



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO  
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA  
ÁREA ACADÉMICA DE COMPUTACIÓN Y ELECTRÓNICA

---

---

**T E S I S**

**“Sistema Administrativo de Autotransporte de  
Carga (TMS)”**

**QUE PARA OBTENER EL TÍTULO DE:**

LICENCIADO EN CIENCIAS COMPUTACIONALES

**PRESENTA:**

José Andrés Cruz Martínez

**Directores:**

Asesor: Ing. Ana Luisa Vargas Ramírez

Coasesor: Dr. Eduardo Cornejo Velázquez



Mineral de la Reforma, Hgo., a 26 de septiembre de 2025

Número de control: ICBI-D/1715/2025  
**Asunto:** Autorización de impresión.

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

Con Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio, le comunico que el Jurado asignado al egresado de la Licenciatura en Ciencias Computacionales **José Andrés Cruz Martínez**, quien presenta el trabajo de titulación "**Sistema Administrativo de Autotransporte de carga (TMS)**", ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

**Presidente:** Mtra. Fabiola Martínez Juárez

**Secretario:** Dr. Eduardo Cornejo Velázquez

**Vocal:** Mtra. Ana Luisa Vargas Ramírez

**Supiente:** Dr. Edgar Olguín Guzmán

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

Atentamente  
 "Amor, Orden y Progreso"

Mtro. Gabriel Velasco Rodríguez  
 Director de ICBI



GVR/YCC

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

"Amor, Orden y Progreso"



## Contenido.

Contenido.....	1
Imágenes.....	3
Tablas.....	4
Resumen.....	5
Abstract.....	6
Capítulo 1.- Introducción.....	7
1.1.- Definición del problema.....	9
1.2.- Propuesta de solución.....	10
1.3.- Objetivos.....	10
1.3.1.- Objetivo general.....	10
1.3.2.- Objetivos específicos.....	10
1.4.- Justificación.....	11
1.5.- Herramientas utilizadas.....	13
1.6.- Limitaciones.....	13
Capítulo 2.- Marco teórico.....	14
2.1.- Sistema de Administración de Transporte (TMS).....	14
2.2.- Principales Funcionalidades de un TMS.....	14
2.3.- Beneficios de la Adopción de un TMS.....	15
2.4.- Arquitectura del TMS: Cliente-Servidor.....	15
2.5.- Herramientas de desarrollo.....	18
2.5.1.- Visual Studio Code.....	20
2.5.2.- Python.....	21
2.5.3.- MySQL.....	23
2.5.4.- Flask.....	24
2.5.5.- JavaScript.....	26
2.5.6.- HTML y CSS.....	26
2.5.7.- Seguridad: HTTP y HTTPS.....	27
Capítulo 3.- Estado del arte.....	30
3.1.- Sistemas TMS comerciales.....	30
3.1.1.- Logistaas.....	32

3.1.2.- Dispatch Science.....	33
3.1.3.- Kuebix.....	35
3.1.4.- MercuryGate.....	37
<b>Capítulo 4.- Metodología.....</b>	<b>39</b>
4.1.- Justificación de la Metodología de Cascada.....	42
4.2.- Aplicación Estructurada de la Metodología de Cascada.....	43
4.2.1.- Comunicación.....	43
4.2.2.- Planeación.....	44
4.2.3.- Modelado.....	44
4.2.4 Construcción.....	44
4.2.5.- Despliegue.....	45
<b>Capítulo 5.- Planeación y Modelado.....</b>	<b>46</b>
5.1.- Casos de uso.....	46
5.2.- Diagrama de clases.....	50
5.3.- Diagrama de secuencia.....	52
5.4.- Diseño de la Base de Datos.....	55
5.5.- Diseño de la interfaz de usuario(UI).....	57
<b>Capítulo 6.- Desarrollo de la aplicación.....</b>	<b>60</b>
6.1.- Creación de la Base de Datos en MySQL Server.....	60
6.2.- Configuración de Flask y Python.....	60
6.3.- Desarrollo del código.....	62
6.4.- Construcción de la UI.....	63
6.5.- Conexión con la base de datos.....	67
6.6.- Conexión con el dominio público.....	70
6.7.- Beneficios observados para la organización.....	72
<b>Trabajos futuros.....</b>	<b>73</b>
<b>Conclusiones.....</b>	<b>75</b>
<b>Anexos.....</b>	<b>76</b>
Anexo 1.- Manual de usuario.....	76
Anexo 2.- Cuestionario.....	86
Anexo 3.- Carta reconocimiento.....	94
<b>Referencias bibliográficas.....</b>	<b>95</b>

## Imágenes.

Imagen 1: Ejemplo ilustrativo de un TMS.....	14
Imagen 2: Diagrama de arquitectura Cliente-Servidor.....	16
Imagen 3: Diagrama conceptual de las herramientas usadas. ....	18
Imagen 4: Logo de Logistaas. ....	32
Imagen 5: Logo de Dispatch Science.....	34
Imagen 6: Logo de Kuebix.....	35
Imagen 7: Logo de MercuryGate.....	37
Imagen 8: Metodología de cascada acorde a Roger S. Pressman .....	42
Imagen 9: Diagrama de Casos de Uso. ....	47
Imagen 10: Diagrama de clases. ....	51
Imagen 11: Diagrama de secuencia.....	53
Imagen 12: Diagrama Relacional.....	55
Imagen 13: Diseño de la UI.....	58
Imagen 14: Diagrama de la conexión de Flask y Python.....	61
Imagen 15: Extensiones de Python instaladas en Visual Studio Code.....	61
Imagen 16: Diagrama conceptual del funcionamiento del sistema.....	63
Imagen 17: Página de inicio de sesión de la aplicación. ....	65
Imagen 18: Sección de inicio donde se muestran graficas. ....	66
Imagen 19: Sección de viajes donde se observan todos los registros de viajes.....	66
Imagen 20: Ventana de detalles de un viaje en específico.....	67
Imagen 21: Aplicación ejecutándose en el dominio tlacu.org.....	71
Imagen 22: Página de inicio de sesión.....	77
Imagen 23: Panel de datos. ....	78
Imagen 24: Página de facturas. ....	79
Imagen 25: Información detallada de los operadores. ....	79
Imagen 26: Gráficas de actividad de los operadores.....	80
Imagen 27: Página de viajes. ....	81
Imagen 28: Log del viaje.....	82

<b>Imagen 29: Página crear viaje.....</b>	<b>83</b>
<b>Imagen 30: Página detalle viaje.....</b>	<b>84</b>
<b>Imagen 31: Página editar viaje.....</b>	<b>84</b>
<b>Imagen 32: Página de inicio con botón salir a la vista. ....</b>	<b>85</b>
<b>Imagen 33: Gráfico de la pregunta 1 .....</b>	<b>87</b>
<b>Imagen 34: Gráfico de la pregunta 2. ....</b>	<b>88</b>
<b>Imagen 35: Gráfico de la pregunta 3. ....</b>	<b>88</b>
<b>Imagen 36: Gráfico de la pregunta 4. ....</b>	<b>89</b>
<b>Imagen 37: Gráfico de la pregunta 5.....</b>	<b>89</b>
<b>Imagen 38: Gráfico de la pregunta 6. ....</b>	<b>90</b>
<b>Imagen 39: Gráfico de la pregunta 8. ....</b>	<b>90</b>
<b>Imagen 40: Gráfico de la pregunta 9. ....</b>	<b>91</b>
<b>Imagen 41: Gráfico de la pregunta 11. ....</b>	<b>92</b>
<b>Imagen 42: Gráfico de la pregunta 12. ....</b>	<b>92</b>
<b>Imagen 43: Gráfico de la pregunta 13. ....</b>	<b>93</b>
<b>Imagen 44: Gráfico de la pregunta 14. ....</b>	<b>93</b>

## **Tablas.**

<b>Tabla 1: Herramientas utilizadas.....</b>	<b>19</b>
<b>Tabla 2: Tabla comparativa de metodologías.....</b>	<b>41</b>
<b>Tabla 3: Formato de Caso de Uso "Administrar Viajes".....</b>	<b>48</b>
<b>Tabla 4: Formato de Caso de Uso "Mandar registros".....</b>	<b>49</b>
<b>Tabla 5: Formato de Caso de Uso "Rastrear envíos".....</b>	<b>50</b>

## Resumen.

La tesis se centra en el desarrollo de un Sistema de Gestión de Transporte (TMS) como respuesta a la necesidad de modernización tecnológica en empresas de transporte de carga, considerando las limitaciones de recursos y las características específicas del sector al que se dirige el estudio.

El problema principal radica en la complejidad de administrar de manera simultánea un número creciente de empleados, contratos y convenios con nuevos socios, lo cual genera sobrecarga administrativa, duplicidad de tareas y dificultades para mantener la integridad de los datos en tiempo real.

La solución propuesta consiste en diseñar e implementar un TMS utilizando tecnologías de fácil acceso como Python, Flask, MySQL y Visual Studio Code, estableciendo módulos para la gestión de usuarios, viajes y recursos, junto con políticas de seguridad que incluyen autenticación con contraseñas robustas y encriptación de datos mediante conexiones seguras HTTPS. El proceso de desarrollo sigue la metodología en cascada, complementada con análisis de requerimientos, modelado UML y normalización de bases de datos para garantizar la consistencia e integridad en la gestión de la información.

Los resultados muestran que el sistema centraliza operaciones críticas, optimiza el acceso y modificación de datos de acuerdo con privilegios de usuario, incrementa la eficiencia en la gestión de viajes y genera reportes gráficos que apoyan la toma de decisiones estratégicas. Todo esto se desarrolla en un entorno seguro, flexible y adaptable a diferentes escalas operativas.

En conclusión, el TMS representa una alternativa funcional y escalable para empresas de transporte de carga que requieren fortalecer su control operativo con recursos limitados. Además, se configura como un prototipo replicable para pequeñas y medianas empresas del sector logístico, contribuyendo a la innovación tecnológica y a la optimización de los procesos administrativos y operativos.

## **Abstract.**

This thesis focuses on the development of a Transportation Management System (TMS) as a response to the need for technological modernization in freight forwarding companies, considering resource limitations and the specific characteristics of the sector under study.

The main problem lies in the complexity of simultaneously managing a growing number of employees, contracts, and agreements with new partners, which generates administrative overload, duplication of tasks, and difficulties in maintaining real-time data integrity.

The proposed solution consists of designing and implementing a TMS using readily available technologies such as Python, Flask, MySQL, and Visual Studio Code, establishing modules for user, trip, and resource management, along with security policies that include strong password authentication and data encryption using secure HTTPS connections. The development process follows the waterfall methodology, complemented by requirements analysis, UML modeling, and database normalization to ensure consistency and integrity in information management. The results show that the system centralizes critical operations, optimizes data access and modification according to user privileges, increases efficiency in trip management, and generates graphical reports that support strategic decision-making. All of this is developed in a secure, flexible environment adaptable to different operational scales.

In conclusion, the TMS represents a functional and scalable alternative for freight forwarding companies that need to strengthen their operational control with limited resources. Furthermore, it serves as a replicable prototype for small and medium-sized companies in the logistics sector, contributing to technological innovation and the optimization of administrative and operational processes.

## Capítulo 1.- Introducción.

La industria del autotransporte se ha convertido en un engrane fundamental de cualquier economía y sociedad a nivel mundial, se encarga de la distribución tanto de bienes como de personas. Representa un gran impulso económico debido a toda la inversión necesaria para llevar a cabo operaciones de autotransporte que a su vez genera empleos en una gran variedad de roles y sectores dentro de la misma industria, así como afuera.

En México el sector de autotransporte se ha convertido en uno de los pilares de la economía nacional. “Este moviliza cerca del 82% de carga terrestre en México, lo cual aporta alrededor de un 3.3% del PIB de la economía nacional y equivale a una cifra de alrededor de 556 millones de toneladas por año, de acuerdo a un estudio del INEGI en 2020 dentro del papel "Conociendo La Industria del Autotransporte de Carga". Centrándose más en el sector de Autotransporte de Carga, se estima que distribuye el 56% de la carga nacional, lo cual contrasta con un 31.6% del transporte marítimo y un 12.8% proveniente del transporte ferroviario.” (INEGI, 2021)

En la actualidad, el transporte y la logística están experimentando una evolución constante, se debe a la rápida influencia de la tecnología del entorno y al aumento en las expectativas de calidad de servicio que los clientes demandan. Aunque predecir el futuro siempre resulta desafiante, es posible vislumbrar que las próximas décadas en la evolución del transporte estarán marcadas por una intensificación aún mayor de la digitalización de los procesos y una optimización integral en todas sus facetas. En esta perspectiva, es probable que los siguientes elementos sean los ejes centrales de evolución:

- **Eficiencia en el transporte:** Se busca gestionar mejor los recursos para aumentar la rapidez y efectividad, reduciendo costos y mejorando la experiencia del usuario.
- **Sostenibilidad en el transporte:** La reducción del impacto ambiental es clave, impulsando modelos y estrategias más sostenibles en el sector.
- **Digitalización integral de los procesos:** La digitalización transforma la logística, agilizando operaciones mediante plataformas tecnológicas para clientes y empresas.
- **Enfoque centrado en el cliente:** Se prioriza optimizar la experiencia del cliente, especialmente en el servicio de última milla, para fomentar la satisfacción y fidelización.

