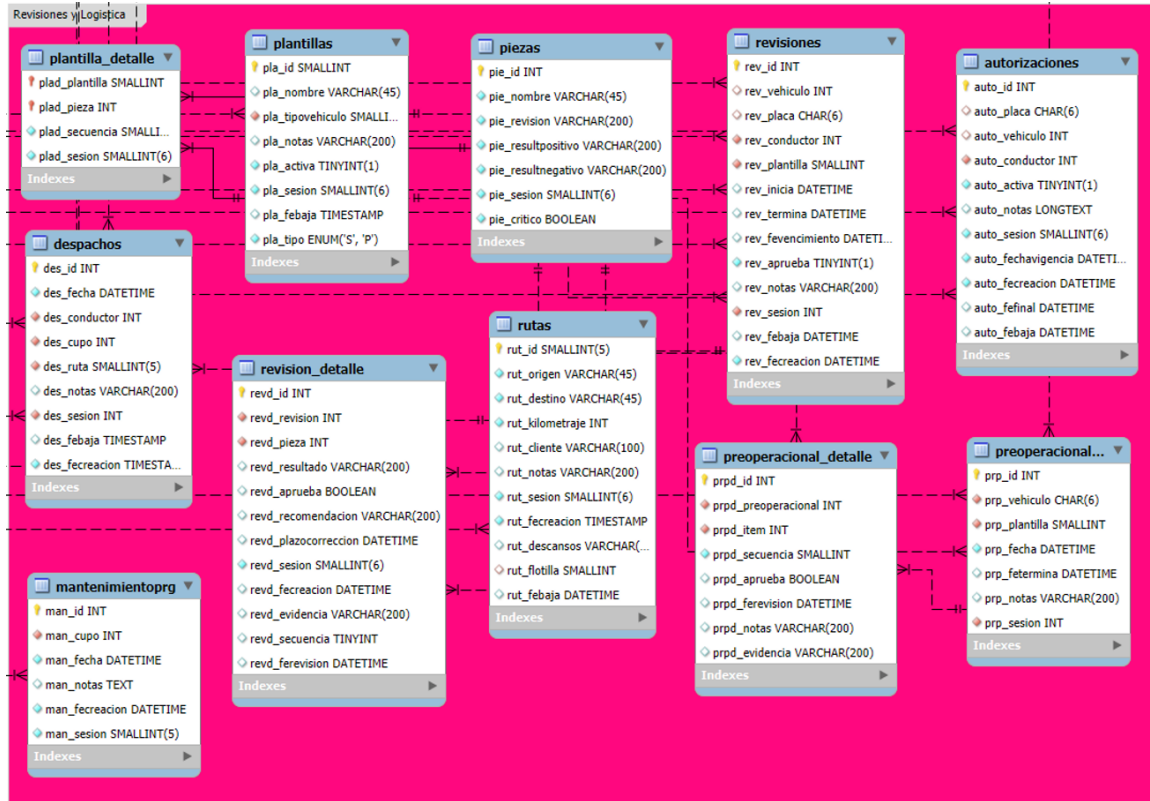


Ilustración 12. Modelo de Revisión y Logística

Este modelo está compuesto por las siguientes tablas:

Tabla 8. Tablas de Revisiones y Logística

Tabla	Relacionada con	Finalidad
Piezas	Plantilla_detalle (1: N) Revisión_detalle (1: N) Preoperacional_detalle (1: N)	Este maestro tiene los ítems que intervienen en una revisión vehicular. Estas mismas se asocian a las plantillas.

TRABAJO DE GRADO

143

Plantillas	Plantilla_detalle (1: N) Preoperacionales (1: N) Revisiones (1: N)	Almacena la información de las plantillas necesarias para las revisiones tanto técnicas como preoperacionales.
Plantilla_detalle	Plantillas (N: 1) Piezas (N: 1)	Permite configurar la plantilla, asociando los ítems o piezas con las plantillas. Es necesaria la tabla para normalizar la relación N: N.
Revisiones	vehículos (N: 1) Empleados (N: 1) Plantillas (N: 1)	En esta tabla se registran las revisiones técnicas de los vehículos realizadas por los inspectores en presencia del conductor.
Revision_detalle	Revisiones (N: 1) Piezas (N: 1)	Es la tabla que registra el resultado detallado de la revisión realizada. Asocia cada ítem con los datos de la revisión. Necesaria para evitar redundancia en los datos.
Preoperacionales	Usuarios (N: 1) vehículos (N: 1) Plantillas (N: 1)	Esta tabla almacena la inspección de alistamiento diaria de cada vehículo llevada a cabo por su conductor autorizado.
Preoperacional_detalle	Preoperacional (N: 1) Piezas (N: 1)	La tabla registra el resultado detallado de la inspección preoperacional, asociando cada ítem con los datos de la preoperacional. Se requiere para normalizar relación muchos a muchos (N: N)
Autorizaciones	vehículos (N: 1) Empleados (N: 1)	Esta tabla permite asociar vehículos y conductores. Es fundamental para mantener el control de conductores autorizados sobre los vehículos.
Rutas	Flotillas (1: N) Despachos (1: N)	Maestro que define los trayectos autorizados a la empresa de transporte.
Despachos	Cupos (N: 1) Empleados (N: 1) vehículos (N: 1) Rutas (N: 1)	En esta tabla se registran los despachos que tiene cada vehículo con su conductor autorizado sobre las rutas autorizadas.

Tesorería y Cartera

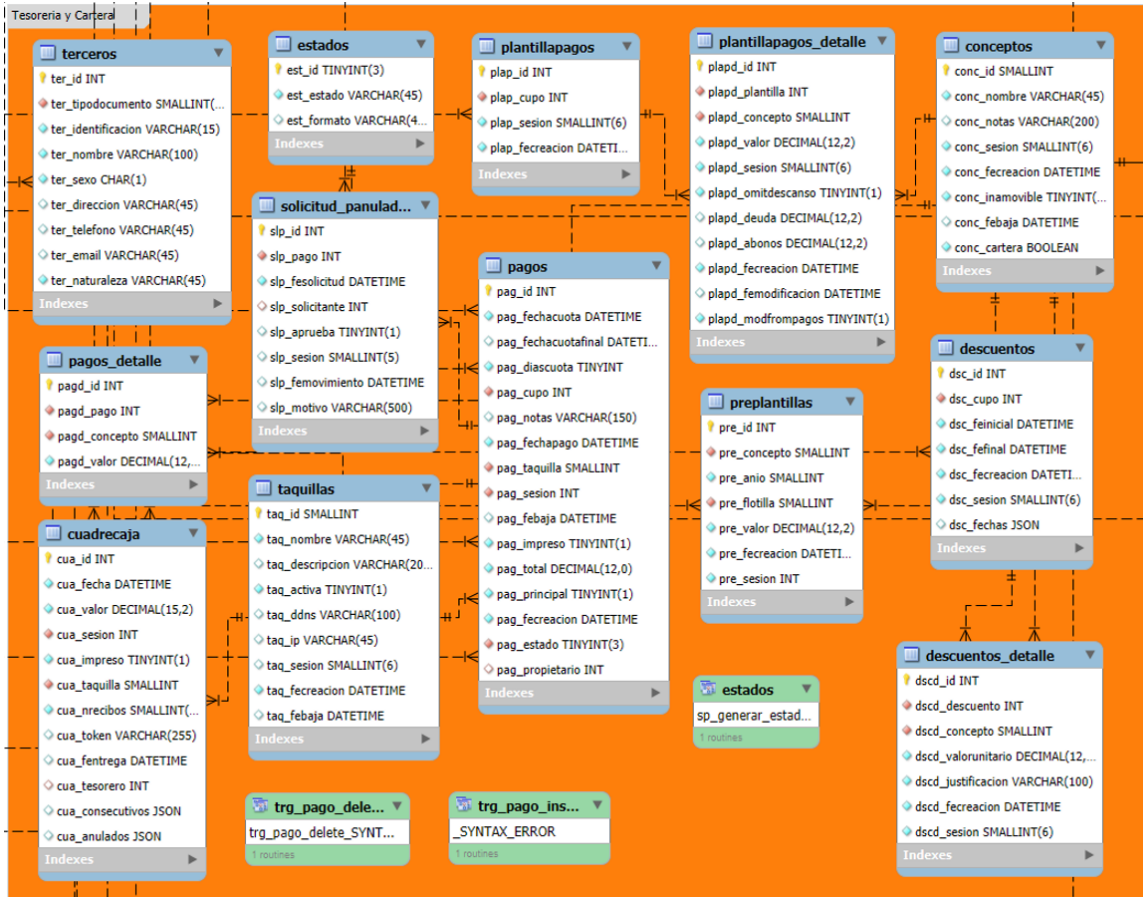


Ilustración 13. Modelo de Tesorería y Cartera

Este ecosistema alberga lo necesario para controlar el tesoro de la compañía. Está compuesto por las siguientes tablas:

Tabla 9. Tablas de Tesorería y Cartera

Tabla	Relacionada con	Finalidad
Conceptos	Pagos_detalle (1: N) Plantillapagos_detalle (1: N) Preplantillas (1: N)	Este maestro contiene los conceptos de tesorería. Necesarios para todo el ecosistema.

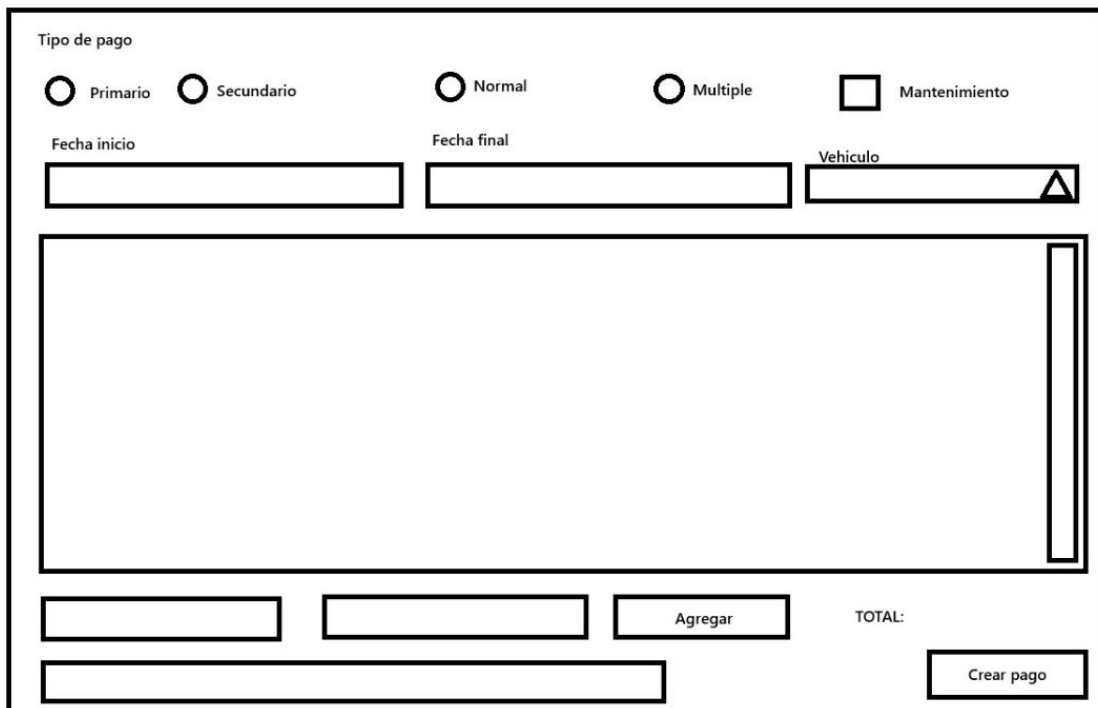
TRABAJO DE GRADO

145

Plantillapagos	Cupos (1: 1) Plantillapagos_detalle (1: N)	Almacena la asociación de un cupo con la configuración de una plantilla de pagos.
Plantillapagos_detalle	Plantillapagos (N: 1) Conceptos (N: 1)	Almacena la configuración o asociación de los conceptos a la plantilla. Es necesaria la tabla para normalizar la relación N: N.
Preplantillas	Flotilla (N: 1) Conceptos (N: 1)	Cada flotilla maneja unos valores asociados a sus pagos. Esta tabla tiene como objetivo definir en un año específico, cuanto corresponde pagar por cada concepto en cada flotilla.
Taquillas	Cuadrecaja (1: N) Pagos (1: N)	Es un maestro con las taquillas autorizadas para recaudo.
Cuadrecaja	Usuarios (N: 1) Taquilla (N: 1)	La tabla permite registrar los valores recaudados por un despachador o recaudador en una jornada específica.
Pagos	Cupo (N: 1) Taquillas (N: 1) Usuarios (N: 1) Estados (N: 1) Solicitud_panulados (1: 1)	Es la entidad que registra cada pago que hace un propietario a través de su vehículo en una fecha específica.
Pagos_detalle	Pagos (N: 1) Conceptos (N: 1)	Es la tabla transaccional que permite normalizar la relación muchos a muchos entre Pagos y Conceptos.
Estados	Pagos (1: N)	Maestro que alberga los estados posibles de un pago.
Solicitud_panulados	Pagos (1: 1) Usuarios (N: 1)	A través de esta tabla se puede controlar y almacenar la historia sobre las solicitudes de anulación que tienen los taquilleros o recaudadores sobre los pagos creados por ellos mismos.

Wireframes

Para este proyecto, no se crearon wireframes para módulos básicos o de maestros. Sin embargo, para el módulo de pagos y de despachos que requerían un poco más de entendimiento visual si se crearon los siguientes Wireframes.



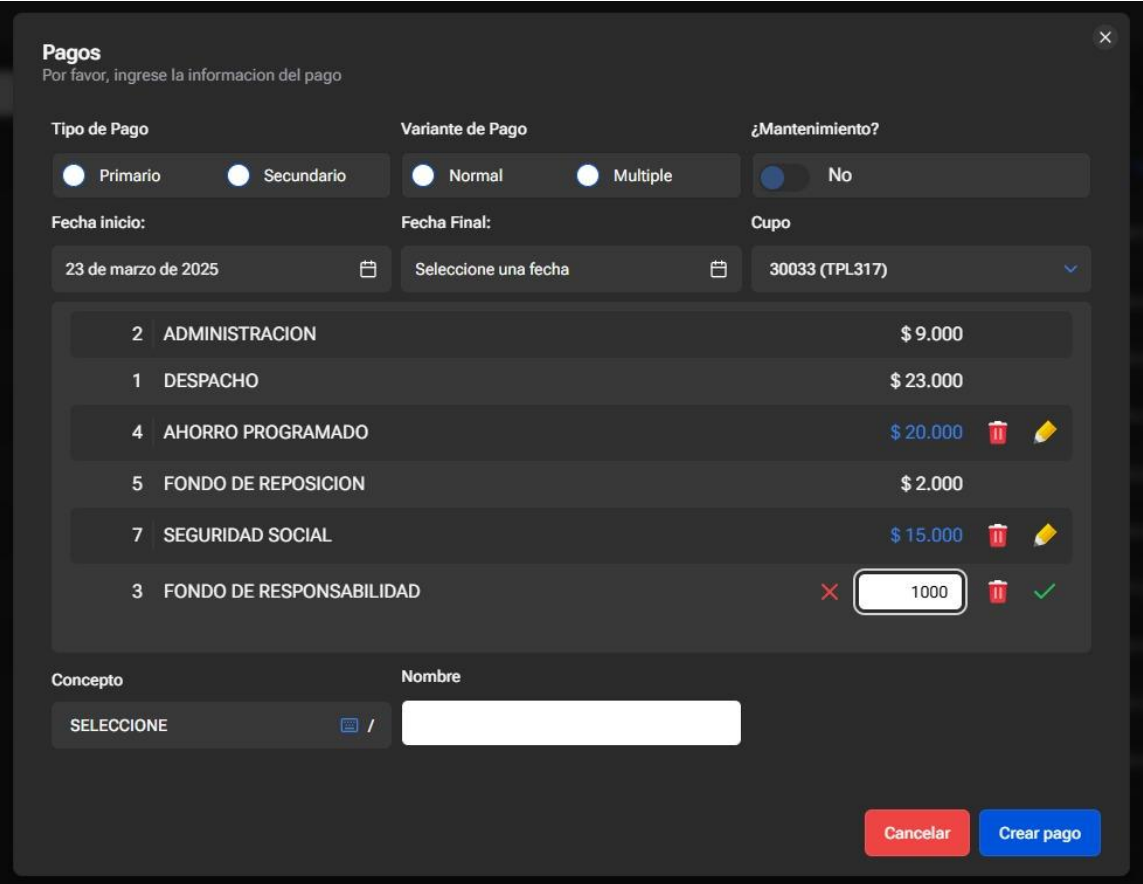
The wireframe shows a payment form layout. At the top, under the heading "Tipo de pago", there are five radio button options: "Primario", "Secundario", "Normal", "Multiple", and "Mantenimiento". Below this, there are three input fields: "Fecha inicio", "Fecha final", and "Vehiculo" (which is a dropdown menu). A large empty rectangular area with a vertical scrollbar on the right side occupies the middle of the form. At the bottom, there are two more input fields, a button labeled "Agregar", a label "TOTAL:", and a button labeled "Crear pago".