- **Sinergia reforzada entre comercio electrónico y logística:** El crecimiento del e-commerce impulsa y fortalece la relación con la logística, vital para la distribución y el comercio minorista.

Para llevar a cabo operaciones de semejante magnitud las empresas de autotransporte de carga por lo general hacen uso de sistemas administrativos generalmente llamados sistemas de gestión del transporte o TMS por sus siglas en inglés; son plataformas de logística que utilizan tecnologías como lo son las bases de datos, GPS, etc.; para planificar, ejecutar y optimizar el movimiento físico de mercancías, tanto de entrada como de salida asegurándose de que el envío cumpla con las normas, y la documentación adecuada esté disponible.

Por lo general, sistemas como estos suelen formar una parte de un sistema más grande, denominado como sistema de gestión de la cadena de suministro o SCM por sus siglas en inglés. Desempeñan una función central en las cadenas de suministro, ya que impactan en cada etapa del proceso, desde la planificación, adquisiciones de la logística, así como la gestión del ciclo de vida. La profundidad de visión proporcionada por un sistema robusto conduce a una administración más eficiente, resultando en una mayor satisfacción del cliente. De igual manera (INEGI, 2021), estimula un aumento en las ventas que favorece el crecimiento empresarial. Dada la dinámica actual del entorno de comercio global donde se opera y realiza transacciones, es esencial contar con un sistema que permita navegar exitosamente por procesos complejos relacionados con políticas y cumplimiento normativo comercial.

Las plataformas de Gestión de Transporte (TMS) han revolucionado por completo la forma en que las empresas abordan las operaciones de transporte y logística. Su impacto va más allá de simplemente mejorar procesos; han impulsado una verdadera transformación en la gestión interna de la cadena de suministro. La evolución es crucial en un entorno en el que la tecnología cambia rápidamente y las expectativas de los clientes son cada vez más altas. Como resultado, las empresas que adoptan y aprovechan eficazmente las capacidades de un TMS se distinguen claramente como líderes en la industria, como ejemplificado por empresas como Amazon, Wal-Mart, Mercado Libre, Castores, etc.

## **1.1.- Definición del problema.**

La problemática principal detectada es la dificultad para administrar de forma precisa la información operativa dentro de pequeñas y medianas empresas de autotransporte de carga debido a la falta de sistemas especializados de gestión lo que impide optimizar los procesos logísticos, mantener la integridad de los datos y asegurar la escalabilidad de las operaciones.

Para contextualizar, en las últimas décadas se ha visto el auge de las empresas de autotransporte nacionales e internacionales debido a la continua tendencia de globalización, ocasionando que compañías busquen maneras más eficientes de administrar, vigilar y contabilizar cada uno de los traslados que realizan. Lográndolo mediante el uso de sistemas diseñados específicamente para las empresas. Sin embargo, no todas las compañías pueden permitirse el uso de tales sistemas ya sea por cuestiones monetarias, de accesibilidad o adaptabilidad, prefiriendo utilizar software comercial como Excel o Access, lo que genera procesos y órdenes más susceptibles a errores debido al factor de error humano al manejar grandes cantidades de datos junto con la posibilidad de incompatibilidad o pérdida de información al realizar actividades entre diferentes plataformas.

El desarrollo del sistema se realizó con una empresa específica en mente como caso de estudio, debido a la facilidad de contacto con esta mediante la familiaridad de ciertos empleados con el autor del programa y la consideración de que esta representa el sector al que se dirige el proyecto. La empresa CANACE Transportes S.A. de C.V. ubicada en Carretera Jaltepec Km. 2.5 Col. Santa María El Chico, C.P 43710 Tulancingo de Bravo. CANACE se dedica al sector económico de transportes de carga pesada como lo son minerales para las industrias metalúrgicas, cementeras y vidrieras de la región.

Actualmente se encuentra en una etapa de crecimiento sostenido derivada de la firma de múltiples contratos y convenios con nuevos socios comerciales. Esta expansión ha generado un incremento considerable en su plantilla laboral, alcanzando aproximadamente 300 empleados. Tal volumen de personal representa un desafío significativo en términos de administración operativa, ya que la gestión manual de datos, horarios, rutas, incidencias y procesos administrativos resulta cada vez más compleja y propensa a errores. Este contexto justifica la necesidad de implementar un sistema que permita optimizar la gestión de recursos humanos, operaciones logísticas y procesos administrativos dentro de la organización.

## **1.2.- Propuesta de solución.**

La tesis propone el desarrollo un TMS para pequeñas y medianas empresas (PyMEs), otorgando una alternativa flexible y económicamente accesible frente a las soluciones comerciales existentes, cuyo costo puede representar una barrera significativa. Alojando funciones de la logística administrativa necesaria para el adecuado funcionamiento de la empresa en su totalidad como lo es el monitoreo del viaje de una unidad, la contabilidad de las operaciones, el mantenimiento y la cartera de clientes, por mencionar algunas.

Además del impacto económico, la tesis busca generar un efecto social positivo al fortalecer la competitividad de las PyMEs, facilitando su digitalización y promoviendo un enfoque administrativo moderno dentro del sector del autotransporte.

## **1.3.- Objetivos.**

### **1.3.1.- Objetivo general.**

Diseñar e implementar un Sistema de Gestión de Transporte (TMS) para mejorar el desempeño de las empresas que utilizan software de oficina en sus procesos administrativos, incorporando características tecnológicas de adaptabilidad dentro del sector de autotransporte en Tulancingo, Hidalgo, tomando como base a la empresa CANACE Transportes S.A. de C.V.

### **1.3.2.- Objetivos específicos.**

- Analizar los procesos que interactúan en la administración del transporte de carga para obtener los requerimientos del sistema como entidades, activos e información con la que se opera.
- Identificar la interacción entre entidades dentro y fuera de la organización, a través de la elaboración de casos de uso UML, con el fin de identificar la información y los momentos de su intercambio, realizando diversos casos de uso.
- Diseñar el sistema siguiendo la metodología en cascada, basándose en el análisis previo de la interacción de los procesos operativos.
- Programar el diseño antes obtenido para transformarlo en un sistema a través de herramientas tecnológicas, empleando Python con el framework Flask para la lógica de negocio, MySQL para la gestión de datos y tecnologías web (HTML, CSS, JavaScript) para la interfaz de usuario.

- Comprobar la funcionalidad del sistema a través de sesiones de retroalimentación con usuarios clave, documentando su interacción con las diferentes funciones implementadas.
- Establecer un repositorio estructurado con el código fuente y documentación técnica del sistema, usando GitHub como plataforma de control de versiones, acompañado de un manual de usuario que facilite su comprensión, uso e implementación por parte del personal operativo.

#### **1.4.- Justificación.**

La presente tesis tiene como objetivo apoyar a las PyMEs en el establecimiento de sistemas administrativos mediante el desarrollo de un software que permita organizar los procedimientos operativos necesarios, al optimizar el tiempo de operación centralizando y consolidando información dispersa en herramientas como Excel, Access, SAP o QuickBooks, comúnmente utilizadas para gestionar rutas, inventarios, facturación y el seguimiento de flotas en empresas de autotransporte de carga. Así facilitando los procesos administrativos dentro de la organización, lo cual puede traducirse en beneficios económicos, al mismo tiempo que se garantiza la seguridad e integridad de los datos e información tanto de los operadores como de los clientes. Algunos de las principales razones que justifican esta propuesta son:

- **Costo-beneficio:** El sector de logística de carga es extremadamente competitivo, donde cualquier ventaja en la agilidad de las operaciones y el manejo de clientes presenta un gran avance que puede dictar la supervivencia de una empresa. También se menciona que las PyMEs, a menudo enfrentan dificultades para costear un TMS debido a sus limitaciones financieras, la complejidad de implementación que requiere recursos adicionales, la falta de experiencia en tecnología para evaluar y seleccionar adecuadamente un TMS, y la necesidad de priorizar inversiones en áreas consideradas más críticas para la competitividad de la empresa, ya que actualmente se observa una variación según factores como el tipo de licencia (SaaS o perpetua), el alcance de la implementación, el soporte y mantenimiento, la capacitación del personal, en caso de implementaciones locales, los gastos de infraestructura. Las suscripciones mensuales pueden oscilar entre \$100 MXN(pesos mexicanos) y \$10,000 MXN, mientras que las

licencias perpetuas pueden costar entre \$10,000 MXN y \$250,000 MXN. La implementación básica puede rondar los \$10,000 - \$50,000 MXN, mientras que las personalizadas pueden superar los \$500,000 MXN. Además, hay que considerar costos adicionales como el mantenimiento anual (15%-25% del costo de la licencia) y la capacitación del personal (\$1,000 - \$10,000 MXN). En el caso de soluciones locales, se deben sumar los gastos de infraestructura (María Rosa López Mejía, 2010). El TMS propuesto presenta una ventaja a los costos mencionados debido a su adaptabilidad al contexto económico de una empresa ya que su código se presenta disponible en un repositorio y la entidad que desee implementarlo puede hacerlo bajo sus presupuestos.

- **Comunicación:** Las empresas de transporte de carga también enfrentan grandes desafíos al integrar todos sus sistemas operativos, como facturación, rastreo y mantenimiento. Los sistemas suelen operar de manera independiente, utilizando diversas plataformas que no siempre son compatibles entre sí. Resultando en problemas de interoperabilidad entre los diferentes departamentos dentro de la empresa. Además, la integración puede ser costosa y compleja, requiriendo una planificación detallada y la adopción de soluciones tecnológicas avanzadas para garantizar una operación fluida y eficiente. El TMS presenta una solución al permitir la intercomunicación de múltiples sistemas mediante tecnologías discutidas más adelante.
- **Facilidad de uso:** Otro punto a considerar, que los empleados de las empresas de transporte de carga a menudo enfrentan desafíos significativos al tener que manejar múltiples sistemas para realizar sus tareas diarias. La complejidad de navegar entre diferentes plataformas puede resultar en errores, retrasos y una curva de aprendizaje prolongada para los nuevos empleados. Por ejemplo, los sistemas independientes utilizados son Access para el manejo de información de cliente y facturas, gomotive para GPS, Excel para costos, etc. Con el TMS se anticipa una mejora considerable a su usabilidad. Un sistema centralizado permitirá a los empleados acceder de manera más intuitiva a todas las herramientas necesarias desde una única interfaz. Facilitando la estandarización de procesos y la integración de datos, lo que promoverá una

operación más eficiente, menos propensa a errores, beneficiando tanto a los empleados como a la empresa en su conjunto.

### **1.5.- Herramientas utilizadas.**

- Herramientas para la lógica del servidor, la gestión de datos y la construcción de funcionalidades:
  - Visual Studio Code
  - Python
  - MySQL Workbench
  - MySQL Server
  - Flask
  - HTTPS
- Herramientas para la construcción de la interfaz de usuario:
  - JavaScript
  - HTML
  - CSS

### **1.6.- Limitaciones.**

- El servidor se ejecuta en una computadora genérica, sin utilizar infraestructura dedicada.
- La migración completa de las operaciones empresariales al TMS aún está en proceso, utilizándose actualmente solo para pruebas.
- La sensibilidad de las operaciones ha llevado a mantener el sistema en fase de pruebas, sin trasladarlo aún a un entorno productivo vinculado con sistemas de terceros.
- Actualmente, el despliegue se realiza sin un nombre de dominio asociado a la empresa por razones de conflictos de interés.

## Capítulo 2.- Marco teórico.

El sector del transporte desempeña un papel crucial en la sustentabilidad y crecimiento de las economías contemporáneas, actuando como el motor que facilita la circulación eficaz de personas y mercancías. Una gestión eficiente ayuda a reducir gastos, aumentar la eficacia operacional asegurando la integridad tanto de los bienes como de los seres humanos en tránsito. En el marco teórico se ofrece como herramienta para explorar, comprender los múltiples elementos, retos que entran en juego en la creación y puesta en marcha de un sistema administrativos logísticos enfocado en el transporte.

### 2.1.- Sistema de Administración de Transporte (TMS).

“Un Sistema de Administración del Transporte (TMS, por sus iniciales en inglés) es una solución integral que centraliza y orquesta las actividades logísticas dentro de una empresa. Esta tecnología recopila y maneja datos críticos sobre las operaciones de transporte, permitiendo una comunicación efectiva entre distintos actores como proveedores, transportistas y clientes finales.” (Imagen 1)



Imagen 1: Ejemplo ilustrativo de un TMS.

(Oracle, 2024)

### 2.2.- Principales Funcionalidades de un TMS.

A continuación, se describen algunas de las principales capacidades que un TMS aporta a la operación logística moderna:

- **Seguimiento en tiempo real:** Una de las capacidades más valiosas de un TMS es ofrecer un monitoreo continuo y en tiempo real de todos los elementos logísticos, desde el estado del pedido hasta las condiciones de tránsito.
- **Armonización de operaciones logísticas:** Gracias a la disponibilidad de datos en tiempo real, un TMS posibilita una sincronización efectiva de las diferentes etapas que componen la cadena logística, lo que se traduce en una mayor agilidad operativa.

- **Maximización de la eficiencia:** Al combinar el seguimiento en tiempo real con la capacidad de sincronizar operaciones, el TMS contribuye a la optimización integral de la cadena de suministro. Lo cual no solo resulta en ahorro de costos, sino también en un nivel de servicio más elevado y confiable.

### 2.3.- Beneficios de la Adopción de un TMS.

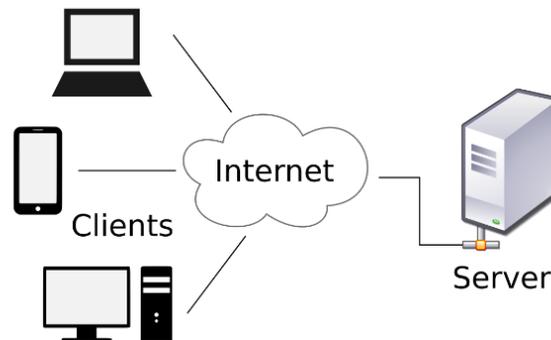
La incorporación de un TMS impacta beneficiosamente dentro de diferentes áreas o procesos como los presentadas a continuación:

- **Visibilidad completa de la cadena de suministro:** Un TMS proporciona una perspectiva global y actualizada en tiempo real de toda la cadena de suministro, asegurando que los datos estén siempre precisos y relevantes para la toma de decisiones logísticas.
- **Control avanzado de inventarios:** Utilizar un TMS permite una mayor eficacia en la gestión de inventarios, minimizando así los costos asociados con el almacenamiento y ajustando las existencias a la demanda real de manera más eficiente.
- **Precisión en la facturación:** La automatización de los procesos de pago y facturación a través del TMS reduce las posibilidades de errores atribuibles al factor humano, mejorando la fiabilidad en este ámbito crítico.
- **Eficiencia en la logística de transporte:** La incorporación de un TMS en las operaciones logísticas conduce a una mayor eficiencia en el transporte, desde la selección de la mejor ruta hasta el seguimiento exhaustivo de los paquetes y la recopilación de métricas relevantes, lo que contribuye a la sostenibilidad del proceso.
- **Reducción integral de gastos:** Uno de los retornos más notables de la implementación de un TMS es la disminución global de costos en las operaciones logísticas. Esta eficiencia se traduce en mayores márgenes de beneficio para la empresa sin sacrificar la calidad del servicio proporcionado.

### 2.4.- Arquitectura del TMS: Cliente-Servidor.

La arquitectura cliente-servidor [\(Imagen 2\)](#) resulta esencial en la implementación de un Sistema de Gestión de Transporte, ya que permite separar las funciones de procesamiento y

presentación de la información. En este modelo, el servidor centraliza la gestión de los datos, los procesos y la lógica del sistema, mientras que los clientes (usuarios) acceden a estos servicios a través de interfaces específicas. Esta estructura facilita un acceso controlado a la información desde distintos puntos, mejora la escalabilidad del sistema y permite una administración más ordenada de los recursos tecnológicos.



*Imagen 2: Diagrama de arquitectura Cliente-Servidor.*

“Cliente/servidor es a menudo un término genérico que se usa para cualquier arquitectura de aplicación que divide el procesamiento entre dos o más procesos, a menudo en dos o más máquinas. Cualquier aplicación de base de datos es una aplicación cliente/servidor si maneja el almacenamiento y la recuperación de datos en el proceso de la base de datos y la manipulación y presentación de los datos en otro lugar. El servidor es el motor de base de datos que almacena los datos, y el cliente es el proceso que obtiene o crea los datos. La idea detrás de la arquitectura cliente/servidor en una aplicación de base de datos es proporcionar acceso a los mismos datos a múltiples usuarios” (Java, 2003).

El principal propósito de la arquitectura cliente-servidor es dividir las funciones y responsabilidades del software en diferentes capas. La separación de responsabilidades es fundamental para un diseño de software eficiente. Una de las divisiones más claras y esenciales es precisamente la que establece la arquitectura cliente-servidor.

Los beneficios de separación son variados. Entre ellos, se encuentra las siguientes:

- Facilita la administración de datos y la seguridad de la información en servidores controlados.

- Mejora la escalabilidad, permitiendo actualizaciones y expansiones independientes en el cliente y el servidor.
- Permite el mantenimiento independiente de las capas de software, aislando problemas.
- Ofrece flexibilidad para trabajar con diferentes tipos de clientes.
- Favorece el desarrollo independiente, ya que cliente y servidor pueden ser gestionados por equipos distintos.

Un TMS necesita coordinar procesos como el seguimiento de envíos, la asignación de recursos, la planificación de rutas, etc. La arquitectura permite separar las responsabilidades al asignar a los clientes la interacción con los datos, mientras el servidor se encarga de su almacenamiento y procesamiento. De tal forma, se asegura la disponibilidad de los datos en tiempo real para múltiples usuarios, garantizando una gestión eficaz de las operaciones logísticas.

La arquitectura se puede clasificar en 3 principales categorías que serían las siguientes:

- **Arquitectura de dos niveles (2-Tier):** Es el modelo tradicional en el que los clientes se conectan directamente con el servidor sin intermediarios. Aquí, el servidor maneja tanto el almacenamiento de datos como la lógica de negocio, mientras que el cliente se encarga de la presentación y algunas tareas ligeras de procesamiento. Es simple de implementar, pero puede resultar menos escalable a medida que aumenta la cantidad de usuarios o la complejidad del sistema.
- **Arquitectura de tres niveles (3-Tier):** En el presente modelo, se introduce una capa intermedia entre el cliente y el servidor, conocida como capa de lógica de negocio o middleware. El intermediario gestiona el procesamiento y las reglas del negocio, separando la presentación (cliente) del acceso a los datos (servidor). Lo cual mejora la mantenibilidad y escalabilidad del sistema, ya que las modificaciones pueden hacerse en la capa intermedia sin afectar a las demás, aunque añade cierta complejidad en el desarrollo y gestión del sistema.
- **Arquitectura de n niveles (N-Tier):** En tal arquitectura, el sistema se divide en múltiples capas o niveles, más allá de los tres tradicionales, donde cada capa tiene una responsabilidad específica. Además de las capas de presentación, lógica de

negocio y datos, pueden añadirse niveles adicionales para manejo de autenticación, caché, integración con servicios externos, entre otros. Esta estructura es altamente escalable y flexible, permitiendo una mayor modularidad y mejor distribución de las responsabilidades, pero también aumenta la complejidad y el costo de implementación y mantenimiento.

## 2.5.- Herramientas de desarrollo.

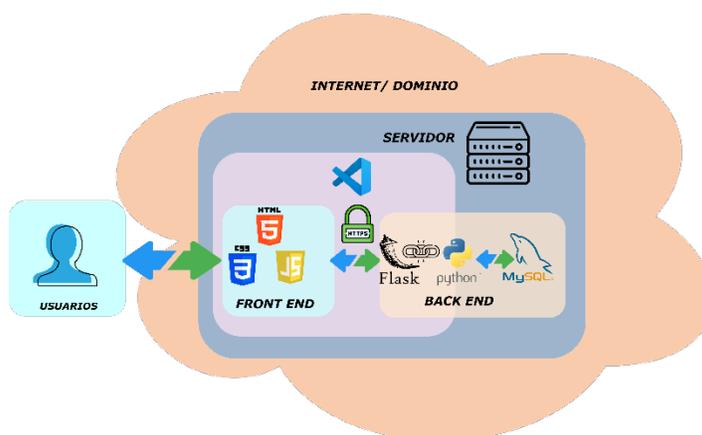


Imagen 3: Diagrama conceptual de las herramientas usadas.

Para el desarrollo de un TMS se recurre al uso de diversas herramientas (Imagen 3 y Tabla 1) que permiten estructurar tanto la interfaz de usuario (frontend en inglés) como la lógica del servidor (backend en inglés) de la aplicación. El frontend corresponde a la parte visible e interactiva del sistema, es decir, la interfaz que utilizan los usuarios para acceder a las funciones del TMS. En este ámbito, se emplean tecnologías como HTML, CSS y JavaScript para construir interfaces dinámicas, visualmente organizadas y funcionales. Por otro lado, el backend se encarga de la lógica del sistema, el procesamiento de datos y la conexión con la base de datos. Aquí se utilizan herramientas como Python y el framework Flask, que permiten crear rutas, controlar flujos de información y gestionar la comunicación con la base de datos desarrollada en MySQL. Todo este entorno de desarrollo se gestiona y edita mediante el entorno de desarrollo integrado Visual Studio Code (VS Code). Además, para asegurar la transferencia de datos de forma segura entre el cliente y el servidor, se utiliza el protocolo HTTPS, que cifra la información y protege la integridad de las comunicaciones.

	Herramienta	Función	Descripción	Característica
<b>Backend</b>	Visual Studio Code	Entorno de desarrollo	Plataforma utilizada para escribir, organizar y depurar el código tanto de la lógica del sistema como de la interfaz visual	Multiplataforma, ligero y con amplia compatibilidad.
	Python	Desarrollo de la lógica del sistema	Lenguaje que permite construir la lógica del sistema, procesar datos, gestionar la base de datos y generar las respuestas web	Sintaxis simple y gran cantidad de librerías.
	Flask	Framework de desarrollo web	Framework que permite crear aplicaciones web, gestionar rutas y procesar las peticiones de los usuarios	Microframework flexible y liviano.
	Mysql Server	Motor de base de datos	Sistema donde se almacenan y gestionan todos los datos operativos del TMS, como viajes, rutas, operadores y facturación	Alta fiabilidad y soporte para grandes volúmenes.
	Mysql Workbench	Gestión de base de datos	Herramienta para la gestión visual de la base de datos, permitiendo modelado, consultas SQL y mantenimiento estructural de la información	Interfaz intuitiva con diagramación visual de datos.
	HTTPS	Seguridad de la comunicación	Protocolo que cifra la información transmitida entre el usuario y el servidor, garantizando la confidencialidad de los datos	Cifrado robusto mediante certificados SSL/TLS.
<b>Frontend</b>	HTML	Estructura del contenido visual	Lenguaje de marcado que define la estructura y el contenido de las páginas web del sistema	Estándar universal para la creación de páginas web.
	CSS	Diseño y estilo visual	Aplica estilos visuales como colores, tipografías, tamaños y disposición, mejorando la experiencia visual del usuario	Separa el contenido del diseño, facilitando cambios.
	Javascript	Interactividad en la interfaz	Permite dotar a la interfaz de comportamientos dinámicos, validaciones y respuestas a interacciones del usuario	Ejecuta funciones directamente en el navegador.

Tabla 1: Herramientas utilizadas.

### 2.5.1.- Visual Studio Code.

Es la plataforma principal para codificar el proyecto, permitiendo la implementación del backend, el diseño del frontend y la conexión con la base de datos. Su interfaz intuitiva, la organización por carpetas, la navegación entre archivos y la terminal integrada optimizaron el flujo de trabajo, agilizando las etapas de desarrollo desde la codificación hasta las pruebas, y facilitando los procesos de depuración. Aunque existen otras opciones viables como Sublime Text o Atom, Visual Studio Code ofrece un entorno más completo y adaptable a las necesidades del sistema.

“Visual Studio Code (VSCode) es un editor de código fuente desarrollado por Microsoft que ofrece un entorno de desarrollo ligero pero potente. Es gratuito y de código abierto, diseñado para ser altamente personalizable y compatible con una amplia variedad de lenguajes de programación.” (Visual Studio Code, 2023)

Algunas de sus principales características por las cuales fue escogido como editor de código para el sistema son:

- **Multiplataforma:** Compatible con Windows, macOS y Linux.
- **Extensibilidad:** Admite extensiones para lenguajes, herramientas y características adicionales.
- **Integración con Git:** Herramientas integradas para control de versiones y colaboración.
- **Resaltado y autocompletado:** Resaltado de sintaxis y autocompletado para múltiples lenguajes.
- **Depuración integrada:** Depuración de errores directamente en el editor.
- **Terminal integrada:** Ejecución de comandos desde el entorno.
- **Temas y personalización:** Personalización del aspecto con temas y configuraciones.
- **Integración en la nube:** Compatible con servicios como Azure para desarrollo e implementación en la nube.

Algunas de las ventajas que presenta VSCode frente a otros editores de código son las siguientes: es ligero y rápido lo que ofrece un rendimiento eficiente con un inicio ágil mejorando así la productividad del desarrollador. Además, cuenta con una comunidad activa

que contribuye con extensiones brindando soporte continuo garantizando actualizaciones y generando mejoras constantes. Su interfaz de usuario es intuitiva haciéndola fácil de usar lo que lo hace accesible para desarrolladores de todos los niveles de experiencia. VSCode también destaca por su versatilidad en la compatibilidad con una amplia variedad de lenguajes de programación lo que lo hace adecuado para proyectos diversos. Asimismo, se integra con herramientas populares de desarrollo como Docker y Kubernetes entre otras. Finalmente, su licencia de código abierto permite a los desarrolladores personalizar y adaptar el entorno según sus necesidades.