Figura 8. Wireframe de formulario de pagos

Este wireframe dio como resultado la siguiente interfaz real:

TRABAJO DE GRADO

147






Pagos ×







Por favor, ingrese la información del pago

Tipo de Pago **Variante de Pago** **¿Mantenimiento?**


Primario Secundario Normal Multiple No

Fecha inicio: **Fecha Final:** **Cupo**

23 de marzo de 2025  Seleccione una fecha  30033 (TPL317) 

2	ADMINISTRACION	\$ 9.000		
1	DESPACHO	\$ 23.000		
4	AHORRO PROGRAMADO	\$ 20.000		
5	FONDO DE REPOSICION	\$ 2.000		
7	SEGURIDAD SOCIAL	\$ 15.000		
3	FONDO DE RESPONSABILIDAD		<input type="text" value="1000"/>	 

Concepto **Nombre**

SELECCIONE  /

Cancelar **Crear pago**

Ilustración 14. Formulario de pagos de SATT

Para despachos, como era un formulario complejo se hizo el siguiente wireframe:

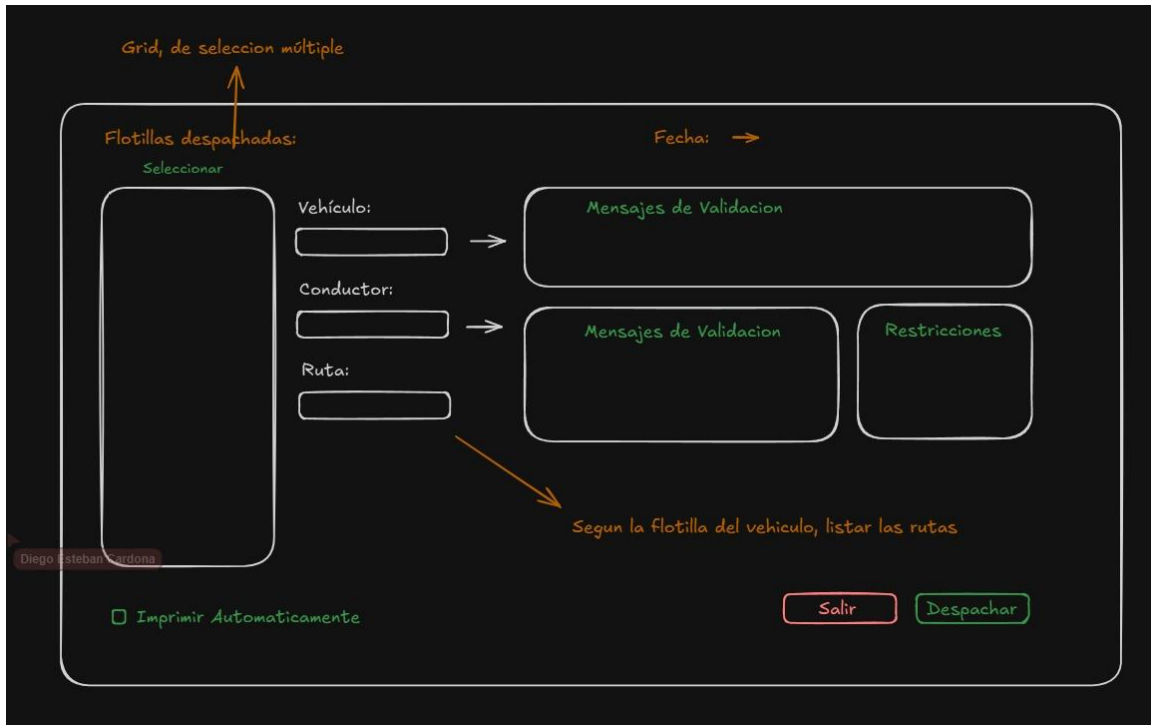


Figura 9. Wireframe del formulario de despachos

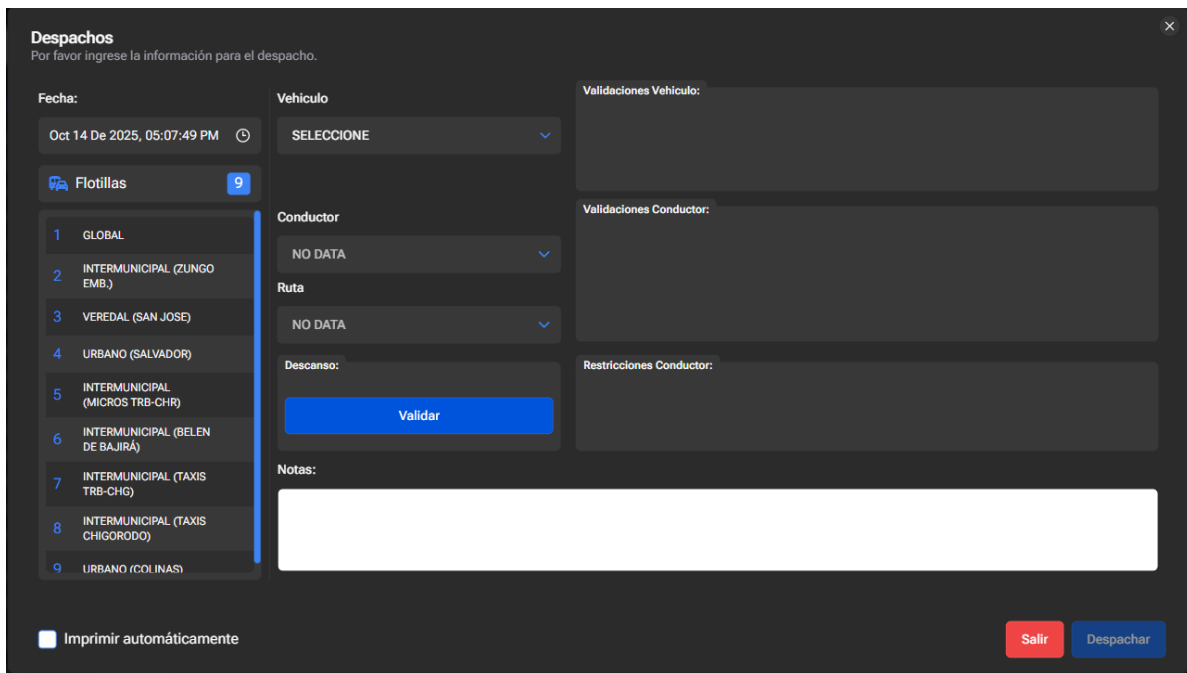


Ilustración 15. Formulario de despachos de SATT

TRABAJO DE GRADO

149

Fase de Desarrollo

Herramientas

Para la fase de codificación se utilizó el siguiente set de herramientas:

Tabla 10. Herramientas para el Desarrollo

<i>Herramienta</i>	<i>Uso</i>
Visual Studio Code	Editor de código.
Git	Herramienta para el control de versiones del código fuente.
GitHub	Repositorio para almacenar el código del proyecto y trabajar con equipo colaborativo.
GitHub Copilot Free	Asistente de IA para auto completado y consultas menores.
ESLint	Extensión para VSCODE que permite escribir código con las mejores prácticas en la industria.
Prettier	Extensión para VSCODE que formatea el código para una mejor indentación y presentación visual que ayuda a la legibilidad del código escrito.
Tailwind CSS IntelliSense	Extensión para VSCODE que permite escribir clases de utilidad de estilos con mayor intuición.

HSL Color Preview	Extensión para VSCODE que permite ver una representación real del color en el código HSL.
MySQL Workbench CE	Herramienta para la administración de motores de base de datos MySQL.
Error Lens	Extensión para VSCODE que muestra los errores en tiempo de edición de código.

Estrategia y Bibliotecas

Como el framework elegido usa React como base de desarrollo, se debía aprovechar al máximo la filosofía de componetización. Esto permitía utilizar fácilmente librerías de terceros y componentes en concreto ya construidos en Front y que podían aprovecharse e incluso extenderse para una mayor adaptación al caso concreto. Muchos de los componentes que se usaron en la UI vendrían de la librería base Shadcn y los iconos de la biblioteca Fontawesome. Algunas de las bibliotecas más importantes son:

Tabla 11. Bibliotecas usadas para el desarrollo

<i>Biblioteca</i>	<i>Uso</i>
Axios	Cliente WEB desde cualquier parte del stack (frontend / backend). Permite realizar las peticiones fetch de forma mas intuitiva.
Bcryptjs	Biblioteca que permite trabajar con criptografía. Útil para el cifrado y comparación de contraseñas en funciones como el login.

TRABAJO DE GRADO

151

Dropbox	Es el cliente de Dropbox para usar su servicio a través de api. Permite tener métodos que interactúan con su servicio en la nube de almacenamiento de archivos.
EJS (Embedded JavaScript Templates)	Es un motor de plantillas para la generación de HTML por parte del servidor. Lo usamos para personalizar los correos electrónicos inyectando variables desde el backend y que son renderizadas al cliente en HTML.
ExcelJS	Es una librería que permite gestionar archivos en formato xlsx (Excel). Se usa para crear exportables en archivos que pueden ser leídos desde Excel.
File-saver	Biblioteca complementaria que permite descargar fácilmente cualquier archivo generado. Utilizado con ExcelJS para permitirle al usuario descargar una vez se genera el archivo xlsx.
jsonwebtoken	Se utiliza para la generación y validación de tokens JWT para controlar el acceso de los usuarios al sistema ya que así se diseñó la estrategia de autenticación en el sistema.
Mime-types	Es una librería que permite reconocer fácilmente el formato de un archivo. Esto ayuda a reconocer los tipos de documentos y validar que se usen los autorizados en el sistema.
Node-cron	Biblioteca super importante para el manejo de tareas automáticas programadas que se ejecutan sobre el hilo de Node. Necesaria para Jobs.
Nodemailer	Esta biblioteca permite enviar correos fácilmente a través del protocolo SMTP desde el backend.
React-hook-form	Es una librería para React que permite manejar el ciclo de vida de un formulario, desde el manejo de errores, la visualización de errores en pantalla, validación y mejoras de rendimiento.
React-to-print	Es una librería que permite escribir código de React y enviarlo al proceso de impresión del navegador. Muy útil para la construcción de imprimibles.

Tailwind CSS	Es un framework de CSS que define clases de utilidad para ayudar a la personalización y rapidez con la que se estilan los componentes de front con CSS.
ZOD	Es una librería extremadamente útil y fácil de manejar que permite crear validaciones por medio de conversiones (parsers), ayudando a validar los formularios y el api del backend de un solo golpe, además se integra muy bien con React-hook-form.
Vitest	Framework de pruebas unitarias del entorno de Vite (Usado por Nextjs). Útil para probar aquellos algoritmos de alta complejidad.

Patrón Aplicado

Para el desarrollo de los módulos con CRUD completos se utilizó un patrón de diseño basado en cliente servidor con API REST. Aunque esto en la practica se encapsula directamente en la capa de next y no hay peticiones cliente que no sean controladas por el framework obedecen al siguiente esquema.

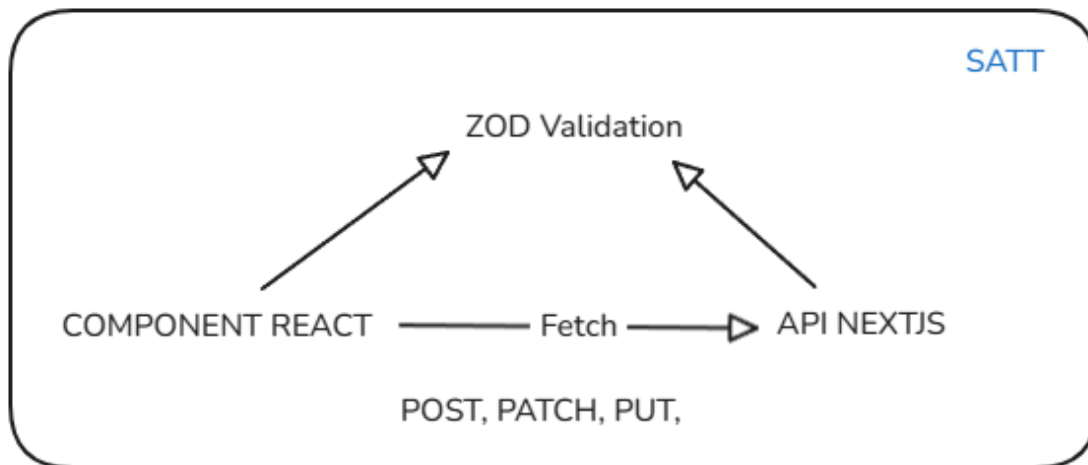


Figura 10. Arquitectura tradicional API REST

TRABAJO DE GRADO

153

Sin embargo, para ciertos componentes de solo lectura de datos que no requerían gestión (actualización, eliminación y actualización al tiempo) se usó la estrategia invocando server actions. Esto difiere del patrón cliente servidor al utilizar un protocolo propio para comunicarse con el backend y obtener los datos.

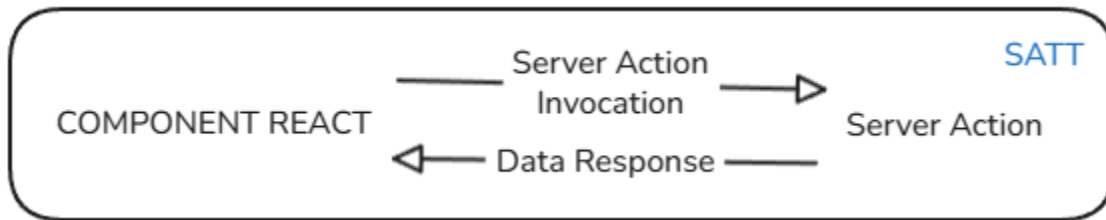


Figura 11. Arquitectura Server Action de Nextjs

La ventaja del patrón tradicional es que los puntos de API o endpoints pueden ser consumidos, desde que se configure bien los CORS, con otras aplicaciones, como por ejemplo una app móvil. Esto puede resultar fácilmente escalable en el tiempo; sin embargo, en contraposición, el modelo con Server Actions simplifica dramáticamente el código dando mas comodidad en la codificación y una integración natural al momento de escribirlo. Veamos un ejemplo con el modulo de flotillas.

El modulo de flotillas esta construido de forma tradicional con puntos de API.