### **2.5.2.- Python.**

El lenguaje Python fue utilizado para desarrollar el backend del TMS, incluyendo funcionalidades como la creación y edición de viajes, la autenticación de usuarios, la gestión de unidades, entre otras. También permitió manejar la lógica de conexión con la base de datos y la comunicación cliente-servidor mediante el framework Flask. La claridad y simplicidad de su sintaxis facilitaron el desarrollo modular y el mantenimiento del sistema. Existen otras alternativas como Java, C++ o Go, estas requieren estructuras más estrictas y mayor complejidad en la configuración inicial. En comparación, Python permite un desarrollo más ágil y accesible, lo cual resulta especialmente útil en proyectos donde se prioriza la velocidad de implementación sin comprometer la legibilidad ni la escalabilidad

“Python es un versátil lenguaje de programación de alto nivel que se utiliza en diversas áreas como desarrollo web, ciencia de datos, aprendizaje automático y desarrollo de software en general. Este lenguaje es apreciado por su eficiencia y accesibilidad, compatible con múltiples plataformas y disponible de forma gratuita. Acelera el proceso de desarrollo y se integra fácilmente con otros lenguajes populares como Java, C y C++.” (Amazon Web Services, Inc, 2024)

#### **Beneficios de Python:**

- La sintaxis de Python es intuitiva y fácil de leer, facilitando el desarrollo y la comprensión del código.
- Se caracteriza por su eficiencia, permitiendo desarrollar programas con menos líneas de código en comparación con otros lenguajes.

- Ofrece una vasta biblioteca estándar con código predefinido para numerosas tareas, ahorrando tiempo y esfuerzo en el desarrollo.
- Goza de una comunidad global activa, proporcionando una red sólida de soporte y recursos didácticos en línea para aprender y mejorar las habilidades en Python.
- Posee la capacidad de funcionar en diversos sistemas operativos, desde Windows y macOS hasta Linux y Unix.

### Características de Python

- **Python como lenguaje interpretado:** Python opera como un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea en tiempo real, en lugar de ser compilado previamente. Esta característica facilita el proceso de depuración, ya que cualquier error en el código resulta en la interrupción inmediata del programa, permitiendo a los desarrolladores identificar y corregir fallos de manera más eficiente.
- **Facilidad de uso de Python:** El diseño de Python se centra en la legibilidad y facilidad de uso. A diferencia de otros lenguajes que utilizan símbolos como llaves para delimitar bloques de código, Python emplea sangrías para esta función. Además, su sintaxis se asemeja al inglés, lo que hace que el código sea más fácil de leer y entender.
- **Python como lenguaje de tipado dinámico:** Una de las grandes ventajas de Python es su tipado dinámico. Significando que no es necesario declarar explícitamente los tipos de variables al escribir el código. Python los determina automáticamente durante la ejecución, lo que acelera el desarrollo permitiendo una mayor flexibilidad en el manejo de datos.
- **Python como lenguaje de alto nivel:** Es un lenguaje de alto nivel, significando que está más cerca del lenguaje humano que del código máquina. Liberando a los desarrolladores de preocupaciones relacionadas con aspectos de bajo nivel, como la administración de memoria y la arquitectura del sistema, permitiéndoles centrarse en la lógica y funcionalidad de la aplicación.
- **Orientación a objetos y versatilidad de Python:** Python es inherentemente un lenguaje orientado a objetos, lo que facilita la estructuración del código. Sin embargo, no se limita a este paradigma; también es apto para programación estructurada y

funcional. Esta versatilidad lo convierte en una excelente opción para una amplia gama de proyectos, desde simples scripts hasta sistemas complejos.

### 2.5.3.- MySQL.

MySQL funciona como la base de datos central del sistema, encargada de almacenar la información operativa. Almacena registros de rutas, viajes, operadores, clientes, facturación, entre otros. La conexión con Python, mediante consultas parametrizadas y el uso de un pool de conexiones, optimiza la seguridad y eficiencia en el procesamiento de datos. Aunque existen otras opciones como PostgreSQL, SQLite o MongoDB, MySQL fue preferido por su equilibrio entre rendimiento, facilidad de integración con las herramientas ya seleccionadas y amplia documentación. PostgreSQL, aunque más robusto en algunas funciones avanzadas, requiere una configuración más detallada; SQLite, por su parte, es más adecuada para aplicaciones locales o de bajo volumen; mientras que MongoDB, al ser una base de datos NoSQL, no se ajusta del todo a la estructura relacional requerida por el sistema.

“MySQL es un sistema de bases de datos de Oracle que se utiliza en todo el mundo para gestionar bases de datos. Se basa en el álgebra relacional y se utiliza principalmente para el almacenamiento de datos de diversos servicios web. Los CMS más conocidos que utilizan MySQL son, por ejemplo, WordPress y TYPO3.” (IONOS, ¿Qué es MySQL?, 2023)

Para este proyecto se hizo uso de las herramientas MySQL Server y MySQL Workbench las cuales se pueden definir como:

- **MySQL Server:** “Es el servidor de bases de datos de MySQL, el componente principal que gestiona todas las operaciones de la base de datos. Este servidor es responsable de todas las tareas como almacenamiento, manipulación y recuperación de datos. Se ejecuta como un servicio en segundo plano en el sistema operativo y escucha las solicitudes provenientes de las aplicaciones cliente a través de la red o del mismo equipo. MySQL Server trabaja con bases de datos relacionales y utiliza el lenguaje SQL (Structured Query Language) para realizar consultas y administrar datos.” (IONOS, ¿Qué es MySQL?, 2023)
- **MySQL Workbench:** “Es una herramienta gráfica integrada para la administración y el diseño de bases de datos en MySQL. Proporciona diversas funcionalidades como

modelado de datos, desarrollo de SQL, administración de servidores, monitoreo, entre otros.” (MySQL, 2023) La interfaz gráfica facilita la visualización y manipulación de estructuras de bases de datos, permitiendo a los usuarios ejecutar consultas, administrar usuarios y permisos, monitorizar el rendimiento y realizar otras tareas sin necesidad de utilizar la línea de comandos.

Ambos componentes, MySQL Server y MySQL Workbench, se complementan entre sí. Mientras que MySQL Server es el corazón que maneja la base de datos, MySQL Workbench actúa como un puente entre el usuario y el servidor, facilitando la administración y desarrollo de bases de datos.

Con la finalidad de lograr una arquitectura Cliente-Servidor en donde se logre el acceso de múltiples clientes se hace uso de un pool de conexiones. Un pool de conexiones en MySQL Server es un mecanismo que gestiona un grupo de conexiones a la base de datos que pueden ser reutilizadas por múltiples clientes o procesos. En lugar de abrir y cerrar una conexión a la base de datos cada vez que se realiza una solicitud, manteniendo un conjunto de conexiones abiertas disponibles para ser utilizadas, lo que optimiza el rendimiento y la eficiencia.

Aparte con la finalidad de garantizar la seguridad de consultas hechas por los clientes al server de MySQL se hace uso de consultas parametrizadas en SQL siendo una técnica que permite ejecutar consultas de base de datos de manera segura y eficiente, utilizando parámetros en lugar de concatenar cadenas de texto para construir las consultas. Ayudando a prevenir ataques de inyección SQL.

#### **2.5.4.- Flask.**

Flask fue el framework utilizado para estructurar la parte del servidor del TMS. Permitiendo definir rutas web, gestionar solicitudes-respuestas, e interactuar con la base de datos, lo que contribuyó a una organización estructurada del backend. Su arquitectura ligera favoreció una implementación ágil de las funcionalidades, asegurando adaptabilidad para futuras actualizaciones.

La arquitectura se basó en el patrón Modelo-Vista-Controlador (MVC), con la siguiente distribución:

- **Modelo:** Gestionó los datos y las operaciones en la base de datos mediante consultas SQL y funciones en Python.
- **Vista:** Consistió en plantillas HTML que definieron la interfaz mostrada al usuario.
- **Controlador:** Implementado a través de rutas y funciones en Flask, actuó como intermediario entre el modelo y la vista, procesando las interacciones del usuario y actualizando el sistema.

Un framework es un conjunto de reglas y convenciones que se usan para desarrollar software de manera más eficiente y rápida. Dichos marcos de trabajo se emplean para ahorrar tiempo y esfuerzo en el desarrollo de aplicaciones, ya que proporcionan una estructura básica que se puede utilizar como punto de partida.

“Flask es un micro-framework de desarrollo web en Python que permite crear aplicaciones web de forma rápida y sencilla. Fue desarrollado inicialmente por Armin Ronacher en 2010 y ha ganado popularidad por su simplicidad y flexibilidad.” (IONOS, ¿Qué es Flask Python? un breve tutorial sobre este microframework, 2023). A diferencia de otros frameworks más completos como Django, Flask se centra en proporcionar las funcionalidades básicas necesarias para poner en marcha una aplicación web, dejando el resto a bibliotecas adicionales o al propio desarrollador.

Ventajas de Flask:

- **Simplicidad y rapidez:** Fácil de entender e implementar, centrándose en la lógica de la aplicación.
- **Flexibilidad:** Extensible con módulos y extensiones según necesidades del proyecto.
- **Desarrollo ágil:** Permite desarrollo y despliegue rápido de aplicaciones web.
- **Restful request dispatching:** Facilita creación de APIs RESTful de manera sencilla.
- **Soporte para Jinja2:** Usa Jinja2 para manipulación y presentación de datos.
- **Desarrollo orientado a pruebas:** Configura entornos de prueba fácilmente, ideal para TDD.
- **Documentación completa:** Extensa documentación y comunidad activa.
- **Ligereza:** Consume menos recursos, ideal para proyectos con limitaciones.
- **Compatibilidad con WSGI:** Versátil y funciona con varios servidores web.

### 2.5.5.- JavaScript.

JavaScript se utilizó para añadir dinamismo a la interfaz de usuario del TMS. Su implementación permitió validar formularios en tiempo real, responder a eventos como clics o envíos de datos de forma inmediata sin necesidad de recargar la página. Gracias a esto, la navegación dentro del sistema resultó más fluida, favoreciendo la interacción con módulos como el panel de control, los registros de viaje y los formularios de gestión.

“JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado.” (MDN, ¿Qué es JavaScript? - Aprende Desarrollo web, 2024)

Las principales características de javascript son:

- **Interpretado:** Se ejecuta en el navegador sin compilación, agilizando pruebas y desarrollo.
- **Tipado dinámico:** Variables cambian de tipo en ejecución, ofreciendo flexibilidad, pero riesgos.
- **Basado en objetos:** Usa prototipos para herencia y reutilización de código.
- **Alto nivel:** Abstrae complejidad, centrándose en lógica y funcionalidad.
- **Funciones de primera clase:** Funciones como objetos, permitiendo modularidad.
- **Interacción con DOM:** Manipula HTML y responde a eventos del usuario.
- **Asincronía:** Soporta callbacks, promesas y async/await para operaciones simultáneas.
- **Multiplataforma:** Funciona en navegadores y servidores (Node.js).
- **Ecosistema rico:** Bibliotecas y frameworks como React y Angular.
- **Comunidad activa:** Apoyo constante para aprendizaje y colaboración.

### 2.5.6.- HTML y CSS.

Se emplearon para construir la estructura y el diseño visual de la interfaz del sistema. HTML fue utilizado para definir el contenido de las páginas, mientras que CSS permitió aplicar

estilos y adaptaciones visuales para diferentes dispositivos. Esta combinación aseguró una experiencia visual clara, ordenada y profesional para los usuarios del TMS.

“HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).” (MDN, HTML: Lenguaje de etiquetas de hipertexto, 2024)

Mientras que a CSS lo podemos definir como “el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe cómo debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.” (MDN, CSS, 2024)

#### **2.5.7.- Seguridad: HTTP y HTTPS.**

Para garantizar la seguridad de los datos transmitidos entre cliente y servidor dentro del TMS, se adoptó el protocolo HTTPS como principal medida de protección. Esto aseguró que toda la comunicación que involucraba credenciales, consultas a la base de datos o descarga de información sensible se realizara mediante un canal cifrado. Además, se aplicaron prácticas adicionales como el uso de contraseñas seguras, autenticación de usuario y consultas parametrizadas para prevenir vulnerabilidades comunes como la inyección SQL.

“El protocolo de transferencia de hipertexto (HTTP) es un protocolo o conjunto de reglas de comunicación para la comunicación cliente-servidor. Cuando visita un sitio web, su navegador envía una solicitud HTTP al servidor web, que responde con una respuesta HTTP. El servidor web y su navegador intercambian datos como texto sin formato. En resumen, el protocolo HTTP es la tecnología subyacente que impulsa la comunicación de red. Como su nombre indica, el protocolo seguro de transferencia de hipertexto (HTTPS) es una versión más segura o una extensión de HTTP. En HTTPS, el navegador y el servidor establecen una conexión segura y cifrada antes de transferir datos.” (Amazon Web Services, 2024).