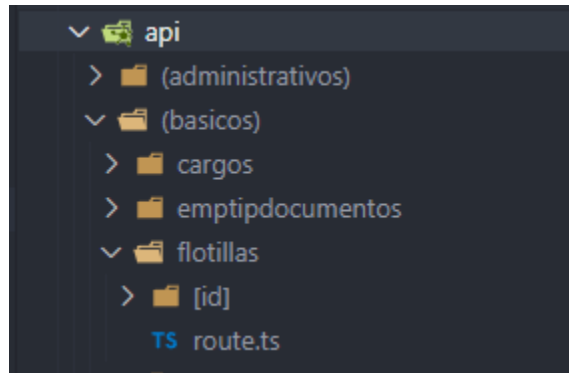


Ilustración 16. Estructura de Árbol para flotillas.

En este caso todo lo que este detrás de la carpeta api será tratado como un endpoint de api y su ruta se irá creando basándose en el orden de carpetas. Esta simplicidad es lo que se conoce como App Router. De modo que tenemos un endpoint `/api/flotillas` y otro enviándole un parámetro o identificador `[id]` `/api/flotillas/[id]` y el método que se desea invocar.

Ahora miremos como se construye un endpoint con sus funciones fundamentales. GET para obtener una lista de flotillas, `[id]/GET` para obtener toda la información de una flotilla con código id, `[id]/PUT` para actualizar todo el registro, POST para crear una nueva flotilla y `[id]/DELETE` para borrar una flotilla.

```
1 export async function GET(req: Request): Promise<NextResponse> {
2   try {
3     const profile = await getCurrentUser();
4     if (!profile) {
5       return NextResponse.json("No se pudo cargar la información de perfil", {
6         status: 400,
7       });
8     }
9     const Modulo = await getAllAccionesByRecurso(["flotillas", "despachos"]);
10    if (!Modulo["flotillas"]?.includes("LIST") && !Modulo["despachos"]?.includes("WRITE")) {
11      return NextResponse.json("No tienes permisos para realizar esta acción", {
12        status: 401,
13      });
14    }
15  }
16
17  const { searchParams } = new URL(req.url);
18
19  const params = {
20    columnFilters: searchParams.get("columnFilters") || "",
21    globalFilter: searchParams.get("globalFilter")?.trim() || "",
22    pageIndex: searchParams.get("pageIndex"),
23    pageSize: searchParams.get("pageSize"),
24    sorting: searchParams.get("sorting") || "",
25    flo_id: searchParams.get("flo_id") || "",
26  };
27
28  const consulta: Prisma.flotillasFindManyArgs = {};
29  const precisionFilters = [];
30  let globalFilters = {};
31
32  if (params.flo_id !== "") precisionFilters.push({ flo_id: Number.parseInt(params.flo_id) });
33  if (params.sorting !== "") {
34    const sorting = JSON.parse(params.sorting);
35    for (const item of sorting) {
36      consulta.orderBy = {
37        [item.id]: item.desc ? "desc" : "asc",
38      };
39    }
40  }
41  if (params.globalFilter !== "") {
42    globalFilters = {
43      OR: [
44        {
45          flo_nombre: {
46            contains: params.globalFilter,
47          },
48        },
49        {
50          flo_descripcion: {
51            contains: params.globalFilter,
52          },
53        },
54        {
55          flo_id: {
56            equals: Number.isNaN(Number(params.globalFilter)) ? 0 : Number(params.globalFilter),
57          },
58        },
59        {
60          flo_diaslabor: {
61            equals: Number.isNaN(Number(params.globalFilter)) ? 0 : Number(params.globalFilter),
62          },
63        },
64      ],
65    };
66  }
67  consulta.select = {
68    flo_id: true,
69    flo_nombre: true,
70    flo_descripcion: true,
71    flo_diaslabor: true,
72  };
73  consulta.where = { AND: [...precisionFilters], ...globalFilters };
74  const totalRecords = await prisma.db.flotillas.count({
75    where: consulta.where,
76  });
77  const registrosOmitir = (Number(params.pageIndex) - 1) * Number(params.pageSize);
78  if (registrosOmitir >= totalRecords || Number(params.pageIndex) <= 0) {
79    return NextResponse.json({ rows: [], totalRecords: 0 });
80  }
81  const registros = await prisma.db.flotillas.findMany({
82    select: consulta.select,
83    where: consulta.where,
84    orderBy: consulta.orderBy,
85    skip: registrosOmitir,
86    take: Number(params.pageSize),
87  });
88  if (registros.length) {
89    const rowWithActions = registros.map((row) => {
90      return { ...row, actions: Modulo.flotillas };
91    });
92    return NextResponse.json({ rows: rowWithActions, totalRecords });
93  } else {
94    return NextResponse.json({ rows: [], totalRecords: 0 });
95  }
96 } catch (error) {
97   console.log("GET_flotillas", error);
98   const errorParse = errorHandler(error);
99   return new NextResponse(errorParse.message, { status: errorParse.status });
100 }
```

Ilustración 17. Metodo GET para obtener todas las flotillas

```
1 export async function POST(req: Request): Promise<NextResponse> {
2   try {
3     const { data } = await req.json().catch(() => {
4       throw "JSONError";
5     });
6
7     if (!data) {
8       return NextResponse.json(
9         "La petición no es correcta. No se ha recibido en el cuerpo de la petición el key: data",
10        {
11          status: 400,
12        },
13      );
14    }
15    await validarRecurso("flotillas");
16    const profile = await getCurrentUser();
17
18    if (!profile) {
19      return NextResponse.json("No se pudo cargar la información de perfil", {
20        status: 400,
21      });
22    }
23    const Modulo = await getAllAccionesByRecurso(["flotillas"]);
24    if (!Modulo["flotillas"].includes("WRITE")) {
25      return NextResponse.json("No tienes permisos para realizar esta acción", {
26        status: 401,
27      });
28    }
29    const registroValidado = await FlotillasSchema.parseAsync(data);
30    const created = await prisma.db.flotillas.create({
31      select: {
32        flo_id: true,
33        flo_nombre: true,
34        flo_descripcion: true,
35        flo_diaslabor: true,
36      },
37      data: { ...registroValidado, flo_sesion: profile.usu_id },
38    });
39    return NextResponse.json({ message: "Registro creado correctamente", data: created }, { status: 201 });
40  } catch (error) {
41    console.log("POST_flotillas", error);
42    const errorParse = errorHandler(error);
43    return new NextResponse(errorParse.message, { status: errorParse.status });
44  }
45 }
```

Ilustración 18. Método POST de Flotillas

Vamos a explicar la ilustración 20 por simplicidad. El método se envuelve en un try catch para controlar los errores y no parar el even loop de node (sería catastrófico).

De la línea 3 a la 5 usamos el cuerpo de la petición que viene en la variable req e intentamos convertir el objeto a json ya que es en este formato que podemos trabajar con

TRABAJO DE GRADO

157

la data enviada por el cliente. Si hay un error en la petición o al menos no es un json lo que nos han enviado, se lanzará una excepción marcada como JSONError.

En la línea 7 se valida que el objeto json tenga una clave llamada “data” de lo contrario el endpoint responde con un mensaje de validación indicando que la petición no es correcta.

En la línea 15 se invoca una función asíncrona “validarRecurso” que valida que el usuario tenga asignado el módulo de flotillas (se quema en el código el nombre del módulo). Como esta es una promesa, si hay un error, se envía al bloque catch y se responde al usuario que no tiene asignado el módulo de flotillas.

En la línea 16 se asigna el objeto de usuario. La función getCurrentUser() retorna un objeto con los datos básicos del usuario que esta con sesión activa y que hace la petición y en la línea 18 se valida que este objeto se haya retornado sino se responde que no se puede cargar información de perfil y por lo tanto no se puede continuar.

En la línea 23 se carga una constante llamada Modulo que contiene un objeto de tipo Array con las acciones cargadas al módulo. Posteriormente se valida que el usuario tenga la acción “WRITE” para escribir o crear flotillas, sino se informa que no tiene el permiso.

En la línea 29 se hace una verificación de parseo con ZOD, cargamos el FlotillaSchema y hacemos un parseAsync para esperar el resultado del parseo. Si hay un problema, la promesa se rechaza y entra al bloque catch enviando un mensaje al usuario sobre que campos tienen problemas. Este mismo esquema es usado en el frontend para validar el formulario antes de enviarse al endpoint.

```
1 function FormFlotillas({ onCloseModal, edit }: FormFlotillasProps) {
2   const { onOpenAlertMessage, onOpenLoader, onCloseLoader } = appConfig.useAppConfig();
3   const form = useForm<z.infer<typeof FlotillasSchema>>({
4     resolver: zodResolver(FlotillasSchema),
5     defaultValues: {
6       flo_nombre: "",
7       flo_descripcion: "",
8       flo_diaslabor: 7,
9     },
10  });
11  const handleOnCloseModal = (data: any) => {
12    form.clearErrors();
13    form.reset();
14    onCloseModal && onCloseModal(data);
15  };
16
17  const onSubmit = async (values: z.infer<typeof FlotillasSchema>) => {
18    try {
19      if (!edit) {
20        onOpenLoader("Creando una flotilla");
21        const res = await axios.post(
22          `/api/flotillas`,
23          { data: values },
24          {
25            headers: {
26              "Content-Type": "application/json",
27            },
28          },
29        );
30        const result = await res.data;
31        handleOnCloseModal(result);
32        onOpenAlertMessage(result.message, "bg-green-600");
33      } else {
34        onOpenLoader("Actualizando tipo de documento");
35        const res = await axios.put(
36          `/api/flotillas/${edit}`,
37          { data: values },
38          {
39            headers: {
40              "Content-Type": "application/json",
41            },
42          },
43        );
44        const result = await res.data;
45        handleOnCloseModal(true);
46        onOpenAlertMessage(result.message, "bg-green-600");
47      }
48    } catch (err: AxiosError | any) {
49      console.log(err);
50      const error: AxiosError = err;
51      onOpenAlertMessage(error.response?.data as string, "bg-red-500");
52    } finally {
53      onCloseLoader();
54    }
55  };
56 }
```

Ilustración 19. Código del formulario de flotillas.

TRABAJO DE GRADO

159

En la línea 4 de la ilustración 21 se puede como se usa en el frontend (formulario) el mismo esquema de Flotillas de zod para validación.

Continuando con la ilustración 20 (endpoint de flotillas) podemos ver que en la línea 30 hasta la 37 se hace el insertado con el adaptador de prisma. Para ello se le indica en el campo data que destructure los valores parseados con zod e incluya los datos del código de usuario en el campo flo_sesion de la persona que hace la transacción (profile.usu_id). Esto crea la flotilla en el sistema y retorna respuesta con código 201 (created) y los datos de la flotilla creada (flo_id, flo_nombre, flo_descripcion, flo_diaslabor).

Las demás líneas corresponden al bloque catch para manejar los errores con errorHandler quien contiene código robusto para manejar los errores, tanto si son de producción o modo de desarrollo; haciendo la traducción específica para evitar retornar al usuario errores con información sensible en producción.

Server Actions

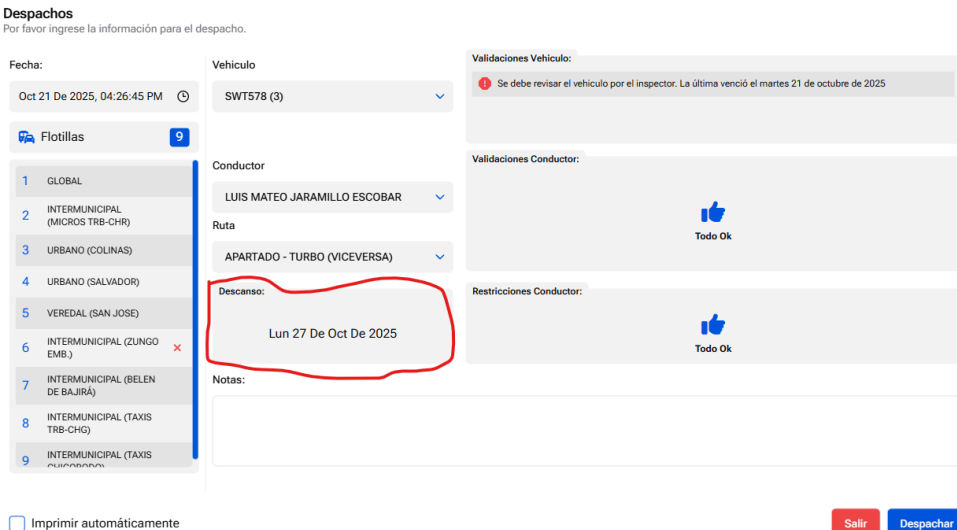
Por su parte los server actions permiten invocar funciones de backend sin necesidad de pasar por el protocolo de fetch tradicional. En la ilustración 21 se puede ver que la función onSubmit en la línea 21 hace una petición post con axios enviando el cuerpo de la petición con los datos del formulario parseados de zod y con unos headers definidos para que el backend sepa tratar la petición.

En el contexto moderno de Nextjs esto se simplifica un poco en la llamada haciendolo mas intuitivo.

```
1 const dataFechaDescanso = await getFechaDescanso(data.des_conductor, data.des_ruta, data.des_fecha.toISOString());
2 if (dataFechaDescanso) {
3   const newFecha = new Date(dataFechaDescanso);
4   setFechaDescanso(new Date(newFecha));
5   const today = new Date();
6   if (
7     newFecha.getFullYear() === today.getFullYear() &&
8     newFecha.getMonth() === today.getMonth() &&
9     newFecha.getDate() === today.getDate()
10  ) {
11    setSolicitarTregua(true);
12  } else {
13    setSolicitarTregua(false);
14  }
15
16  setDescansar(true);
17 }
```

Ilustración 20. Invocando un Server Action

Vemos que en la línea 1 basta con invocar la función pasando los parámetros que se solicitan para obtener la fecha de descanso de un conductor. Basta con el código del conductor, la ruta y la fecha de despacho). La función se invoca al backend sin necesidad de pasar por los protocolos de fetch y manejo de respuestas y simplemente la función retorna la fecha que podemos pintar en frontend.



Despachos
Por favor ingrese la información para el despacho.

Fecha: Oct 21 De 2025, 04:26:45 PM

Flotillas: 9

- 1 GLOBAL
- 2 INTERMUNICIPAL (MICROS TRB-CHR)
- 3 URBANO (COLINAS)
- 4 URBANO (SALVADOR)
- 5 VEREDAL (SAN JOSE)
- 6 INTERMUNICIPAL (ZUNGO EMB.)
- 7 INTERMUNICIPAL (BELEN DE BAJIRÁ)
- 8 INTERMUNICIPAL (TAXIS TRB-CHG)
- 9 INTERMUNICIPAL (TAXIS TRB-CHG)

Vehículo: SWT578 (3)

Conductor: LUIS MATEO JARAMILLO ESCOBAR

Ruta: APARTADO - TURBO (VICEVERSA)

Descanso: **Lun 27 De Oct De 2025**

Notas:

Validaciones Vehículo: Se debe revisar el vehículo por el inspector. La última venció el martes 21 de octubre de 2025

Validaciones Conductor: Todo Ok

Restricciones Conductor: Todo Ok

Imprimir automáticamente

Salir Despachar

Ilustración 21. Formulario de Despacho

TRABAJO DE GRADO

161

No nos cansaremos de decir que esto simplifica enormemente las llamadas asíncronas a backend pero con la desventaja de no poderse invocar directamente de una aplicación externa, sino que solamente del stack de nextjs. Esta función es altamente compleja y contiene acceso a datos a través de prisma, pero con invocar la función desde el frontend se obtiene los datos a una velocidad increíble.