**HTTP:** Protocolo de la capa de aplicación en el modelo OSI (Interconexión de Sistemas Abiertos) que especifica varios tipos de solicitudes y respuestas. Por ejemplo, cuando se quiere acceder a datos de un sitio web, se usa una solicitud HTTP GET. Si se desea enviar

información, como al completar un formulario de contacto, se utiliza una solicitud HTTP PUT. Para visualizarlo de una manera más práctica si se quisiera acceder a <http://www.example.com>, el protocolo seguiría los siguientes pasos:

- **Iniciación de la solicitud:** El usuario ingresa una URL en el navegador o hace clic en un enlace.
- **Envío de la solicitud HTTP:** El navegador envía una solicitud HTTP al servidor correspondiente, indicando el método (como GET o POST) y el recurso que se desea acceder.
- **Procesamiento en el servidor:** El servidor recibe la solicitud, la procesa y busca el recurso solicitado (como una página web o un archivo).
- **Respuesta del servidor:** El servidor envía una respuesta HTTP al navegador, que incluye un código de estado (por ejemplo, 200 para indicar éxito) y el contenido del recurso solicitado.
- **Visualización del contenido:** El navegador recibe la respuesta, interpreta el contenido (como HTML, CSS o imágenes) y lo presenta al usuario como una página web.
- **Interacción adicional:** El usuario puede interactuar con la página, lo que puede generar más solicitudes HTTP para cargar más recursos o enviar datos, repitiendo el proceso.

**HTTPS:** Debido al problema de seguridad que presenta HTTP al momento de transmitir datos sin cifrado surgió HTTPS que combina las solicitudes y respuestas de HTTP con encriptación SSL (Secure Sockets Layer) y TLS (Transport Layer Security). Cuando los sitios web hacen uso de HTTPS deben adquirir un certificado SSL/TLS de una autoridad de certificación (CA) independiente. Este certificado se proporciona al navegador antes de realizar el intercambio de datos, lo que establece un nivel de confianza. Asimismo, el certificado incluye información criptográfica que facilita el intercambio de datos cifrados entre el servidor y los navegadores web. Por ejemplo, cuando se intenta ingresar a <https://www.youtube.com/watch?v=tjmk0C64eJg>, el protocolo sigue los siguientes pasos:

- **Iniciación de la solicitud:** El usuario ingresa una URL que comienza con "https://" en el navegador o hace clic en un enlace seguro.

- **Establecimiento de conexión segura:** Antes de enviar la solicitud, el navegador y el servidor establecen una conexión segura mediante el protocolo SSL/TLS, que cifra la comunicación para proteger los datos.
- **Envío de la solicitud HTTPS:** Una vez establecida la conexión segura, el navegador envía una solicitud HTTPS al servidor, indicando el método (como GET o POST) y el recurso que se desea acceder.
- **Procesamiento en el servidor:** El servidor recibe la solicitud, la procesa y busca el recurso solicitado (como una página web o un archivo).
- **Respuesta del servidor:** El servidor envía una respuesta HTTPS al navegador, que incluye un código de estado (por ejemplo, 200 para indicar éxito) y el contenido del recurso solicitado, también cifrado.
- **Visualización del contenido:** El navegador recibe la respuesta, descifra el contenido y lo presenta al usuario como una página web segura.
- **Interacción adicional:** El usuario puede interactuar con la página, lo que puede generar más solicitudes HTTPS para cargar más recursos o enviar datos, continuando el proceso de forma segura.

Existen diversas formas de garantizar la seguridad de los datos en una aplicación adicionales a la mencionada anteriormente, como el uso de contraseñas seguras, la autenticación de doble factor, consultas parametrizadas, entre otras. Algunas de estas estrategias se abordarán en detalle más adelante. Sin embargo, dado que este es un prototipo, HTTPS se destaca como la principal medida de seguridad del proyecto, proporcionando un cifrado robusto que protege la transmisión de información sensible entre el cliente y el servidor.

## Capítulo 3.- Estado del arte.

Las plataformas de Gestión de Transporte (TMS) han revolucionado por completo la forma en que las empresas abordan las operaciones de transporte y logística. Su impacto va más allá de simplemente mejorar procesos; han impulsado una verdadera transformación en la gestión interna de la cadena de suministro. La evolución es crucial en un entorno en el que la tecnología cambia rápidamente y las expectativas de los clientes son cada vez más altas. Como resultado, las empresas que adoptan y aprovechan eficazmente las capacidades de un TMS se distinguen claramente como líderes en la industria, como ejemplificado por empresas como Amazon, Wal-Mart, Mercado Libre, Castores, etc.

Las plataformas de TMS más avanzadas no solo han mejorado la visibilidad y la gestión de las operaciones de transporte, sino que han desencadenado una optimización profunda en los procesos internos de las organizaciones. Al permitir la manipulación de volúmenes masivos de datos en tiempo real, las plataformas optimizan los envíos diarios y generan una transformación en la cadena de valor empresarial.

### 3.1.- Sistemas TMS comerciales.

Algunos de los sistemas de gestión de autotransporte más importantes en la actualidad son los siguientes: Logistaas, Dispatch Science, Kuebix y Mercury Gate.

La siguiente tabla (Tabla 2) presenta sus características principales como referencia general, las cuales se explican con mayor detalle en los apartados posteriores.

	<b>Logistaas</b>	<b>Dispatch Science</b>	<b>Kuebix</b>	<b>MercuryGate</b>
<b>Ámbito de aplicación</b>	Gestión logística para tránsito terrestre, marítimo y aéreo.	Entregas de última, media y primera milla con soporte de IA.	Transporte multimodal adaptable a empresas de todos los tamaños.	Gestión de transporte multimodal centralizada (camión, aire, mar, ferrocarril, etc.).

<p><b>Principales características</b></p>	<ul style="list-style-type: none"> <li>• CRM especializado para clientes y proveedores.</li> <li>• Administración de tarifas, gestión de oportunidades de venta.</li> <li>• Seguimiento de envíos multimodales.</li> <li>• Consolidación de cargas.</li> <li>• Automatización de documentos y flujos de trabajo.</li> </ul>	<ul style="list-style-type: none"> <li>• Automatización de pedidos.</li> <li>• Optimización de rutas por IA.</li> <li>• Seguimiento en tiempo real.</li> <li>• Portal de autoservicio para clientes.</li> <li>• Facturación automatizada.</li> <li>• Paneles de análisis en vivo.</li> <li>• Cumplimiento de estándares de seguridad (AES-256, ISO 27001, HIPAA, SOC).</li> </ul>	<ul style="list-style-type: none"> <li>• Comparación de tarifas negociadas en tiempo real.</li> <li>• Reservas directas con transportistas.</li> <li>• Seguimiento de envíos.</li> <li>• Auditoría automática de facturas.</li> <li>• Informes analíticos, gestión financiera y de reclamaciones.</li> <li>• Análisis predictivo de gastos y desempeño.</li> </ul>	<ul style="list-style-type: none"> <li>• Plataforma unificada para todos los modos de transporte.</li> <li>• Visibilidad integral de la cadena logística.</li> <li>• Planificación estratégica.</li> <li>• Coordinación entre actores.</li> <li>• Escalabilidad global.</li> <li>• Herramientas de análisis y control de productividad.</li> </ul>
<p><b>Capacidades de integración</b></p>	<ul style="list-style-type: none"> <li>• Integración con aerolíneas y navieras.</li> <li>• Generación automática de documentos.</li> <li>• Importación/exportación electrónica de datos</li> </ul>	<ul style="list-style-type: none"> <li>• Integración vía interfaz de programación de aplicaciones (API por sus siglas en inglés) con otros sistemas de gestión</li> <li>• Comunicación directa entre clientes, conductores y despachadores.</li> </ul>	<ul style="list-style-type: none"> <li>• Integración con sistemas de planificación de recursos empresariales (ERP por sus siglas en inglés), comercio electrónico y sistemas externos mediante API abiertas e Intercambio Electrónico de Datos (EDI en inglés).</li> <li>• Integración de pedidos y facturas.</li> </ul>	<ul style="list-style-type: none"> <li>• Integración con múltiples actores logísticos en una sola plataforma.</li> <li>• Gestión nativa de transporte multimodal.</li> </ul>

<b>Escalabilidad</b>	Empresas globales con operaciones multimodales.	Empresas de mensajería, retail y distribución.	PYMEs y grandes corporativos con cadenas de suministro.	Multinacionales y operadores logísticos a gran escala.
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>• Centralizar operaciones logísticas.</li> <li>• Reducir carga manual.</li> <li>• Mejorar eficiencia en ventas y operaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Optimizar rutas y tiempos de entrega.</li> <li>• Aumentar seguridad y transparencia.</li> <li>• Mejorar experiencia del cliente.</li> </ul>	<ul style="list-style-type: none"> <li>• Reducir costos de transporte.</li> <li>• Optimizar relaciones con transportistas.</li> <li>• Integrar procesos financieros y operativos.</li> </ul>	<ul style="list-style-type: none"> <li>• Mejorar coordinación en la cadena de suministro.</li> <li>• Facilitar decisiones estratégicas.</li> <li>• Garantizar continuidad operativa.</li> </ul>

### 3.1.1.- Logistaas.

Logistaas (Imagen 4) es uno de los principales sistemas administrativos de carga con una presencia mundial, dentro de aerolíneas, cargueros marinos y transporte terrestre, al contar con una huella tan grande en el mercado se seleccionó como un ejemplo a seguir para el TMS a desarrollar.



Imagen 4: Logo de Logistaas.

En su página web menciona que “Desde su inicio en 2014, Logistaas ha revolucionado la gestión logística a través de su avanzada plataforma en la nube. Diseñada para agentes de carga, consolidadores (NVOCC) y líneas de envío, esta solución abarca el ciclo económico completo, mejorando desde el primer contacto con un cliente hasta el cierre del archivo de trabajo. En más de 30 países, cientos de miles de envíos son procesados anualmente en Logistaas. Su flexibilidad, eficiencia y enfoque colaborativo han hecho de esta plataforma

una piedra angular en la evolución de la cadena de suministro global” (Loogistas, 2023). Algunas de sus características principales son:

- **Gestión de ventas y precios**

- CRM(Gestión de Relación con los Clientes) especializado para transitarios, NVOCC (Non-Vessel Operating Common Carrier) y navieras.
- Clasificación de contactos, gestión de direcciones y asignación de administradores de cuentas.
- Creación y seguimiento de oportunidades, cotizaciones y actividades de ventas.
- Análisis de clientes inactivos y desempeño de usuarios.
- Gestión y carga de tarifas, diferenciación de precios y solicitudes a proveedores.

- **Manejo de operaciones**

- Administración de envíos multimodales y despachos aduaneros con integración a aerolíneas y navieras.
- Gestión de flujo de trabajo con paneles personalizados, automatización de tareas y seguimiento de envíos.
- Conexión con transportistas mediante reservas con más de 50 transportistas y envíos de datos a aerolíneas.
- Reducción de trabajo manual haciendo uso de la generación de documentos automáticos, importación de datos y transferencia electrónica de información.
- Consolidación de envíos permitiendo una agrupación, movimiento y división de costos.

### **3.1.2.- Dispatch Science.**

Dispatch Science  es una innovadora plataforma de gestión de entregas diseñada para optimizar y automatizar los procesos logísticos de empresas de mensajería. Gracias a su uso de inteligencia artificial, permite una planificación eficiente de rutas, seguimiento en tiempo real y una integración fluida con otros sistemas a través de su interfaz de programación de aplicaciones (API por sus siglas en inglés). Sus características de IA la hacen destacar de entre todos los TMS disponibles en el mercado actualmente.

“Dispatch Science es un sistema de gestión de entregas disruptivo para remitentes, transportistas y mensajeros. Simplificamos y automatizamos el ingreso de pedidos, el

despacho, el enrutamiento, el transporte, el seguimiento en vivo, la facturación y los informes. Nuestras capacidades se extienden más allá de la última milla para incluir también la gestión del transporte de media y primera milla y la visibilidad de la cadena de custodia.” (Dispatch Science, 2023)



*Imagen 5: Logo de Dispatch Science.*

Algunas de sus principales características son:

- **Gestión integral:** Maneja todo el proceso, desde cotización hasta pago, optimizando cada punto de contacto con el cliente.
- **Seguimiento en tiempo real:** Portal de autoservicio para clientes, alertas vía SMS, correo o llamadas, y comunicación directa entre despachadores, conductores y clientes.
- **Optimización de entregas:** Asigna pedidos según servicio, vehículo y ubicación, optimiza rutas para mayor eficiencia y permite recopilación de firmas, pagos y comprobantes de entrega.
- **Transparencia y control:** Paneles de análisis en tiempo real, acceso de conductores a comisiones y generación automatizada de facturación con reglas flexibles.
- **Seguridad:** Cumple certificaciones de seguridad como AES-256, ISO 27001, HIPAA, FedRAMP y SOC 1/2.

### 3.1.3.- Kuebix

Kuebix (Imagen 6) ofrece un TMS con inteligencia de fletes que permite a las empresas capitalizar las oportunidades de la cadena de suministro a través de la visibilidad, el control y el uso del análisis predictivo.