Para resumir, vemos que escribir código en nextjs obedece mas a un paradigma funcional que a uno de orientación a objetos. Si bien el core de nextjs trabaja con estructuras de clase definidas en su interior, permite abstraer ese patrón con la creación de hooks que simulan el comportamiento de clases. Por ejemplo, el useEffect es un hook propio de React que abstrae el constructor de una clase. Un componente cuando es instanciado llama a su constructor para inicializar su estado. Sin embargo, gracias al useEffect es cuestión de escribir las rutinas que serán ejecutadas al momento de inicializar el componente (constructor) y esto en cierta forma genera cierta comodidad para muchos programadores.

```
1 "use server";
2 import errorHandler from "@lib/errorManager";
3 import prismaDb from "@lib/prismaDb";
4 import { truncateHour } from "@lib/utills";
5
6 export default async function getFechaDescanso(
7   conductor: number,
8   rutaId?: number,
9   fechaString: string = new Date().toISOString(),
10 ): Promise<string | null> {
11   try {
12     // Obtener configuración de descanso
13     const fecha = truncateHour(new Date(fechaString));
14     const config = await prismaDb.keytags.findFirst({
15       where: { tag: "ads", keychar: "SI" },
16       select: { keynum: true },
17     });
18
19     if (!config) return null;
20     let diasContinuos = 0;
21     let diasTrabajados = 0;
22
23     if (rutaId) {
24       const diasLaborablesrut = await prismaDb.rutas.findUnique({
25         where: { rut_id: rutaId },
26         select: { flotillas: { select: { flo_diaslabor: true } } },
27       });
28       diasContinuos = Number(diasLaborablesrut?.flotillas?.flo_diaslabor);
29     } else {
30       const ultimaRuta = await prismaDb.despachos.findFirst({
31         where: { des_conductor: conductor, des_febaja: null },
32         select: { rutas: { select: { flotillas: { select: { flo_diaslabor: true } } } } },
33         orderBy: { des_fecha: "desc" },
34       });
35       diasContinuos = Number(ultimaRuta?.rutas?.flotillas?.flo_diaslabor);
36     }
37
38     if (!diasContinuos) return null;
39
40     // Obtener despachos dentro del margen de días laborables hacia atrás
41     const fechaInicio = new Date(fecha);
42     fechaInicio.setDate(fechaInicio.getDate() - diasContinuos);
43
44     const registros = await prismaDb.despachos.findMany({
45       where: {
46         des_fecha: { gte: fechaInicio, lt: fecha },
47         des_conductor: conductor,
48         des_febaja: null,
49       },
50       select: { des_fecha: true },
51       orderBy: { des_fecha: "desc" },
52     });
53
54     // Determinar el último día no trabajado
55     const fechasTrabajadas = new Set(registros.map((d) => truncateHour(d.des_fecha).toISOString()));
56     const fechaPuntero = new Date(fecha);
57     fechaPuntero.setDate(fechaPuntero.getDate() - 1);
58
59     while (fechasTrabajadas.has(fechaPuntero.toISOString()) && fechaPuntero.getDate() >= fechaInicio.getDate()) {
60       fechaPuntero.setDate(fechaPuntero.getDate() - 1);
61       diasTrabajados += 1;
62     }
63
64     // Si no se encontró un día de descanso en el rango, hoy debe ser su descanso
65     if (diasContinuos >= diasTrabajados) {
66       // Verificar si la fecha actual está autorizada
67       const fechaAutorizada = await prismaDb.solicitudjornadas.findFirst({
68         where: {
69           solj_empleado: conductor,
70           solj_estado: "A",
71           AND: [
72             { solj_fechaInicio: { gte: fecha }, // Fecha de inicio menor o igual a la búsqueda
73             { solj_fechaFinal: { gte: fecha }, // Fecha de fin mayor o igual a la búsqueda
74           ],
75         },
76         orderBy: { solj_fechaRespuesta: "desc" },
77       });
78
79       if (fechaAutorizada) {
80         const nuevoDescanso = new Date(fechaAutorizada.solj_fechaFinal);
81         nuevoDescanso.setDate(nuevoDescanso.getDate() + 1);
82         return normalizarFechaDescanso(nuevoDescanso);
83       }
84       return normalizarFechaDescanso(fecha);
85     }
86
87     // Calcular la fecha de descanso sumando los días laborables desde el último día no trabajado
88     if (fechaPuntero >= fechaInicio && fechaPuntero < fecha) {
89       fechaPuntero.setDate(fechaPuntero.getDate() + 1);
90     }
91
92     const fechaPosible = new Date(fechaPuntero);
93     fechaPosible.setDate(fechaPosible.getDate() + diasContinuos);
94
95     return normalizarFechaDescanso(fechaPosible);
96   } catch (error) {
97     console.log("GET_FECHA_DESCANSO_CONDUCTOR", error);
98     const response = errorHandler(error, "GET_FECHA_DESCANSO_CONDUCTOR");
99     console.log(response);
100    return null;
101  }
102 }
```

Ilustración 22. Contenido de una Server Action

TRABAJO DE GRADO

163

Estas funciones se caracterizan por tener la directiva “use server” y se invocan desde cualquier parte de nextjs excepto procesos críticos de node.

En resumen, se utilizó los endpoint de api para la mayoría de modulos que tenían gestión completa (CRUD). Por su parte, para funciones muy específicas como autorizaciones u obtención de datos muy puntuales se usaron las server actions.

Estructura Principal de Carpetas

Se usó una estructura que no sigue fielmente un estándar, pero se acomoda perfectamente a un desarrollo ordenado con este framework.

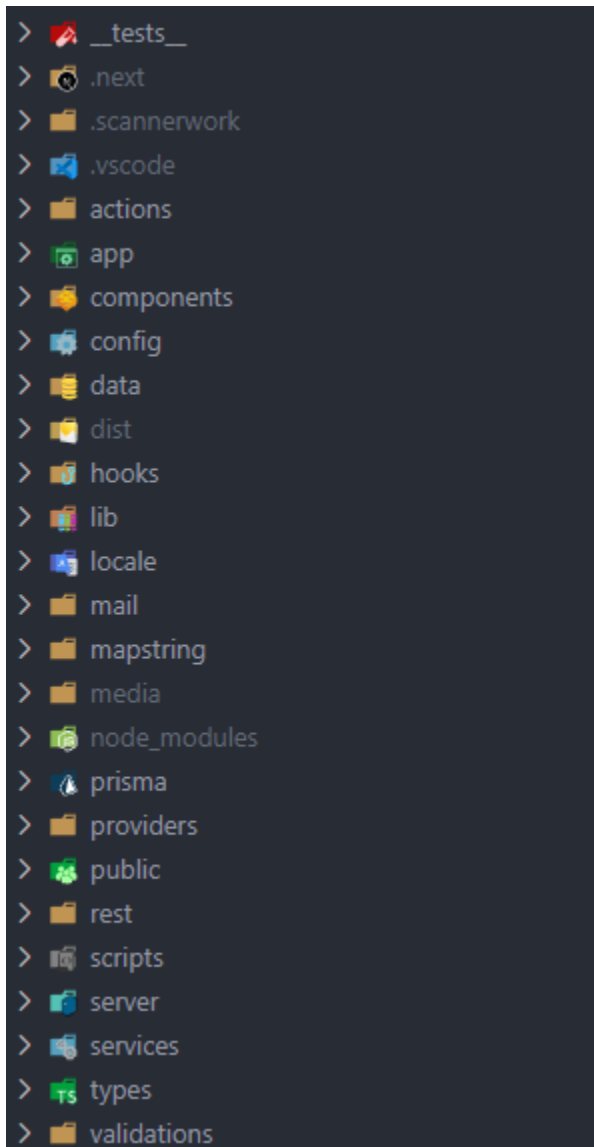


Ilustración 23. Esquema de Carpetas de SATT

TRABAJO DE GRADO

165

Tabla 12. Estructura de Carpetas de SATT.

<i>Carpeta</i>	<i>Uso</i>
__test__	Permite crear test unitarios que se correrán con el framework vitest.
.next	Contiene los compilados de next para trabajar en modo de desarrollo.
.scannerwork	Es utilizado para el escáner de sonar Qube en el modo manual y enviar el código al repositorio para su posterior análisis y poder sacar las métricas de código.
.vscode	Permite configurar el editor a nivel de proyecto.
Actions	Carpeta que contiene las Server Actions del sistema.
App	Carpeta principal del framework de nextjs. Aquí se utiliza los puntos de entrada de la aplicación y su uso es fundamental para la creación de todo el sistema.
Components	Contiene componentes personalizados de React que serán usados en toda la aplicación.
Config	Contiene información básica de la aplicación como el nombre y la versión.
Data	Contiene archivos de acceso a datos para la autenticación. Necesario para el ecosistema Auth.js y mantenerlas separadas de otros accesos a datos con lógica de negocio.

TRABAJO DE GRADO

166

Dist	Carpeta usada por Next para guardar los archivos ya compilados de la aplicación (con la ejecución del comando next build).
Hooks	Aca se guardan ciertas funciones personalizadas que requieren control de estados, efectos o comportamientos específicos de la aplicación.
Lib	Contiene funciones especializadas que están mas que todo del lado del backend. Estas pueden incluir funciones de API avanzadas y otras que se invocan por un endpoint de API.
Locale	Se utiliza para incluir archivos de traducción de la aplicación. En caso de internacionalizar el sistema, aca estarán todos los archivos de idioma necesarios para hacer la traducción o cambiar el idioma.
Mail	Carpeta que contiene plantillas y vistas necesarias para la construcción de emails.
Mapstring	Carpeta que contiene diccionario de datos. Útil para crear estructuras de diccionarios que requieren múltiples ítems y necesitan ser traducidos a un lenguaje mas natural y humano. Como es el caso de los estados de documentos.
Media	Esta carpeta contiene los documentos cargados a empleados y vehículos. También cualquier recurso multimedia que le pertenece al cliente como fotos de usuario y anexos para historiales.
Node_modules	Contiene todas las librerías instaladas en el proyecto. Node controla su contenido.
Prisma	Contiene el modelo y vistas creadas por prisma ORM para el trabajo con entidades de datos.
Providers	Contiene los proveedores que envuelven a la aplicación como por ejemplo manejadores de sesión y manejadores de estilos de tema.

TRABAJO DE GRADO

167

Public	Contiene recursos de acceso público, como iconos corporativos, imágenes de inicio de sesión y otros que hacen parte de la identidad de la aplicación.
Rest	Contiene archivos que entiende la extensión Res Client y que permite probar puntos de api de forma casi documentada. Útil cuando hay separación de responsabilidades y se desarrolla el backend sin estar necesariamente listo el frontend.
Scripts	Contiene archivos javascript personalizados, necesarios para la configuración inicial de ciertos servicios como por ejemplo Dropbox.
Server	Contiene archivos personalizados de Typescript para el levantamiento del servidor y otros servicios anexos como Sockets por ejemplo.
Types	Contiene declaración de tipos globales para ciertas librerías. Útil para el autocompletado y la validación en tiempo de desarrollo de typescript.
Validations	Contiene todos los archivos de esquemas de la librería ZOD necesarios para validación de datos de formularios.

Estrategia de Pruebas

El proyecto no fue codificado con un esquema riguroso de pruebas de caja blanca. Cuando se tenía una construcción del software estimado en más del 70% se implementó un servidor QA para tener interacción con ciertos usuarios quien dieron sus opiniones y mostraron satisfacción en su uso.

Algunos algoritmos de alta complejidad si fueron testeados con vitest, un ejemplo es el algoritmo que controla los descansos de los conductores. Estos se incluyeron en casos de prueba unitarias arrojando éxito en 10 casos de prueba funcionales y 1 de archivo.

```
1 // Mock de prismadb
2 vi.mock("../lib/prismadb", () => ({
3   default: {
4     keytags: {
5       findFirst: vi.fn(),
6     },
7     rutas: {
8       findUnique: vi.fn(),
9     },
10    despachos: {
11      findFirst: vi.fn(),
12      findMany: vi.fn(),
13    },
14    solicitudjornadas: {
15      findFirst: vi.fn(),
16    },
17  },
18 }));
19
20 // Mock de errorManager
21 vi.mock("../lib/errorManager", () => ({
22   default: vi.fn(),
23 }));
24
25 // Mock de utils
26 vi.mock("../lib/utils", () => ({
27   truncateHour: vi.fn((date: Date) => {
28     const d = new Date(date);
29     d.setHours(0, 0, 0, 0);
30     return d;
31   }),
32 }));
33
34 import prismadb from "../lib/prismadb";
35 import errorManager from "../lib/errorManager";
36
37 describe("getFechaDescanso", () => {
38   beforeEach(() => {
39     vi.clearAllMocks();
40   });
41
42   afterEach(() => {
43     vi.restoreAllMocks();
44   });
45
46   it("debe retornar null cuando no hay configuración de descanso", async () => {
47     // Mock sin configuración
48     vi.mocked(prismadb.keytags.findFirst).mockResolvedValue(null);
49
50     const result = await getFechaDescanso(1);
51
52     expect(result).toBeNull();
53     expect(prismadb.keytags.findFirst).toHaveBeenCalledWith({
54       where: { tag: "ads", keychar: "SI" },
55       select: { keynum: true },
56     });
57   });
58
59   it("debe retornar null cuando no hay días continuos configurados", async () => {
60     // Mock configuración con keynum = 0
61     vi.mocked(prismadb.keytags.findFirst).mockResolvedValue({ keynum: new Decimal(0) } as any);
62     vi.mocked(prismadb.despachos.findFirst).mockResolvedValue(null);
63
64     const result = await getFechaDescanso(1);
65
66     expect(result).toBeNull();
```

Ilustración 24. Muestra de código de prueba

```
DEV v3.1.1 C:/Proyectos/SATT3
stdout | __tests__/getdescansoConductor.test.ts > getFechaDescanso > debe manejar errores y retornar null
GET_FECHA_DESCANSO_CONDUCTOR Error: DB Error
  at C:\Proyectos\SATT3\__tests__\getdescansoConductor.test.ts:223:59
  at file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:103:11
  at file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:595:26
  at file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:877:20
  at new Promise (<anonymous>)
  at runWithTimeout (file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:850:10)
  at runTest (file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:1345:12)
  at processTicksAndRejections (node:internal/process/task_queues:105:5)
  at runSuite (file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:1491:8)
  at runSuite (file:///C:/Proyectos/SATT3/node_modules/@vitest/runner/dist/index.js:1491:8)
  { message: 'Error', status: 500 }

✓ __tests__/getdescansoConductor.test.ts (10 tests) 86ms
✓ getFechaDescanso > debe retornar null cuando no hay configuración de descanso 4ms
✓ getFechaDescanso > debe retornar null cuando no hay días continuos configurados 0ms
✓ getFechaDescanso > debe retornar null para conductor sin despachos previos 0ms
✓ getFechaDescanso > debe calcular descanso cuando alcanza límite de días consecutivos 70ms
✓ getFechaDescanso > debe calcular la fecha de descanso desde el ultimo día que descansó 1ms
✓ getFechaDescanso > no debe bloquear cuando días no son consecutivos 1ms
✓ getFechaDescanso > debe ajustar fecha cuando hay solicitud de jornada autorizada de un solo día 0ms
✓ getFechaDescanso > debe ajustar fecha cuando hay solicitud de jornada autorizada de mas de un día 0ms
✓ getFechaDescanso > debe mover descanso del domingo al lunes 1ms
✓ getFechaDescanso > debe manejar errores y retornar null 7ms

Test Files 1 passed (1)
Tests 10 passed (10)
Start at 09:26:27
Duration 21.72s (transform 234ms, setup 2.06s, collect 216ms, tests 86ms, environment 14.62s, prepare 1.08s)

PASS Waiting for file changes...
press h to show help, press q to quit
```

Ilustración 25. Resultado de test

Sin embargo, la mayor parte del tiempo se hicieron pruebas de caja negra, probando el software por el equipo de desarrollo.