*Imagen 6: Logo de Kuebix.*

“Kuebix TMS es una solución modular que se adapta para satisfacer las necesidades únicas de cada cadena de suministro con una amplia gama de aplicaciones, integraciones y programas de servicios gestionados de primer nivel. Basado en la última tecnología en la nube, Kuebix ofrece implementaciones rápidas y un bajo coste total de propiedad.” (Kuebix Transportation Management System, 2023)

Algunas de sus características importantes:

- **Tarifas negociadas y cotizaciones TME lado a lado:** Conecta automáticamente operadores permitiendo comparar tarifas para elegir la mejor opción, ahorrando tiempo y dinero.
- **Servicio puntual:** Reduce costos solicitando cotizaciones al contado para fletes de volumen, aprovechando una amplia comunidad naviera para obtener tarifas competitivas.
- **Libro:** Facilita la reserva de envíos directamente con transportistas, agilizando el proceso como si se reservara un vuelo en línea.
- **Pista:** Ofrece visibilidad en tiempo real a todas las partes interesadas, mejorando la gestión de la carga y el servicio al cliente.
- **Informes y análisis:** Genera informes y paneles detallados para optimizar el gasto en flete, mejorar relaciones con transportistas y tomar decisiones más inteligentes.
- **Pago de flete y auditoría:** Audita automáticamente facturas, identifica excepciones y aprueba pagos, optimizando el flujo de efectivo y reduciendo errores.

- **Gestión financiera:** Permite agregar códigos GL a facturas para un análisis preciso de costos por producto.
- **Gestión de reclamaciones:** Administra reclamos y mantiene un historial detallado para garantizar que no se pasen por alto.
- **Prepagado y agregado:** Añade recargos a tarifas negociadas para cubrir costos adicionales, mejorando el margen de beneficio.

## **CAPACIDADES DE INTEGRACIÓN**

- **Integración de pedidos:** Facilite la automatización del ingreso de envíos al completar automáticamente los pedidos por artículo de línea directamente desde un sistema ERP a Kuebix y luego devolver los detalles del envío al sistema ERP para administrar el ciclo de vida de la información del pedido. Significando que se ahorra tiempo al no tener que volver a ingresar las órdenes de compra y elimina los errores del usuario que pueden provocar una solución de problemas tediosa y entregas incorrectas.
- **Integración de facturas:** Kuebix TMS audita automáticamente las facturas de los transportistas con respecto a la tarifa acordada para el envío y luego se crean reclamos de excepción de tarifas en caso de discrepancias. Ahorre tiempo mirando únicamente las facturas incorrectas y administre el flujo de caja pagando únicamente las facturas cuando vencen.
- **Integraciones personalizadas:** Integra con cualquier sistema de comercio electrónico, ERP(Enterprise Resource Planning), cotización o almacenamiento de terceros que utilice EDI(Electronic Data Interchange), API(Application Programming Interface) abiertas u otros protocolos estándar.
- **API abiertas:** Utiliza las API abiertas de Kuebix para agregar seguimiento de envíos y la capacidad de calificar, reservar y programar la entrega directamente desde el sitio web de comercio electrónico o el sistema de cotizaciones de la empresa. Brindando a los clientes una experiencia perfecta en la que pueden administrar sus pedidos y entregas directamente desde un sitio web, lo que aumenta la probabilidad de ventas adicionales o repetidas.

### 3.1.4.- MercuryGate.

“MercuryGate despliega soluciones de gestión de transporte multimodal que han demostrado ser una ventaja competitiva para una variedad de actores en la industria logística, incluyendo transportistas, agentes de carga y operadores logísticos a nivel global. La singularidad de las soluciones de MercuryGate radica en su capacidad inherente para gestionar de forma nativa todos los modos de transporte en una única plataforma integral. Desde la administración de paquetería, carga consolidada, FTL(carga completa de camión) y LTL(carga parcia) hasta servicios aéreos, marítimos, ferroviarios e intermodales, MercuryGate abarca un espectro completo de opciones. “ (MercuryGate International, 2021)

El valor que MercuryGate(Imagen 7) aporta a los usuarios de su Sistema de Gestión de Transporte (TMS) se manifiesta en una mayor productividad y eficiencia operativa. La capacidad de abordar diversos modos de transporte en un entorno unificado permite una visión integrada de las operaciones logísticas, facilitando la toma de decisiones informadas y mejorando la coordinación en toda la cadena de suministro. Esta integración completa se traduce en una ejecución más fluida de las operaciones, dotando a los usuarios de herramientas poderosas para enfrentar con éxito los desafíos logísticos en un entorno de mercado dinámico y altamente competitivo.



*Imagen 7: Logo de MercuryGate.*

Sistemas como los mencionados anteriormente forman una base y guía para el proyecto en cuanto a las funciones planteadas a desarrollar como lo es la facturación, rastreo de unidades, visualización de estadísticas mediante un dashboard, etc. Aunque también se busca innovar en ciertos aspectos como lo es en un panel de información accesible al cliente, donde generalmente otros sistemas no lo brindan y generar un entorno no tan restringido para los

operadores de unidades, debido a la tendencia a cumplir con demandas de tiempo para lograr las cuotas establecidas por los sistemas administrativos, pero haciendo la implementación efectiva de los componentes dentro de una plataforma de TMS permite a las organizaciones operar de manera más inteligente, eficiente y estratégica en un entorno de cadena de suministro dinámico y altamente competitivo.

## Capítulo 4.- Metodología.

Para el desarrollo del TMS se requiere una metodología que garantice un proceso ordenado, dada la complejidad de sus requerimientos. El enfoque debe permitir una definición clara de requisitos desde el inicio, con etapas bien delimitadas para asegurar que cada componente cumpla su función antes de avanzar al siguiente, garantizando así un sistema robusto y coherente. Para esto se analizaron diversas posibles metodologías de ingeniería de software cada una ofreciendo distintos enfoques que varían en estructura, nivel de iteración, tiempos de entrega, etc.

Entre estas alternativas, destaca en primer lugar el modelo en cascada, el cual constituye una de las metodologías más tradicionales dentro del desarrollo de software. Su enfoque secuencial facilita el avance ordenado desde la recopilación de requisitos hasta su despliegue, siendo especialmente apropiado para entornos en los que los requerimientos se encuentran bien definidos desde el inicio y las probabilidades de cambios son mínimas.

Como una evolución estructurada de este enfoque, el modelo en V introduce una correlación directa entre las fases de desarrollo y sus respectivas etapas de verificación y validación. Este modelo enfatiza la calidad mediante mecanismos formales de prueba que aseguran la revisión sistemática del producto en cada etapa del ciclo de vida.

Ante las limitaciones de los modelos estrictamente secuenciales, el enfoque iterativo plantea una solución flexible que permite el desarrollo del sistema a través de repeticiones sucesivas. Cada iteración integra análisis, diseño, codificación y pruebas, lo que posibilita una retroalimentación continua con una adaptación temprana a cambios en los requerimientos.

Entre los enfoques ágiles, destaca Scrum, cuya organización en sprints, definición clara de roles y entregas incrementales permite una adaptación rápida al cambio y una colaboración

constante con el cliente. Su estructura fomenta la eficiencia en equipos orientados a resultados en contextos dinámicos.

En esta misma línea de agilidad, Kanban proporciona una metodología visual que optimiza el flujo de trabajo mediante tableros y una gestión eficiente del trabajo en curso. Al no requerir iteraciones fijas, permite una organización continua y flexible de tareas.

Extreme Programming (XP), por su parte, se centra en mejorar la calidad del software junto con su capacidad de respuesta al cambio mediante prácticas como la programación en pareja, la integración continua y el desarrollo guiado por pruebas. Este enfoque busca mantener un diseño simple ajustándose a los requerimientos cambiantes.

El uso del prototipado se presenta como una estrategia adecuada cuando los requisitos no están completamente definidos, ya que permite construir versiones preliminares del sistema para facilitar la validación temprana con los usuarios y ajustar las funcionalidades de acuerdo con sus expectativas.

De forma similar, el Desarrollo Rápido de Aplicaciones (RAD) prioriza la entrega acelerada de software funcional a través de herramientas visuales y una interacción constante con el usuario final. Esta metodología reduce los tiempos de desarrollo sin comprometer la alineación con las necesidades emergentes. Por último, el enfoque DevOps propone la integración de los procesos de desarrollo y operación en una sola estrategia continua. La automatización, la colaboración interdisciplinaria y la entrega continua permiten una implementación ágil, con mayor confiabilidad y una respuesta oportuna frente a incidentes o cambios operativos.

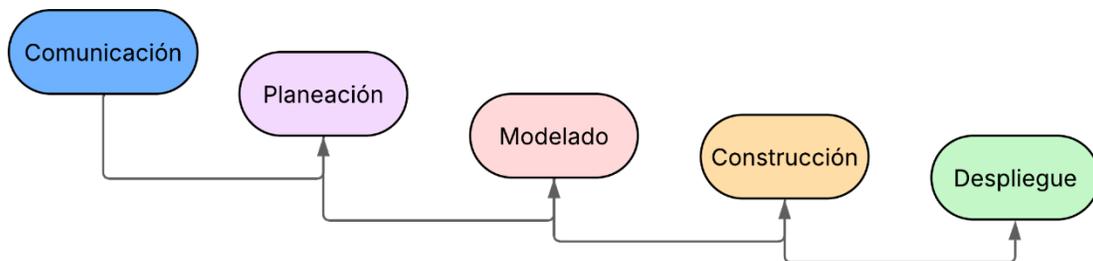
Las características descritas han sido ampliamente abordadas en obras como las de Pressman, Sommerville y Schach. Gracias a sus aportes, fue posible comprender las distintas metodologías de forma clara. A continuación, se presenta una tabla (Tabla 3) que resume los principales aspectos de cada metodología permitiendo una mejor comprensión de sus características.

<b>Modelo</b>	<b>Enfoque</b>	<b>Flexibilidad</b>	<b>Documentación</b>	<b>Adaptabilidad</b>	<b>Entrega</b>	<b>Mejor Uso</b>
<b>Cascada</b>	Secuencial	Baja	Alta	Baja	Final	Requisitos fijos y claros
<b>Modelo V</b>	Validación Paralela	Baja	Muy Alta	Baja	Final	Sistemas críticos
<b>Iterativo</b>	Incremental	Alta	Media	Media	Frecuente	Requisitos parcialmente definidos
<b>Scrum</b>	Ágil	Muy alta	Baja	Muy Alta	Cada sprint	Proyectos dinámicos con cambios frecuentes
<b>Kanban</b>	Flujo continuo	Alta	Mínima	Alta	Continúa	Mantenimiento y mejoras progresivas
<b>XP</b>	Ágil	Muy Alta	Baja	Muy alta	Continúa	Proyectos pequeños con alta calidad requerida
<b>Proto-tipado</b>	Pruebas tempranas	Alta	Baja	Alta	Rápida	Requisitos poco claros o innovadores
<b>RAD</b>	Desarrollo rápido	Alta	Media	Alta	Rápida	Proyectos urgentes con cliente disponible
<b>DevOps</b>	Automatización CI/CD	Alta	Técnica	Alta	Continúa	Entornos cloud y entrega frecuente

Tabla 2: Tabla comparativa de metodologías

Habiendo considerado cada aspecto, se optó por la metodología de cascada debido a sus características específicas que alinean con las principales necesidades del proyecto. La metodología de cascada ofrece una definición detallada desde el inicio, así como un desarrollo lineal y estructurado. Los aspectos proporcionan una mayor claridad y control en cada fase del proyecto. Además, su enfoque en la documentación detallada asegura un seguimiento riguroso y una fácil referencia a lo largo del proceso. En el capítulo se explica en detalle la justificación de la elección y se describe cómo se implementará la metodología de cascada en las distintas etapas del proyecto: recolección de requisitos, diseño, construcción, verificación y pruebas.

“El modelo de la cascada, a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado” (Roger S. Pressman, 2020)(Imagen 8)



*Imagen 8: Metodología de cascada acorde a Roger S. Pressman*

En el ámbito de los sistemas administrativos, especialmente al tratar con sistemas complejos como lo es un TMS, la selección de una metodología adecuada es esencial para garantizar el éxito del proyecto. La metodología de desarrollo de software de cascada se presentó como una serie de pasos ideal para desarrollar el proyecto debido a su estructura lineal de trabajo. A continuación, se describen en detalle las razones y el enfoque dentro de la aplicación:

#### **4.1.- Justificación de la Metodología de Cascada.**

Al abordar sistemas complejos, como un TMS, la comprensión exhaustiva de los requisitos iniciales se convierte en la base para el éxito del proyecto. Así, la metodología de cascada se

presenta como un enfoque que busca proporcionar una estructura robusta, enfocándose en la completa definición de requisitos en las fases iniciales del desarrollo. Apuntando a reducir la posibilidad de ambigüedades y cambios drásticos en etapas posteriores, estableciendo así las bases sólidas necesarias para enfrentar los desafíos inherentes a proyectos de esta envergadura.

La metodología de cascada aborda la complejidad de un TMS a través de una cuidadosa planificación y ejecución secuencial. Los requisitos iniciales, obtenidos a través de un diálogo profundo con todas las partes interesadas, se convierten en el mapa detallado que guía cada paso del proceso de desarrollo. La fase inicial se destaca por su búsqueda de claridad y exhaustividad, evitando ambigüedades y estableciendo una visión clara que servirá como referencia en las etapas subsiguientes.

## **4.2.- Aplicación Estructurada de la Metodología de Cascada.**

Es necesario brindar una visión general de cómo se aplicó la metodología en cascada dentro de proyecto, destacando su estructura secuencial y su impacto en el desarrollo del mismo, por lo cual a continuación se brindarán las generalidades de este proceso basado en el autor Roger S. Pressman en su libro “Ingeniería del Software” (Roger S. Pressman, 2020)

### **4.2.1.- Comunicación.**

La fase de comunicación comprende el análisis de requerimientos, cuyo propósito fundamental es determinar y documentar las necesidades del sistema. Este proceso implica la recolección sistemática de información mediante diversas técnicas como entrevistas estructuradas, cuestionarios especializados y sesiones de trabajo colaborativo con los usuarios clave, entre los que se incluyen, administradores de flota y personal técnico especializado.

A partir de los datos obtenidos, se elaboró un documento de especificaciones técnicas detallado que define los comportamientos esperados del sistema. Este documento incorpora funcionalidades esenciales como la optimización de rutas, la gestión inteligente de entregas, el monitoreo en tiempo real de unidades de transporte y la generación automatizada de informes operativos.

#### **4.2.2.- Planeación.**

Habiendo identificado los requerimientos, se desarrollan escenarios basados en los requerimientos previamente definidos, los cuales permiten validar funcionalidades con usuarios finales asegurando una visión compartida entre los involucrados. Estos modelos se representan mediante descripciones detalladas de procesos y diagramas que ilustran la interacción entre los diferentes actores del sistema y las capacidades principales de la plataforma.

Durante esta fase, también se definen los recursos necesarios, los tiempos estimados y los riesgos asociados al proyecto. Esta planificación contribuye a establecer una base sólida sobre la cual se desarrollaron las siguientes etapas.

#### **4.2.3.- Modelado.**

Para el diseño del sistema, se emplean herramientas de modelado estructurado basadas en el lenguaje UML (Unified Modeling Language). Estos modelos gráficos permiten representar tanto la arquitectura estática como el comportamiento dinámico del sistema. Los diagramas de casos de uso describen las interacciones funcionales entre los actores y el sistema, mientras que los diagramas de clases definen la estructura interna mediante entidades fundamentales como Ruta, Viaje, Pedido, Operador, Cliente y Usuario, detallando sus relaciones estructurales.

Complementariamente, se utilizaron diagramas de secuencia para modelar el flujo de procesos operativos clave, como la asignación de entregas o el seguimiento de unidades. Estas especificaciones de diseño constituyen la base técnica para la posterior implementación del sistema, asegurando una transición fluida entre las fases de modelado y construcción.

#### **4.2.4 Construcción**

Durante la fase de construcción, el TMS es desarrollado utilizando tecnologías de código abierto que permiten una solución web escalable y eficiente. La capa de lógica de negocio se desarrolla en Python, seleccionado por su capacidad para manejar operaciones complejas de manera eficiente. El framework Flask proporciona la estructura base de la aplicación, permitiendo controlar rutas, manejar solicitudes del usuario y conectar la lógica del negocio con la interfaz.

Los datos se gestionan con MySQL, garantizando seguridad y organización de la información clave. La interfaz usa tecnologías web básicas: HTML, CSS y JavaScript creando una plataforma fácil de usar desde cualquier dispositivo. Esta solución tecnológica forma un sistema estable que se adapta a las necesidades logísticas. Se discutirá más a fondo el uso de estas herramientas durante el CAPÍTULO 6.- DESARROLLO DE LA APLICACIÓN.

#### **4.2.5.- Despliegue.**

Finalmente, la fase de despliegue corresponde a la entrega del sistema al cliente y su puesta en funcionamiento en un entorno real. Sin embargo, el TMS, actualmente se encuentra en fase de desarrollo, siendo implementado parcialmente en un entorno de pruebas controlado. Este despliegue inicial ha permitido verificar su correcto funcionamiento, identificar posibles fallos y validar el cumplimiento de los requisitos establecidos. Las evaluaciones se han llevado a cabo en condiciones que emulan escenarios reales de operación, con especial énfasis en las funcionalidades centrales como administración de rutas, distribución de entregas y control de flota vehicular.

## **Capítulo 5.- Planeación y Modelado.**

Para lograr la creación del TMS, fue necesario llevar a cabo una etapa integral de análisis y diseño, la cual permitió sentar las bases conceptuales y estructurales del sistema antes de su desarrollo. Esta fase se llevó a cabo de acuerdo con la metodología seleccionada, abarcando las etapas de comunicación, planeación y modelado. Durante la comunicación, se recopilieron los requerimientos del sistema a partir de las necesidades de los usuarios clave. Posteriormente, en la planeación, se definieron los alcances, restricciones y prioridades del proyecto, lo que permitió establecer una dirección clara para su desarrollo. Finalmente, en la fase de modelado, se representaron los procesos y estructuras del sistema mediante diversos diagramas, permitiendo visualizar las funcionalidades esperadas, los actores involucrados, las secuencias de interacción de las clases que compondrán la solución. Este proceso fue esencial para asegurar que el sistema responda adecuadamente a los requerimientos funcionales identificados.

### **5.1.- Casos de uso.**

Mediante el proceso de comunicación con la empresa CANACE, se logró identificar y los distintos escenarios operativos que el sistema debía cubrir, destacando que gran parte de sus procesos operativos eran gestionados manualmente, principalmente mediante hojas de cálculo en Excel junto con plataformas externas de paquetería. Esta dinámica generaba duplicidad de tareas, falta de control en los registros y dificultades para dar seguimiento en tiempo real. A partir de estas observaciones y gracias a una serie de entrevistas con los distintos actores involucrados —administradores, operadores y clientes— se logró establecer un conjunto claro de necesidades que el sistema debía cubrir.

El rol de Administrador concentra las funciones de gestión del sistema, abarcando la administración de usuarios, el control de viajes (creación, modificación y eliminación), la supervisión de unidades y los procesos de facturación. Estas capacidades permiten mantener un control centralizado de todas las operaciones críticas.

El Operador, puede consultar sus viajes asignados, acceder a información de nómina y registrar cambios en sus viajes actualizando su ubicación, permitiendo el monitoreo en tiempo real.

Los Clientes cuentan con la habilidad de obtener las cartas porte, descarga de facturas y seguimiento de los envíos. Estas funciones están diseñadas para proveer un mejor servicio al cliente.

Para organizar claramente las necesidades de cada rol, se creó un diagrama de casos de uso (Imagen 9), permitiendo representar de forma estructurada las interacciones de los diferentes roles del sistema.

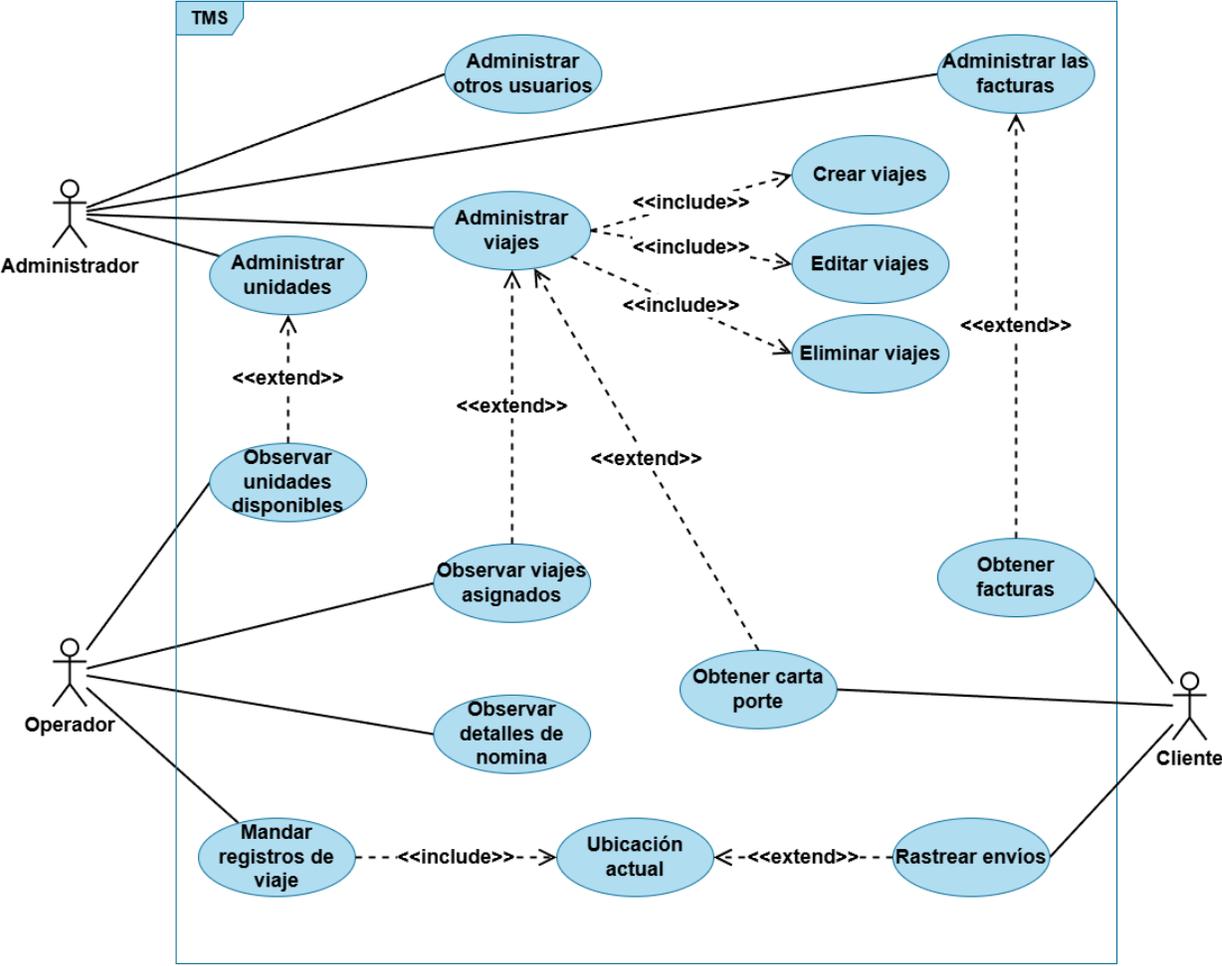


Imagen 9: Diagrama de Casos de Uso.

Para explicar más a detalle los casos de uso, se siguió el formato propuesto por Roger S. Pressman, como se puede observar a continuación en el caso de “Administrar viajes”. (Tabla 4)

<b>Elemento</b>	<b>Descripción</b>
<b>Caso de uso</b>	Administrar viajes
<b>Actor</b>	Administrador
<b>Descripción</b>	Permite al administrador gestionar los viajes: crearlos, editarlos y eliminarlos.
<b>Precondiciones</b>	El administrador debe haber iniciado sesión correctamente.
<b>Postcondiciones</b>	Se actualiza el sistema con los cambios en los registros de viaje.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El administrador accede al módulo de viajes.</li> <li>2. Selecciona la opción deseada (crear, editar, eliminar).</li> <li>3. El sistema procesa la acción y actualiza la información.</li> </ol>
<b>Flujos alternativos</b>	<ul style="list-style-type: none"> <li>-Si el viaje no existe, muestra mensaje de error.</li> <li>- Si los datos están incompletos, solicita corrección.</li> </ul>
<b>Inclusiones</b>	<ul style="list-style-type: none"> <li>-Crear viajes</li> <li>- Editar viajes</li> <li>- Eliminar viajes</li> </ul>
<b>Extensiones</b>	Obtener carta porte (en caso de viaje creado)

*Tabla 3: Formato de Caso de Uso "Administrar Viajes".*

Para el caso de uso “Mandar registros” con el actor operador el formato generado es el siguiente (Tabla 5):

<b>Elemento</b>	<b>Descripción</b>
<b>Caso de uso</b>	Mandar registros de viaje
<b>Actor</b>	Operador
<b>Descripción</b>	El operador envía los registros del viaje realizado para mantener informada a la administración.
<b>Precondiciones</b>	El operador debe estar autenticado y tener asignado un viaje.
<b>Postcondiciones</b>	El sistema guarda los registros enviados por el operador.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El operador accede a su panel.</li> <li>2. Selecciona el viaje asignado.</li> <li>3. Llena y envía el formulario con los detalles del viaje.</li> </ol>
<b>Flujos alternativos</b>	Si los datos están incompletos, solicita corrección.
<b>Inclusiones</b>	Ubicación actual
<b>Extensiones</b>	Ninguna

*Tabla 4: Formato de Caso de Uso "Mandar registros".*

Como último ejemplo usando al actor de cliente con el caso de uso “Rastrear envíos” es explicado de la siguiente manera (Tabla 6):

Elemento	Descripción
Caso de uso	Rastrear envíos
Actor	Cliente
Descripción	El cliente puede rastrear la ubicación de sus envíos en tiempo real.
Precondiciones	El cliente debe tener acceso a la plataforma y un envío activo.
Postcondiciones	Se muestra la ubicación actual del envío.
Flujo de eventos	<ol style="list-style-type: none"> <li>1. El cliente inicia sesión.</li> <li>2. Accede al módulo de rastreo.</li> <li>3. El sistema muestra la ubicación y el estado actual del envío.</li> </ol>
Flujos alternativos	Si el envío aún no está activo, muestra mensaje informativo.
Inclusiones	Ninguna
Extensiones	Ubicación actual

Tabla 5: Formato de Caso de Uso "Rastrear envíos".