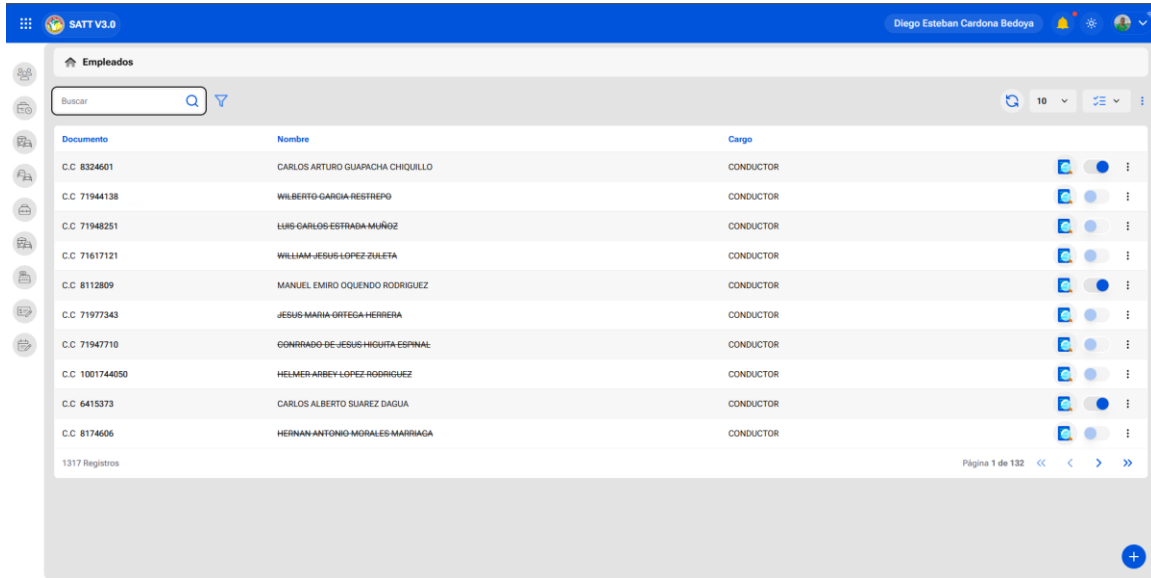
Al momento del despliegue se tuvo una aceptación del 99% gracias a los acercamientos al programa desde la instancia de QA. El otro 1% fueron abordados con estrategias de reinducción y capacitación reforzadas que ayudaron a entender la funcionalidad del sistema.

Algunas Capturas de Pantalla

El resultado fue una aplicación web que cumple con los requisitos funcionales del negocio. Se ejecuta la mayoría de módulos sobre el dispositivo móvil haciendo más simple la portabilidad y el acceso.

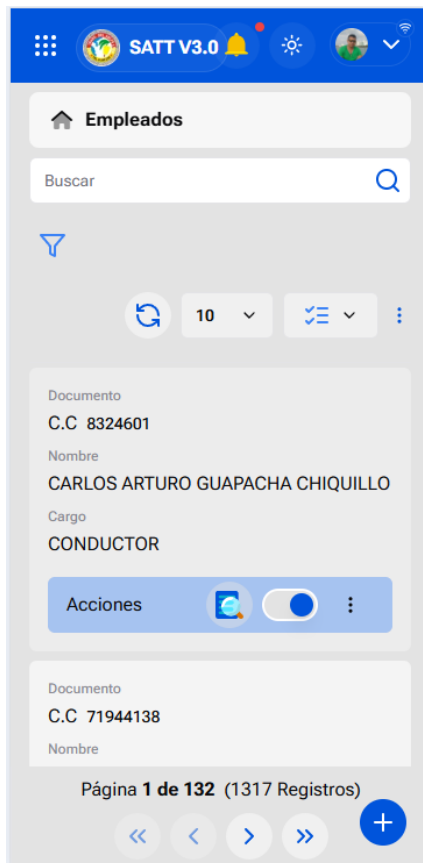
TRABAJO DE GRADO

170



Documento	Nombre	Cargo
C.C. 8324601	CARLOS ARTURO GUAPACHA CHIQUILLO	CONDUCTOR
C.C. 71944138	WILBERTO GARCIA RESTREPO	CONDUCTOR
C.C. 71948251	LUIS CARLOS ESTRADA MUÑOZ	CONDUCTOR
C.C. 71617121	WILLIAM JESUS LOPEZ ZULETA	CONDUCTOR
C.C. 8112809	MANUEL EMIRO OQUENDO RODRIGUEZ	CONDUCTOR
C.C. 71977343	JESUS MARIA ORTEGA HERRERA	CONDUCTOR
C.C. 71947710	IGNACIO DE JESUS HIGUETA ESPINAL	CONDUCTOR
C.C. 1001744050	HELMER ARBEY LOPEZ RODRIGUEZ	CONDUCTOR
C.C. 6415373	CARLOS ALBERTO SUAREZ DAGUA	CONDUCTOR
C.C. 8174606	HERNAN ANTONIO MORALES MARRIAGA	CONDUCTOR

Ilustración 26. Módulo de Empleados vista escritorio



Documento C.C. 8324601	Nombre CARLOS ARTURO GUAPACHA CHIQUILLO	Cargo CONDUCTOR
Documento C.C. 71944138	Nombre	

Ilustración 27. Vista móvil del módulo de empleados

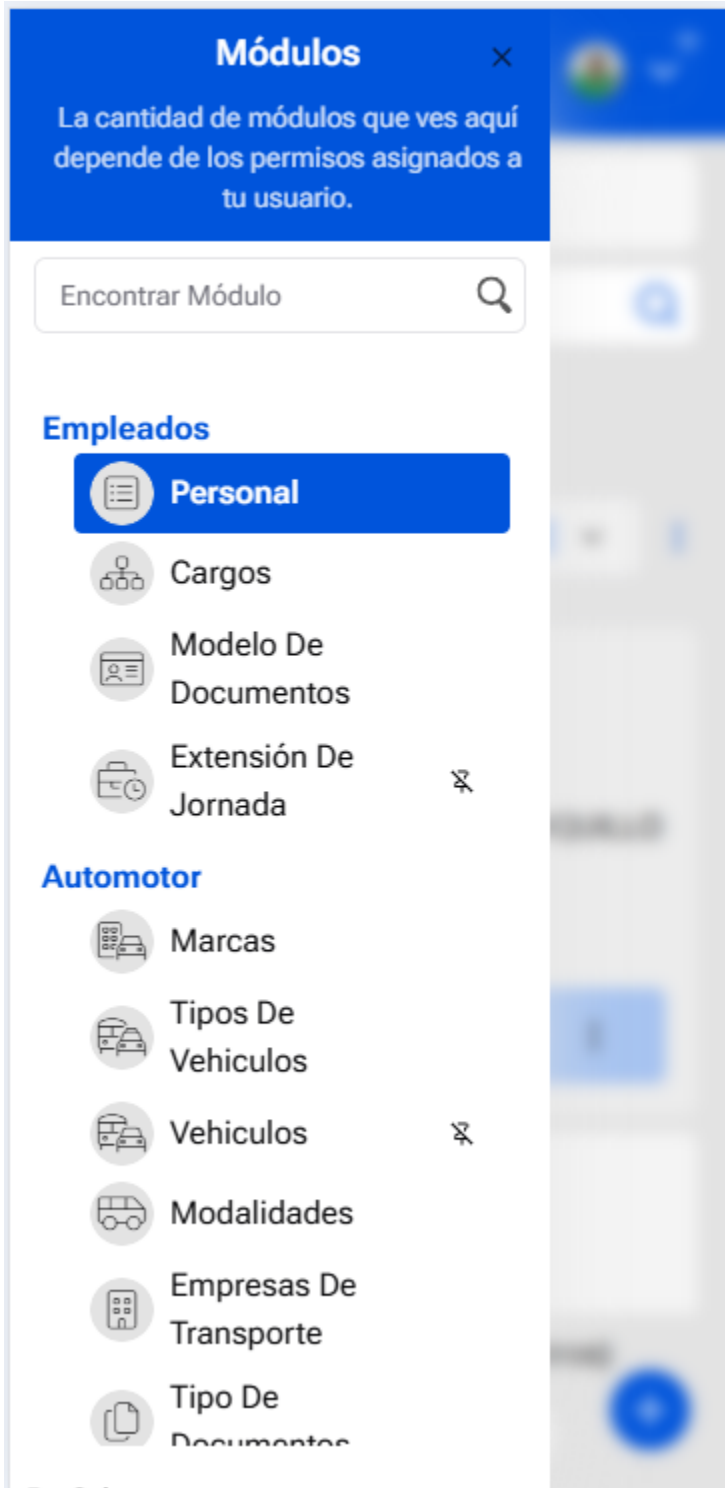


Ilustración 28. Menú de Acceso versión móvil.

TRABAJO DE GRADO

172

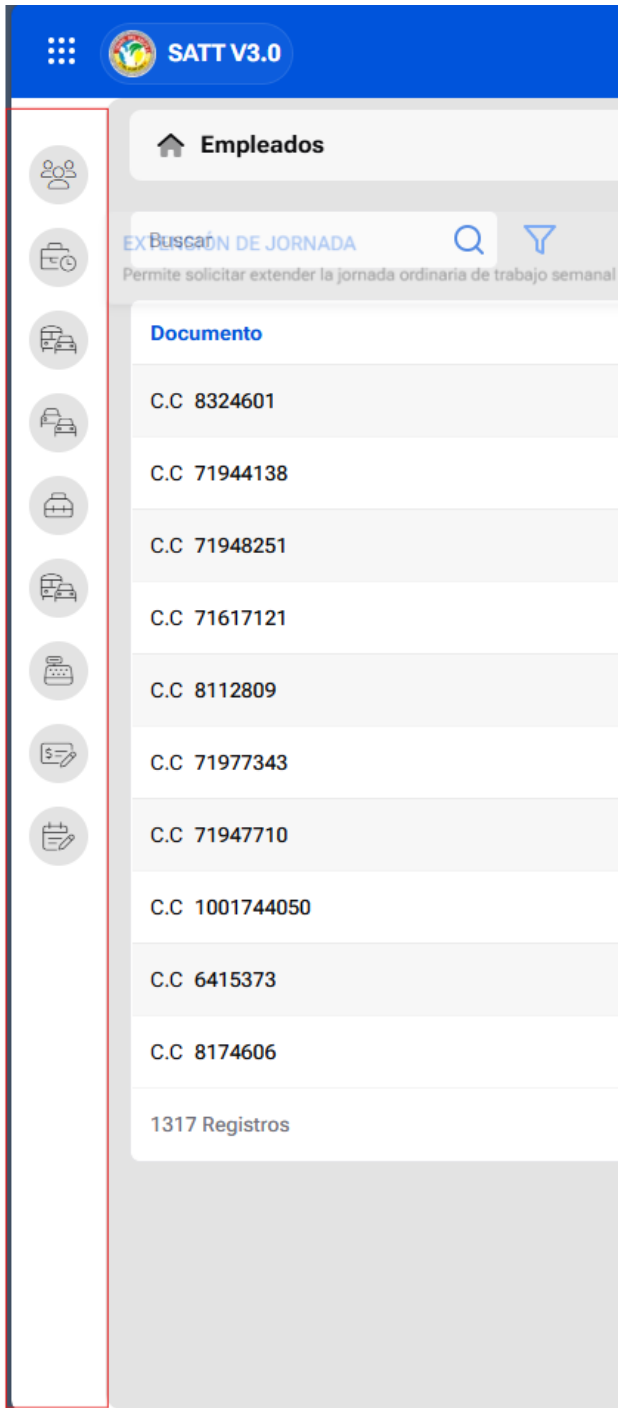



Ilustración 29. Barra de acceso rápido (solo en versión escritorio)

TRABAJO DE GRADO

173

Agregar colaborador ✕

Por favor, diligenciar los campos para agregar un nuevo colaborador



OG

Imagen

Tipo de documento	Documento	Cargo	Tipo de Licencia
SELECCIONE ▾	<input type="text"/>	SELECCIONE ▾	SELECCIONA EL TIPO DE LICE ▾
Nombres		Apellidos	
<input type="text"/>		<input type="text"/>	
Genero	Fecha de nacimiento	Fuma	Tipo de sangre [RH]
<input type="radio"/> Masculino <input type="radio"/> Femenino	Seleccione Una Fecha <input type="text"/>	<input checked="" type="radio"/> No	SELECCIONE ▾
Ciudad de Nacimiento	Ciudad de Residencia	Dirección	Teléfono
SELECCIONE ▾	SELECCIONE ▾	<input type="text"/>	<input type="text"/>
Teléfono Alternativo	Barrio	Deporte	Correo electrónico
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>


Cancelar

Guardar

Ilustración 30. Formulario de empleados (versión de escritorio).

Agregar colaborador ✕

Por favor, diligenciar los campos para agregar un nuevo colaborador



Imagen

Tipo de documento

SELECCIONE ▾

Documento

Cargo

SELECCIONE ▾

Tipo de Licencia

SELECCIONA EL TIPO DE LICENCIA ▾

Nombres

Cancelar

Guardar

Ilustración 31. Formulario de empleados (versión móvil).

TRABAJO DE GRADO

174

Métricas de código

Se ha usado el servicio de sonar Qube cloud para obtener métricas de código del programa.

















	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
 satt							
 __tests__	261	0	0	5	0	0.0%	6.0%
 actions	5,825	0	2	119	1	0.0%	4.0%
 app	68,865	0	19	1341	3	0.0%	22.4%
 components	13,402	0	31	417	3	0.0%	20.1%
 config	9	0	0	0	0	0.0%	0.0%
 data	255	0	0	12	0	0.0%	0.0%
 hooks	433	0	0	13	0	0.0%	0.0%
 lib	2,931	0	5	97	1	0.0%	12.6%
 locale	395	0	1	3	0	0.0%	0.0%
 mail	766	0	7	9	0	—	0.0%
 mapstring	59	0	0	0	0	0.0%	0.0%
 providers	272	0	0	13	0	0.0%	0.0%
 public	68	0	0	0	0	—	0.0%
 scripts	134	0	0	7	1	0.0%	0.0%
 server	108	0	0	8	0	0.0%	0.0%

Ilustración 32. Métricas por carpetas

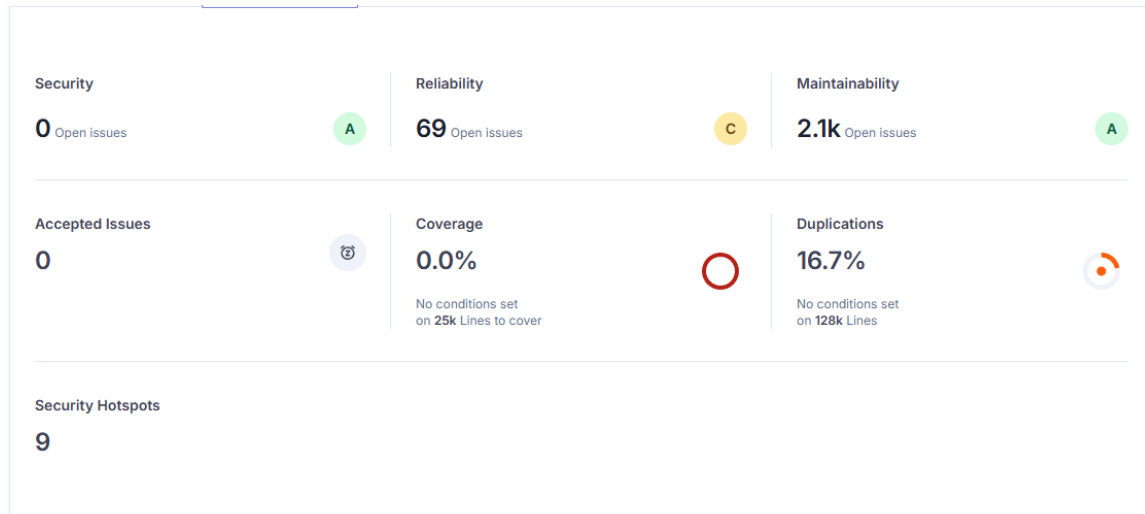


Ilustración 33. Resumen de métricas de sonar

Se obtiene una clasificación en seguridad A (la mas alta), fiabilidad C, mantenibilidad A, cobertura de test 0% y duplicaciones de 16.7% del código.

Cabe destacar que Sonar hace un escaneo estático y no en tiempo de ejecución. Muchas de los issues de fiabilidad corresponden a problemas de accesibilidad, pero por la complejidad de los componentes heredados de otros desarrolladores o la misma funcionalidad programada exige este tipo de comportamientos. También sugiere utilizar funciones de reemplazado u otros métodos más modernos en el código que por el contexto no pueden ser aplicados (falsos positivos). Como por ejemplo el reemplazo de string a traves de expresiones regulares. Veamos un ejemplo:

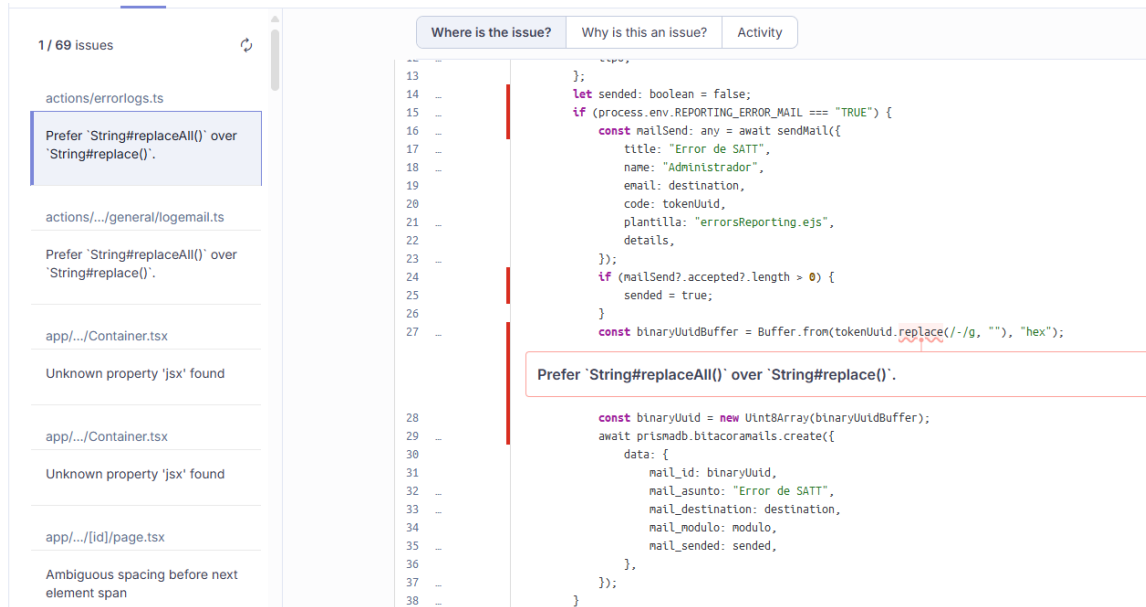


Ilustración 34. Revisión de una Issue de fiabilidad

Vemos que sonar nos sugiere cambiar el replace con expresión regular por String.ReplaceAll(), sin embargo, esto es incorrecto porque el token no es una clase string directamente sino un objeto entonces colocar ReplaceAll daría un problema al momento de compilar. Veamos



Ilustración 35. Corrección de Issue de Sonar

Después de hacer la “corrección” y ejecutar el compilador vemos la siguiente salida:

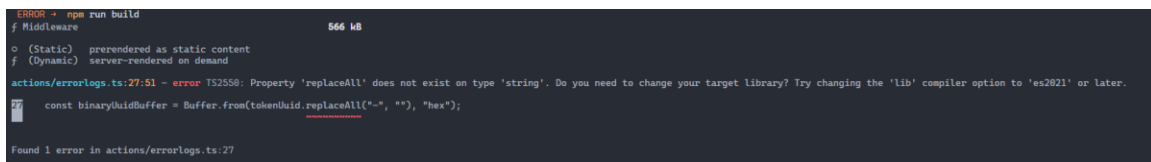


Ilustración 36. Resultado del proceso de compilación

Claramente se puede observar que el método no es soportado por el stack tecnológico usado en el proyecto.

Otro de los problemas de fiabilidad reportados por sonar es el de sintaxis no reconocida pero que es totalmente valida en nuestro stack tecnológico. Por ejemplo, la sintaxis de estilado en línea no es reconocida y marcada como un problema.

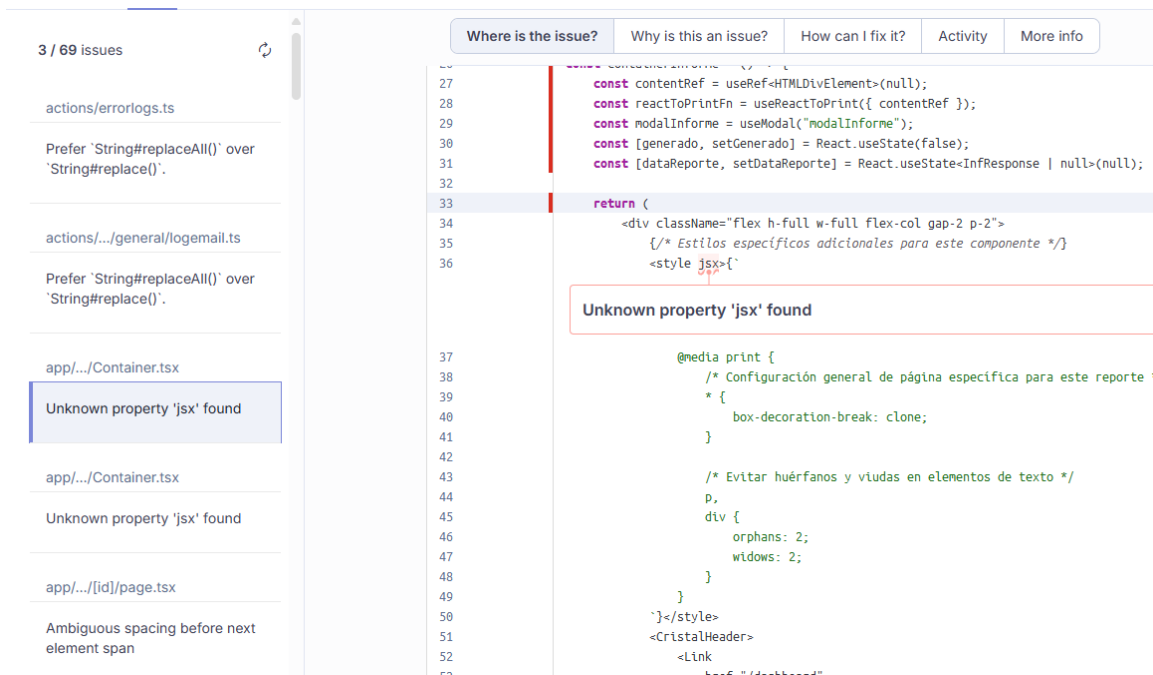


Ilustración 37. Issue jsx reportada por sonar

TRABAJO DE GRADO

178

Sin embargo, claramente esa sintaxis es totalmente válida para estilos en línea de componentes jsx del framework de next.

En ese contexto, entendiendo que muchos de los problemas de fiabilidad reportados por sonar son en realidad falsos positivos, concluimos que nuestro código es altamente fiable, seguro, mantenible y con buenas prácticas de codificación.

Metodología de Implementación

La empresa patrocinadora cuenta con infraestructura física, un servidor HP Proliant G9 con 32 GB de RAM, procesador Intel XEON, 2TB de almacenamiento en disco y 6 interfaces de red. Para esto se decide instalar un hypervisor que permita tener múltiples máquinas virtuales dentro del servidor.

Con esta infraestructura disponible se ha diseñado el siguiente modelo de despliegue:

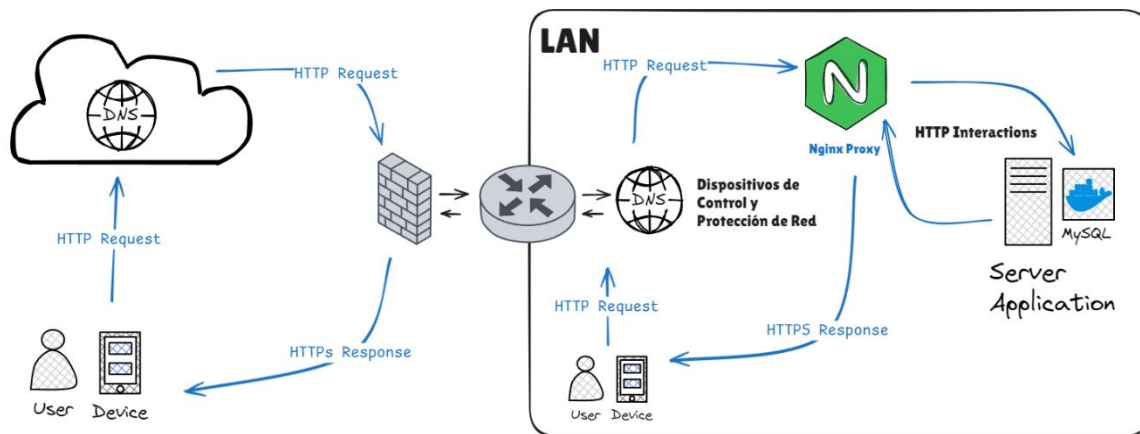
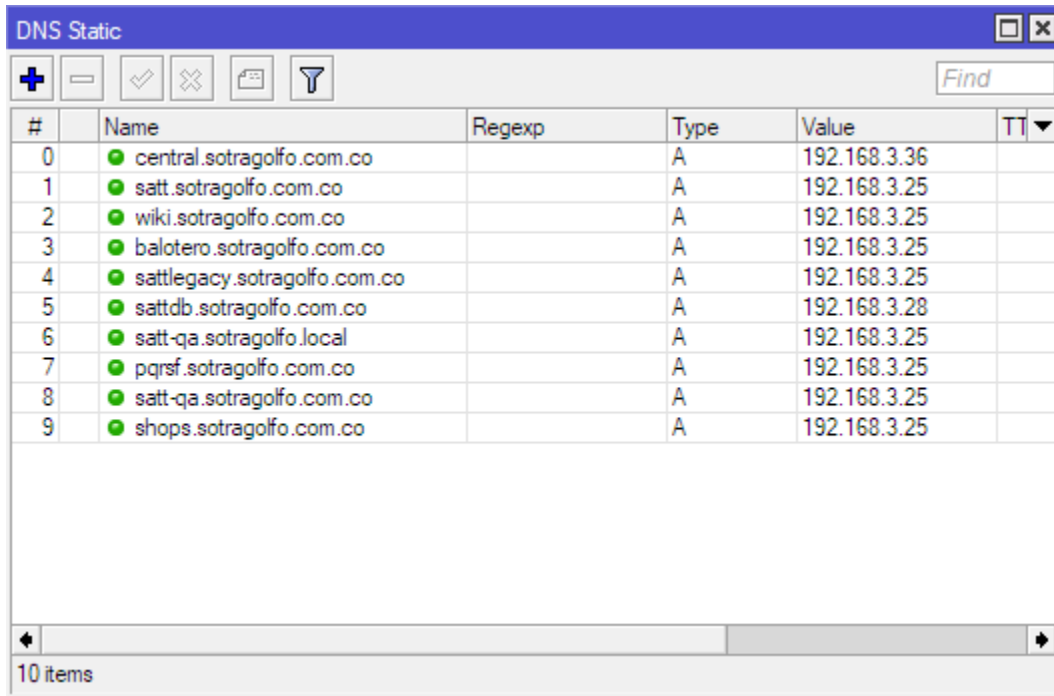


Figura 12. Arquitectura de Despliegue

Con este esquema se pretende crear una intranet, configurando un router mikrotik para que maneje un DNS local. Esto permite redirigir las solicitudes http sin necesidad de salir a la WAN.

TRABAJO DE GRADO

180



#	Name	Regex	Type	Value	TT
0	central.sotragolfo.com.co		A	192.168.3.36	
1	satt.sotragolfo.com.co		A	192.168.3.25	
2	wiki.sotragolfo.com.co		A	192.168.3.25	
3	balotero.sotragolfo.com.co		A	192.168.3.25	
4	sattlegacy.sotragolfo.com.co		A	192.168.3.25	
5	sattdb.sotragolfo.com.co		A	192.168.3.28	
6	satt-qa.sotragolfo.local		A	192.168.3.25	
7	pqrfs.sotragolfo.com.co		A	192.168.3.25	
8	satt-qa.sotragolfo.com.co		A	192.168.3.25	
9	shops.sotragolfo.com.co		A	192.168.3.25	

Ilustración 38. Entradas DNS estático en routerboard.

De esta manera se aprovecha la implementación on premise, que permite acceder al software incluso en escenarios donde la conectividad WAN puede fallar. Además, los tiempos de espera y latencia son casi nulos al estar conectados en una misma infraestructura LAN.

Sin embargo, el routerboard ha sido configurado para recibir peticiones por el puerto 80, redirigiendo el tráfico a un único servicio de Nginx, quien se encargará de responder las diferentes aplicaciones configuradas y proporcionar el encriptado SSL.

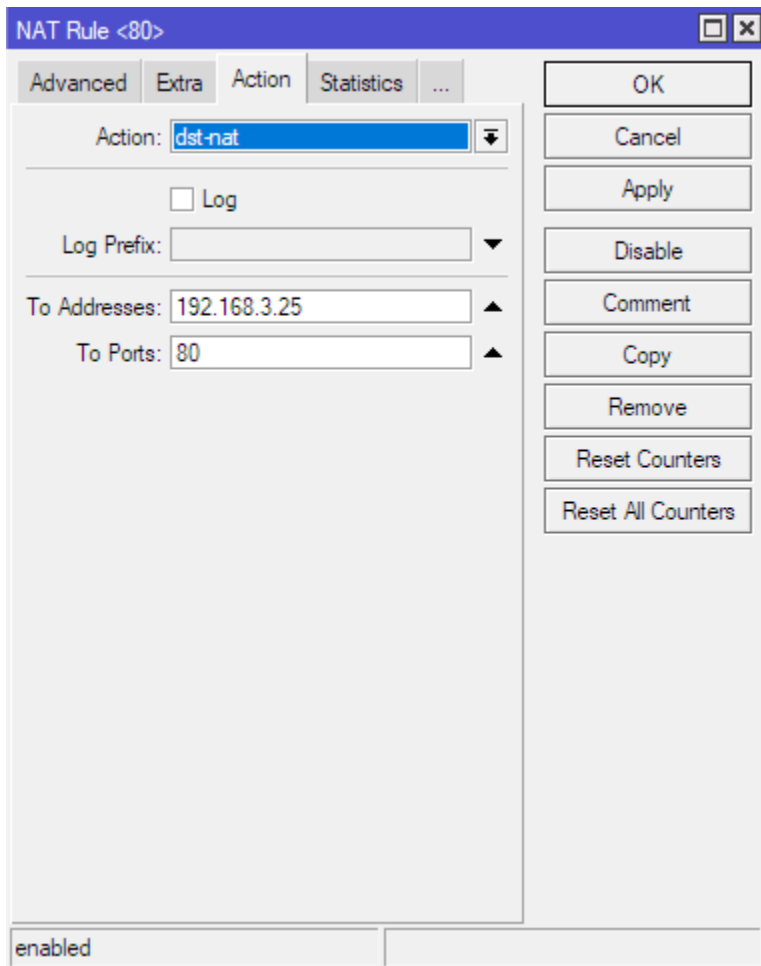


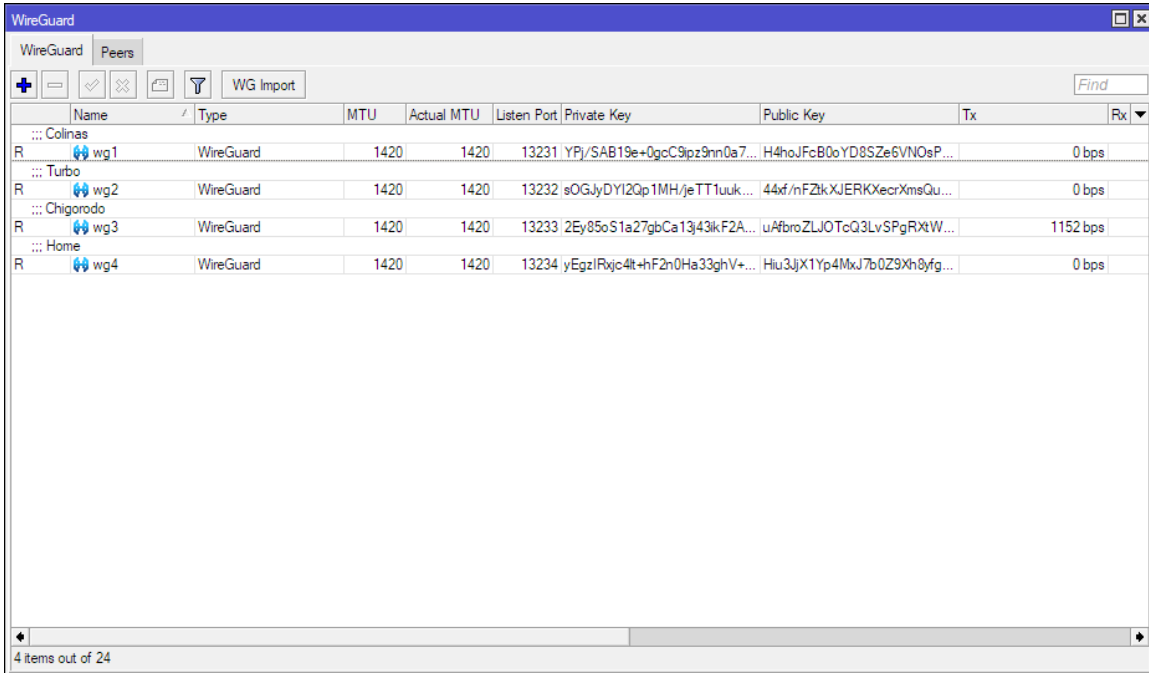
Ilustración 39. Regla NAT de puerto 80 a Nginx.

De esta manera, se puede acceder desde cualquier parte del mundo con conexión a internet a la aplicación SATT.

Algunas sedes, sin embargo, fueron conectadas con VPN sitio a sitio a través de túneles Wireguard, los cuales ofrecen mayor rapidez y una criptografía inigualable lo que los hace fiables, rápidos y extremadamente seguros.

TRABAJO DE GRADO

182



The screenshot shows the WireGuard Peers window with the following data:

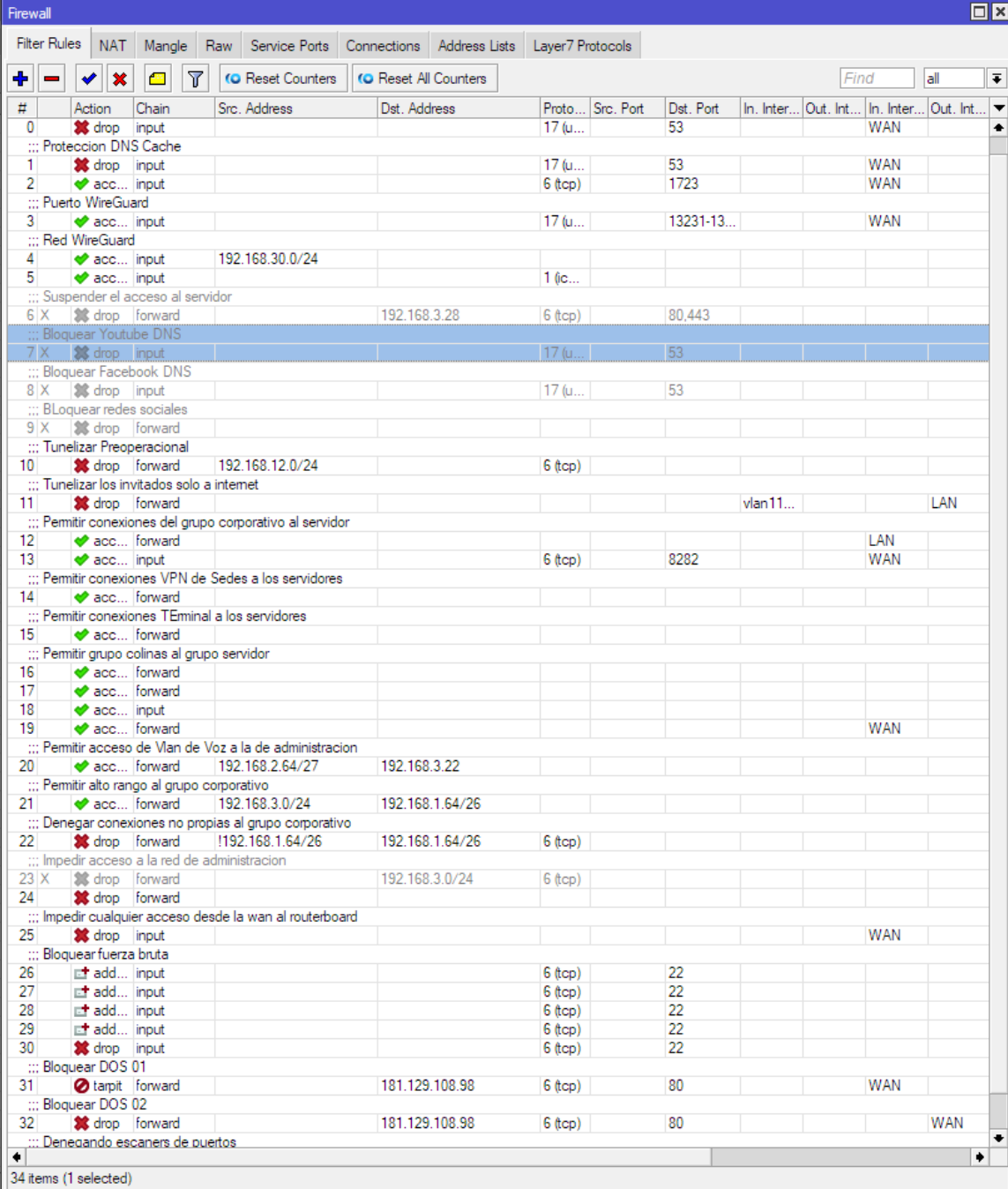
Name	Type	MTU	Actual MTU	Listen Port	Private Key	Public Key	Tx	Rx
Colinas wg1	WireGuard	1420	1420	13231	YPj/SAB19e+0gcC9lpz9nn0a7...	H4hoJFcB0oYD8SZe6VNOsP...	0 bps	
Turbo wg2	WireGuard	1420	1420	13232	sOGJyDY12Qp1MH/jeTT1uuk...	44xf/nFZlkXJERKXecrXmsQu...	0 bps	
Chigorodo wg3	WireGuard	1420	1420	13233	2Ey85oS1a27gbCa13j43kF2A...	uAfbroZLJOTcQ3LvSPgRXtW...	1152 bps	
Home wg4	WireGuard	1420	1420	13234	yEgzlRxc4lt+hF2n0Ha33ghV+...	Hiu3ljX1Yp4MxJ7b0Z9Xh8yfg...	0 bps	

Ilustración 40. Túneles Wireguard para conexión a sedes.

También se configuró reglas de firewall de segunda generación basados en listas ACL, ya que con eso contaba el cliente para tratar de mantener los intrusos al margen

TRABAJO DE GRADO

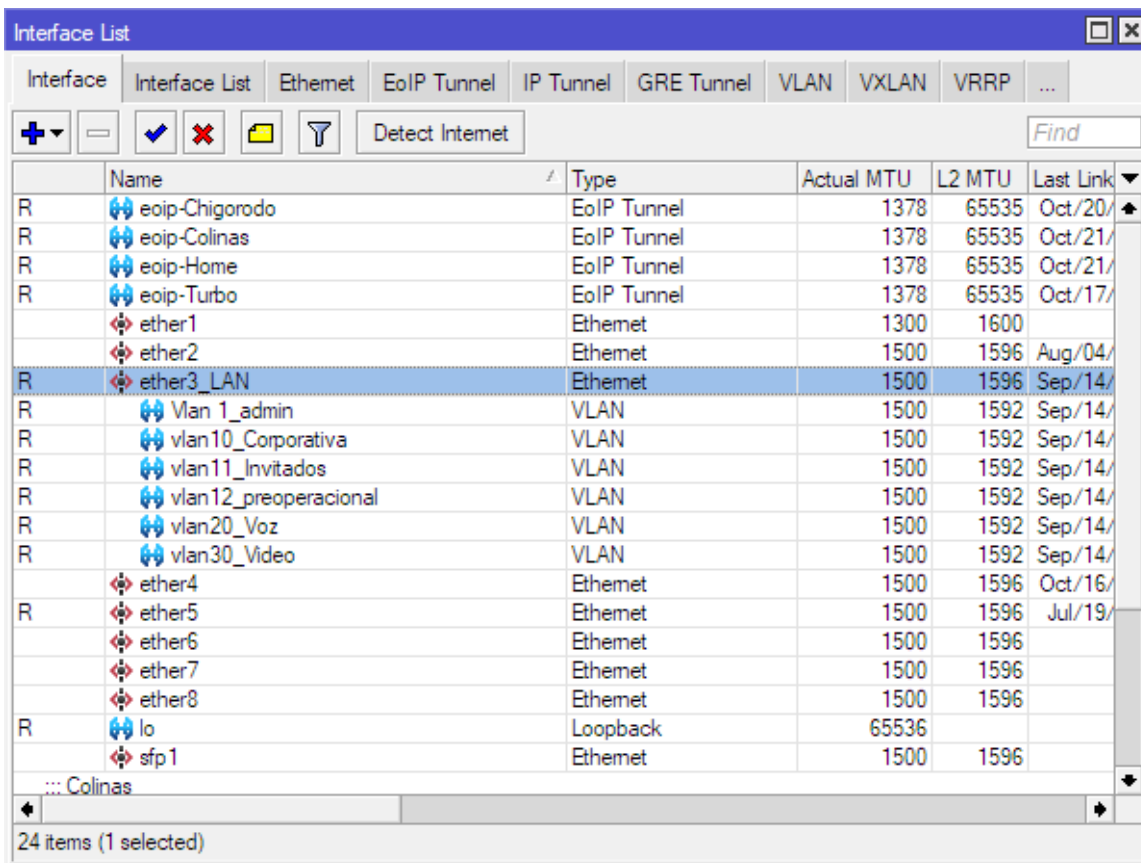
183



#	Action	Chain	Src. Address	Dst. Address	Proto.	Src. Port	Dst. Port	In. Inter...	Out. Int...	In. Inter...	Out. Int...
0	drop	input			17 (u...)		53			WAN	
::: Proteccion DNS Cache											
1	drop	input			17 (u...)		53			WAN	
2	acc...	input			6 (tcp)		1723			WAN	
::: Puerto WireGuard											
3	acc...	input			17 (u...)		13231-13...			WAN	
::: Red WireGuard											
4	acc...	input	192.168.30.0/24								
5	acc...	input			1 (ic...)						
::: Suspender el acceso al servidor											
6	drop	forward		192.168.3.28	6 (tcp)		80,443				
::: Bloquear Youtube DNS											
7	drop	input			17 (u...)		53				
::: Bloquear Facebook DNS											
8	drop	input			17 (u...)		53				
::: BLoquear redes sociales											
9	drop	forward									
::: Tunelizar Preoperacional											
10	drop	forward	192.168.12.0/24		6 (tcp)						
::: Tunelizar los invitados solo a internet											
11	drop	forward						vlan11...			LAN
::: Permitir conexiones del grupo corporativo al servidor											
12	acc...	forward								LAN	
13	acc...	input			6 (tcp)		8282			WAN	
::: Permitir conexiones VPN de Sedes a los servidores											
14	acc...	forward									
::: Permitir conexiones TEminal a los servidores											
15	acc...	forward									
::: Permitir grupo colinas al grupo servidor											
16	acc...	forward									
17	acc...	forward									
18	acc...	input									
19	acc...	forward								WAN	
::: Permitir acceso de Vlan de Voz a la de administracion											
20	acc...	forward	192.168.2.64/27	192.168.3.22							
::: Permitir alto rango al grupo corporativo											
21	acc...	forward	192.168.3.0/24	192.168.1.64/26							
::: Denegar conexiones no propias al grupo corporativo											
22	drop	forward	!192.168.1.64/26	192.168.1.64/26	6 (tcp)						
::: Impedir acceso a la red de administracion											
23	drop	forward		192.168.3.0/24	6 (tcp)						
24	drop	forward									
::: Impedir cualquier acceso desde la wan al routerboard											
25	drop	input								WAN	
::: Bloquear fuerza bruta											
26	add...	input			6 (tcp)		22				
27	add...	input			6 (tcp)		22				
28	add...	input			6 (tcp)		22				
29	add...	input			6 (tcp)		22				
30	drop	input			6 (tcp)		22				
::: Bloquear DOS 01											
31	drop	forward		181.129.108.98	6 (tcp)		80			WAN	
::: Bloquear DOS 02											
32	drop	forward		181.129.108.98	6 (tcp)		80				WAN
::: Denegando escaners de puertos											

Ilustración 41. ACL de firewall

Esto no podría haberse logrado sin una segmentación correcta de la red y creación de VLAN's según roles del negocio



	Name	Type	Actual MTU	L2 MTU	Last Link
R	eoip-Chigorodo	EoIP Tunnel	1378	65535	Oct/20/
R	eoip-Colinas	EoIP Tunnel	1378	65535	Oct/21/
R	eoip-Home	EoIP Tunnel	1378	65535	Oct/21/
R	eoip-Turbo	EoIP Tunnel	1378	65535	Oct/17/
	ether1	Ethernet	1300	1600	
	ether2	Ethernet	1500	1596	Aug/04/
R	ether3_LAN	Ethernet	1500	1596	Sep/14/
R	Vlan 1_admin	VLAN	1500	1592	Sep/14/
R	vlan10_Corporativa	VLAN	1500	1592	Sep/14/
R	vlan11_Invitados	VLAN	1500	1592	Sep/14/
R	vlan12_preoperacional	VLAN	1500	1592	Sep/14/
R	vlan20_Voz	VLAN	1500	1592	Sep/14/
R	vlan30_Video	VLAN	1500	1592	Sep/14/
	ether4	Ethernet	1500	1596	Oct/16/
R	ether5	Ethernet	1500	1596	Jul/19/
	ether6	Ethernet	1500	1596	
	ether7	Ethernet	1500	1596	
	ether8	Ethernet	1500	1596	
R	lo	Loopback	65536		
	sfp 1	Ethernet	1500	1596	

24 items (1 selected)

Ilustración 42. Interfaces VLAN

De esta manera se creó el siguiente esquema:

Tabla 13. Segmentación de Red y Distribución de VLANS

VLAN	Direccionamiento	Finalidad
1 Admin	192.168.3.0/24	Es la red nativa de dispositivos. Se crea por compatibilidad con ciertos equipos o conmutadores que requieren explícitamente la existencia de la VLAN 1. Todos los equipos de red y servidores obtienen IP en este segmento.

TRABAJO DE GRADO

185

10 corporativa	192.168.1.64/26	Es la red de computadores propios de la oficina. Las estaciones de trabajo de propiedad de la empresa al uso de empleados van en este segmento.
11 invitados	10.10.10.9/24	Esta red es proporcionada para aquellos visitantes o celulares de colaboradores. Tiene reglas de restricciones fuertes con control de ancho de banda.
12 preoperacional	192.168.12.0/24	Es una red creada para permitir a los conductores realizar preoperacional cuando sus dispositivos no cuenten con conexión a internet. Esta red solo permite al acceso a la aplicación.
20 voz	192.168.2.64/27	En esta red se conectan los dispositivos telefónicos de la compañía para manejo de protocolos VOIP.
30 video	192.168.15.64/28	Red que permite la conexión de circuitos cerrados de televisión.

TRABAJO DE GRADO

186

Resultados y Discusión

El software fue diseñado e implementado en la infraestructura de la empresa patrocinadora. El equipo estuvo conformado por dos personas. Un Full Stack Developer quien se encargó de la construcción del core del sistema. Este core es la base de componentes sobre los cuales se construyó el aplicativo. Representa elementos fundamentales como el navbar, datagrid y componentes personalizados como el combobox, inputs y otros componentes base. Por su parte la concesión del proyecto, estructura de requerimientos, análisis, diseño de la base de datos, despliegue, capacitación y mantenimiento estuvo a mi cargo. A medida que los requerimientos se complejizaban estuve ayudando con la escritura del todo el sistema ya que los requerimientos, aunque parecían simples, requerían gran construcción de estructuras y codificación.

El sistema resultó ser un éxito y aunque algunos usuarios se están acoplando, reconocen que se acomoda muy bien a la operación de la empresa. Sin embargo, su proceso evolutivo apenas comienza.

Conviene realizar auditorías de seguridad, ya que se implementaron las más básicas como enrutamiento obligado por el puerto 443 (HTTPS), conexiones cifradas, configuración de protección básica de interceptación de tráfico cruzado (XSS) en los middelwares de nextjs, entre otras protecciones comunes. Se ha realizado un escaneo del sitio a través de la plataforma web-check.xyz y se han obtenido estos resultados:

TRABAJO DE GRADO

187


Ilustración 43. Resultado de análisis de web-check.xyz

Corresponde, sin duda, seguir aprovisionando de elementos de seguridad como los que ofrece Cloudflare. Los cuales van en vanguardia con lo que se ve al día de hoy en ataques de seguridad. Aunque este no es un servicio que se paga por uso y las protecciones de Cloudflare están más orientadas a proteger los ataques de despilfarro de recursos y por lo tanto el pago de facturas impresionantes, conviene seguir con actividades pensadas para la mejora continua de la seguridad de la aplicación al menos en cuanto ataques de denegación de servicio y de alta disponibilidad.

De otra parte, aunque está muy bien optimizado para trabajar desde los dispositivos de escritorio, conviene realizar ajustes poco a poco de temas como la accesibilidad y el rendimiento en dispositivos móviles. En un principio y por modelo arquitectónico, se dio prioridad a la fiabilidad, usabilidad, funcionalidad y portabilidad, sin embargo, en dispositivos móviles no está totalmente optimizado y eso lo podemos comprobar con PageSpeed Insights.

TRABAJO DE GRADO

188

 Celulares  Escritorio

 Diagnostica problemas de rendimiento

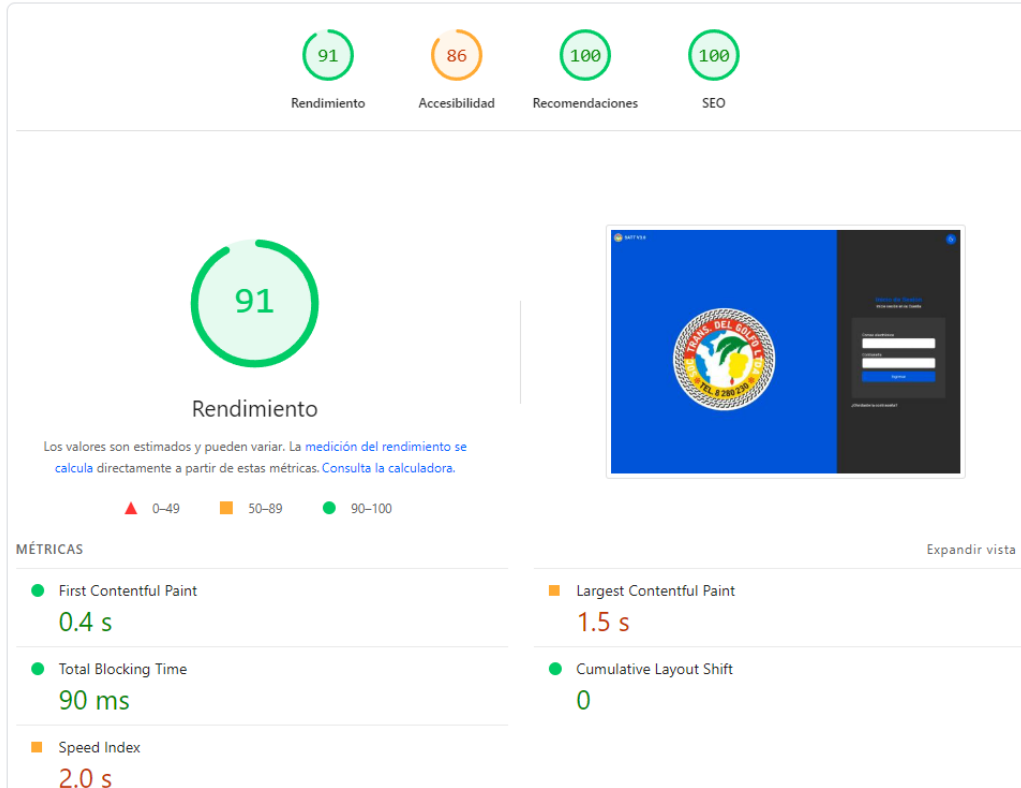


Ilustración 44. Resultado de análisis con PageSpeed Insights (escritorio).

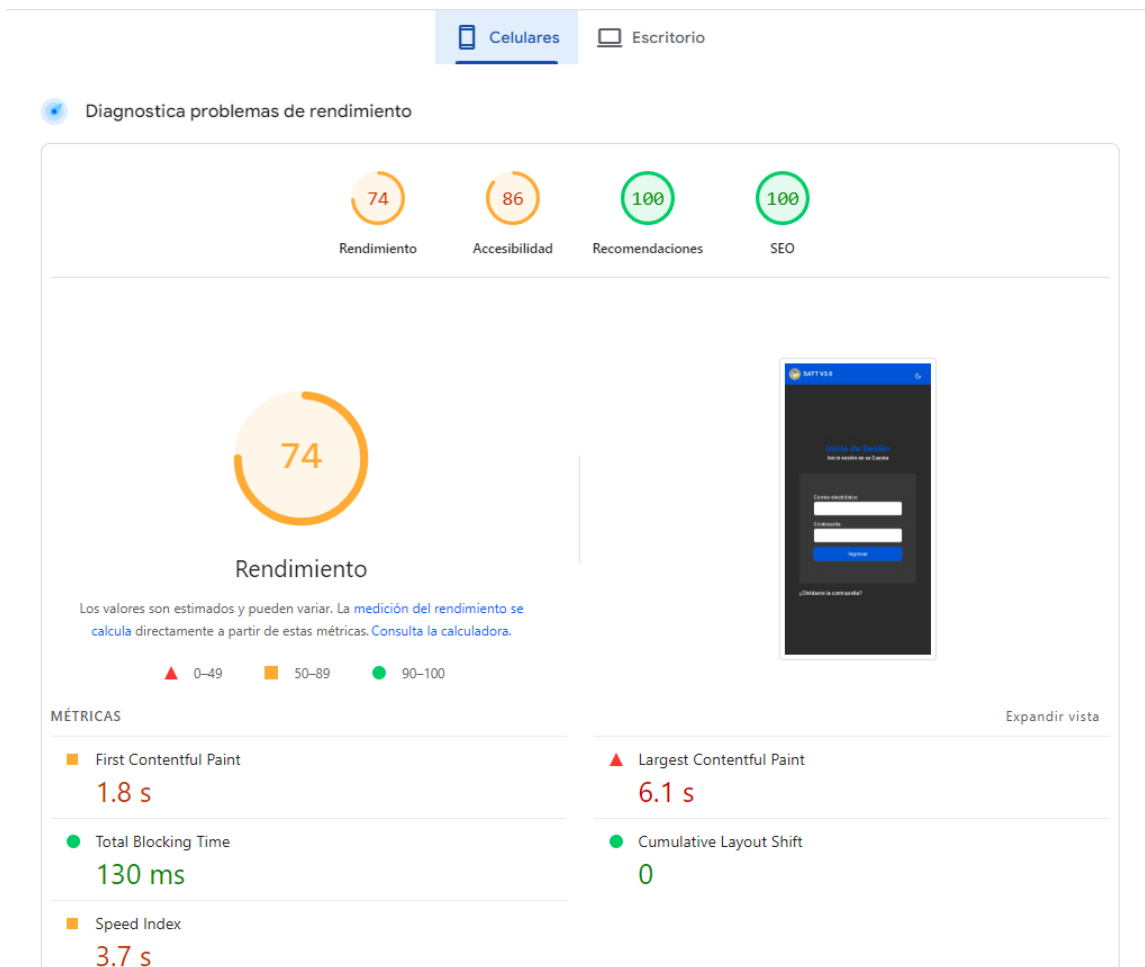


Ilustración 45. Resultado de análisis con PageSpeed Insights (versión móvil).

También conviene evaluar entornos de alta escalabilidad, en donde se introduzcan conceptos como el balanceo de carga y faillover; ya que en estos momentos la carga de trabajo es baja con no más de 25 usuarios concurrentes. Es altamente conveniente invertir en infraestructura que permita albergar al menos 350 usuarios concurrentes, aunque con la capacidad instalada puede albergar un valor cercano a ese.

TRABAJO DE GRADO

190

En cuanto al stack tecnológico, nextjs es un framework en constante desarrollo. En la etapa de mantenimiento perfectivo, conviene mucho estar revisando en entornos controlados si se pueden hacer actualizaciones de versión sin romper la funcionalidad programada. Muchas de las librerías son constantemente actualizadas y algunas podrían tener revisiones de seguridad importantes. Es por ello que el proceso de observabilidad es altamente importante en proyectos como éste.

TRABAJO DE GRADO

191

Conclusiones

Todas las empresas en algún momento van a necesitar sistemas de información que mejoren y optimicen sus procesos misionales. Sin duda la ingeniería del software demuestra que los procesos sistematizados maximizan la productividad de cualquier punto de la industria y corresponde a los profesionales juiciosamente aplicar las metodologías y estrategias necesarias que mejor se adapten a esos procesos.

Muchos sectores comerciales aun no creen en la evolución digital, consideran que eso solo genera gastos innecesarios que impactan negativamente la rentabilidad, sin embargo, cuando se cuantifican las perdidas por falta de control, espacio, capacidad operativa, personal y otros factores, es posible viabilizar un proyecto de software. Sin duda esa fue una decisión acertada de la empresa de transporte, una de esos sectores reacios al cambio, por cierto, que en todo caso sí fue necesaria la realización de inversiones pero que agregan controles importantes que impiden fugas de dinero.

El objetivo sin duda, era transformar por completo como la empresa realizaba sus procesos, desde manuales hasta algunos técnicamente mejor elaborados, en una sola base de datos correctamente conectada e integrada. Que permitiera ser más eficientes, pero sobre todo efectivos. Los objetivos fueron claramente logrados con un porcentaje de aceptación del 100%.

Durante el proceso, como expertos en la materia, pudimos observar que las metodologías en cascada son mayormente lentas de mucha planificación y contexto. Algunos requisitos fueron descartados en fase de diseño y por lo tanto no fueron codificados; algunos otros seguramente tuvieron leves cambios por normatividad recién

TRABAJO DE GRADO

192

salida, lo que sin duda pudimos manejar correctamente pero que en el fondo nos dejan un aprendizaje claro: todo cambia con más velocidad de lo que notamos. De modo que las metodologías ágiles están llamadas a posicionarse en una era donde los procesos son altamente cambiantes y los sistemas deben responder rápidamente con sus adaptaciones.

Aun con lo anterior, las metodologías ágiles traerán otros retos que deberán abordarse de la forma mas profesional posible pero que sin duda tendrán mejor respuesta al mercado.

De otro lado, el proceso de codificación es un mundo inmenso, muchos frameworks y tecnologías se desarrollan constantemente para hacer la vida del desarrollador mas fácil. Lograr sinergia en este sentido puede apalancar cualquier proyecto de ingeniería de software. Utilizar el ecosistema NPM nos permitió centrarnos en solucionar los requisitos funcionales mas que en los no funcionales, pero sobre todo permitió hacer mucho más con menos. Si bien no es simplemente magia, es sin duda se sintió más productivo. Usar componentes y bibliotecas de otros desarrolladores hacen la diferencia entre el éxito y el fracaso (eligiendo bien eso sí, porque hay miles de cosas) porque al final los proyectos son esfuerzos temporales y económicos que, si no mantienen equilibrada la balanza, alguna parte interesada seguramente va a salir perjudicada.

El compromiso es, en síntesis, nunca dejar de aprender, pues es el proceso mas importante que involucra cualquier proyecto de tecnología. Se aprende en cada modelo, en cada requisito, en cada línea de código; con cada cliente, cada implicado, siempre hay un constante aprendizaje porque todo evoluciona increíblemente rápido. Seguramente después de algunos meses posteriores al despliegue muchas de nuestras librerías usadas serán

TRABAJO DE GRADO

193

marcadas como obsoletas por NPM y será necesaria una actualización. Seguramente habrá alguna vulnerabilidad que corregir, alguna funcionalidad que optimizar: todo cambia cada minuto. De modo que seguir aprendiendo es la forma de no quedar obsoletos como profesionales. Por su parte, esperamos que la empresa en algún momento tenga un stack tecnológico que pueda cubrir muchos procesos críticos de software como análisis de seguridad y alta disponibilidad.

En cuanto a lo académico... (suspiro). Solo ruego que tampoco paren de innovar (docentes), de cambiar la forma como enseñan tecnología, que vivan este mundo en carne propia y enseñen con pasión esta carrera, pues no basta simplemente aprender algoritmos, PHP o JAVA. Es importante también orientar al estudiante a profundizar en algo pero que le permita desarrollar la habilidad de abstraer. No es difícil aprender un lenguaje de programación, pero lleva tiempo. No es difícil escribir diferentes proyectos similares en varios lenguajes, pero lleva tiempo. Considero que si la estrategia es aprender JAVA o PHP o Python puede ser más provechoso tomar uno o máximo dos y entender hasta lo más profundo que no quede en duda ni una sola expresión sintáctica, que aprender muchos lenguajes para simplemente saber mostrar un “hola mundo” en pantalla. Entender los conceptos arquitectónicos no tiene discusión, pero saber escribir código no se puede tomar folclóricamente, pues es un proceso que se descuida en la academia y resulta ser tan importante como la concepción misma del proyecto.

TRABAJO DE GRADO

194

Referencias

Función Pública. (26 de 05 de 2015). *Decreto 1079*. Obtenido de

www.funcionpublica.gov.co:

https://www.funcionpublica.gov.co/eva/gestornormativo/norma_pdf.php?i=77889

Ministerio de Transporte. (30 de Diciembre de 1993). *Ley 105*. Bogota DC, Colombia.