## 5.2.- Diagrama de clases.

Tras analizar cómo trabajaba la empresa y definir sus necesidades con casos de uso, se creó un modelo de clases (Imagen 10): que convierte esos requerimientos en una estructura técnica clara. Este diagrama organiza los componentes clave del sistema (operadores, clientes, camiones, viajes, etc.) y cómo se relacionan entre sí, sentando las bases para desarrollar el TMS.

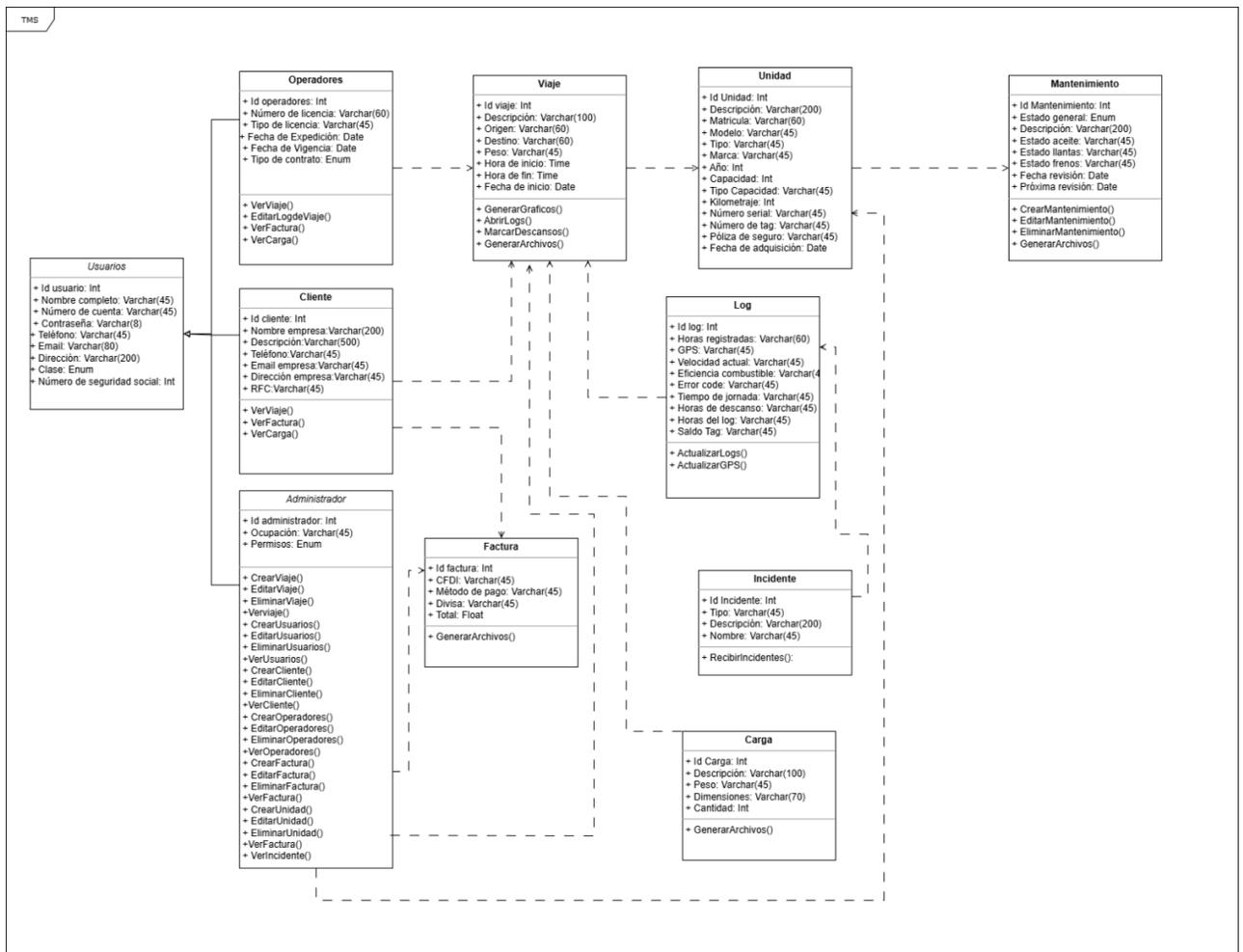


Imagen 10: Diagrama de clases.

Basándose en los actores de los casos de uso se identificaron tres tipos de usuarios:

- **Administradores:** Son los usuarios con máximo nivel de control sobre el sistema. Tienen permisos para crear, editar, eliminar o consultar las demás áreas del sistema
- **Clientes:** Usuarios que representan a las personas o empresas que solicitan los servicios. El sistema almacena su información relevante como su RFC, nombre de la empresa, teléfono, etc. A través del sistema, pueden obtener toda la información relevante acerca de sus envíos como lo es facturas, ubicación actual o su carta porte.
- **Operadores:** realizan los viajes. Para ellos, el sistema guarda datos como su licencia, tipo de contrato y fechas importantes. Además, se pensó en que puedan ver sus viajes asignados, consultar detalles de su nómina y reportar en tiempo real su ubicación y actividades a través de bitácoras (logs).

Respecto a las principales áreas del sistema:

- **Viaje:** Aquí se almacena todo lo que tiene que ver con los traslados, como los puntos de origen-destino, fechas, horarios, permitiendo generar la documentación necesaria como lo es la carta porte.
- **Unidad:** Registra toda la información de los vehículos (marca, modelo, placas, capacidad, etc.), conectándose tanto con los viajes como con las actividades de mantenimiento.
- **Carga:** Es el detalle de lo que se transporta incluyendo la descripción, peso, dimensiones, cantidad y precio, con opción de generar documentos asociados.
- **Factura:** Guarda la información relacionada a la cobranza de los servicios, incluyendo los datos fiscales del cliente y el monto.
- **Log:** Permite registrar eventos importantes durante el viaje, como el avance, paradas, consumo de combustible o cualquier incidente técnico. También actualiza la ubicación GPS de los operadores.
- **Mantenimiento:** Lleva el control del estado de los vehículos, tanto revisiones preventivas como correctivas.
- **Incidente:** Aquí se documentan los eventos imprevistos que puedan ocurrir en la operación, como accidentes o fallos de la unidad.

### 5.3.- Diagrama de secuencia.

Una vez establecida la arquitectura del sistema mediante el diagrama de clases, el siguiente paso fue diseñar el flujo de operaciones usando diagramas de secuencia (Imagen 11). Este diagrama describe cómo los diferentes módulos del sistema colaboran para llevar a cabo las operaciones esenciales en respuesta a las acciones de los usuarios.

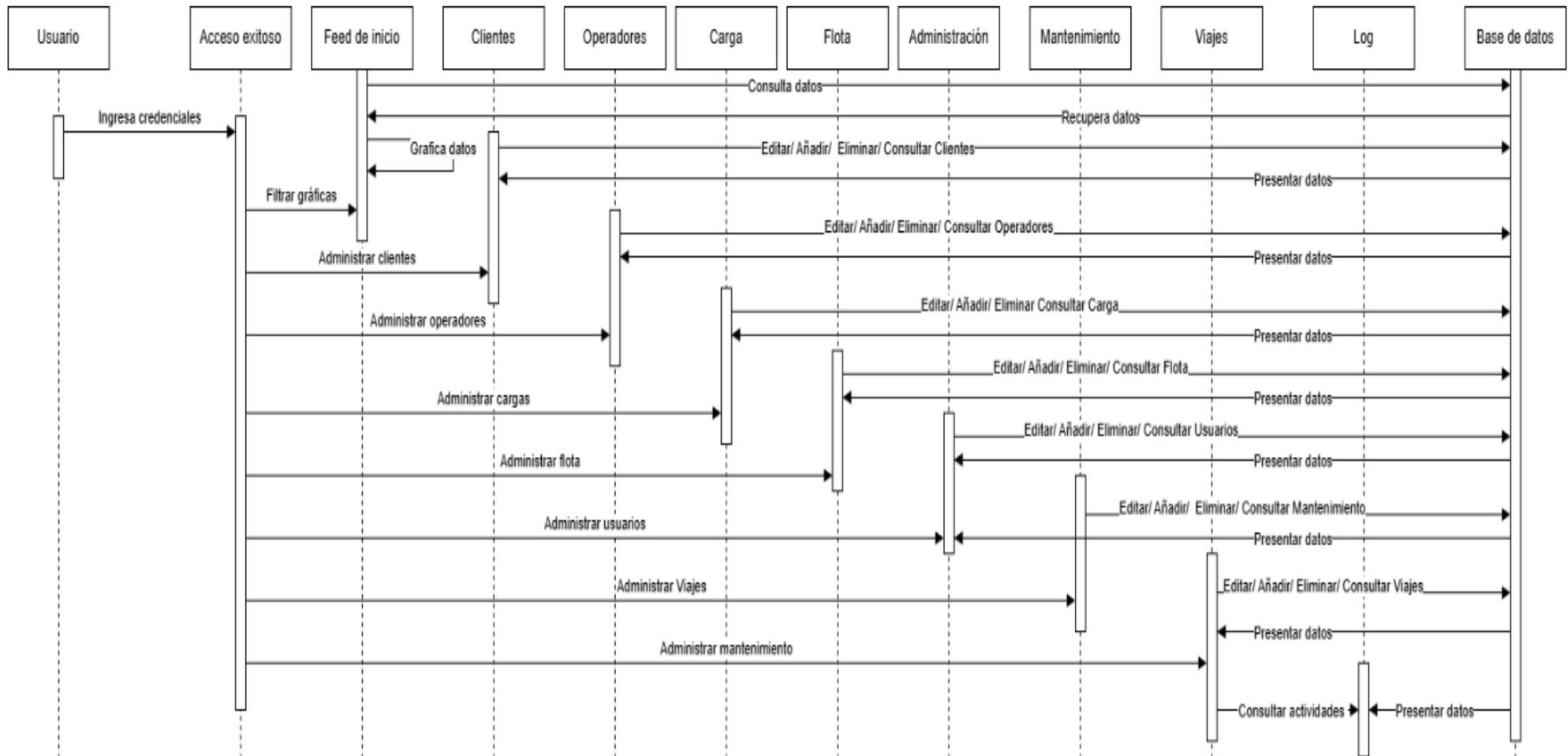


Imagen 11: Diagrama de secuencia.

El proceso comienza cuando el usuario ingresa sus credenciales para acceder al sistema. Tras validar esta información, el acceso es concedido y el usuario es redirigido a la página de inicio, donde se presentan gráficas que resumen información operacional. Desde este punto, el usuario puede aplicar filtros a las gráficas o navegar hacia los distintos módulos disponibles, dependiendo de las necesidades específicas de gestión.

Desde la página de inicio el usuario puede observar y acceder a múltiples categorías relevantes para la administración de sistema como lo son:

- **Clientes:** Permite agregar, consultar, editar o eliminar información de los clientes registrados.
- **Operadores:** Facilita la administración de los datos de los operadores, incluyendo su actualización o eliminación.
- **Carga:** Se lleva el registro detallado de las mercancías transportadas, permitiendo agregar, eliminar, editar o consultar sus registros.
- **Flota:** Administra la información de las unidades vehiculares, así como sus actualizaciones y consultas.
- **Administración:** Encargada del control de acceso y permisos dentro del sistema.
- **Unidades:** Registra, consulta y gestiona las actividades de mantenimiento preventivo o correctivo de los vehículos.
- **Viajes:** Permite registrar, modificar y consultar viajes asignados a las unidades.
- **Log:** Brinda acceso al historial de eventos registrados durante los viajes.

Cada una de las categorías mencionadas está conectada directamente con la Base de datos, garantizando la recuperación y persistencia de la información en tiempo real, asegurando así la integridad del sistema. En este sentido, cada categoría fue diseñada teniendo en cuenta la posibilidad de que el usuario realice las operaciones básicas de consultar, crear, actualizar o eliminar. Esto asegura que los usuarios puedan interactuar con los datos según las necesidades específicas de cada categoría, manteniendo siempre la información actualizada y disponible.

## 5.4.- Diseño de la Base de Datos.

Con un entendimiento sólido de las necesidades de CANACE y de la estructura deseada para el sistema, se dio paso al diseño del modelo Relacional (Imagen 12) para la creación de la base de datos, el cual fue sometido al proceso de tercera normalización. Esta normalización permitió optimizar la estructura, eliminando redundancias y mejorando la integridad de los datos.

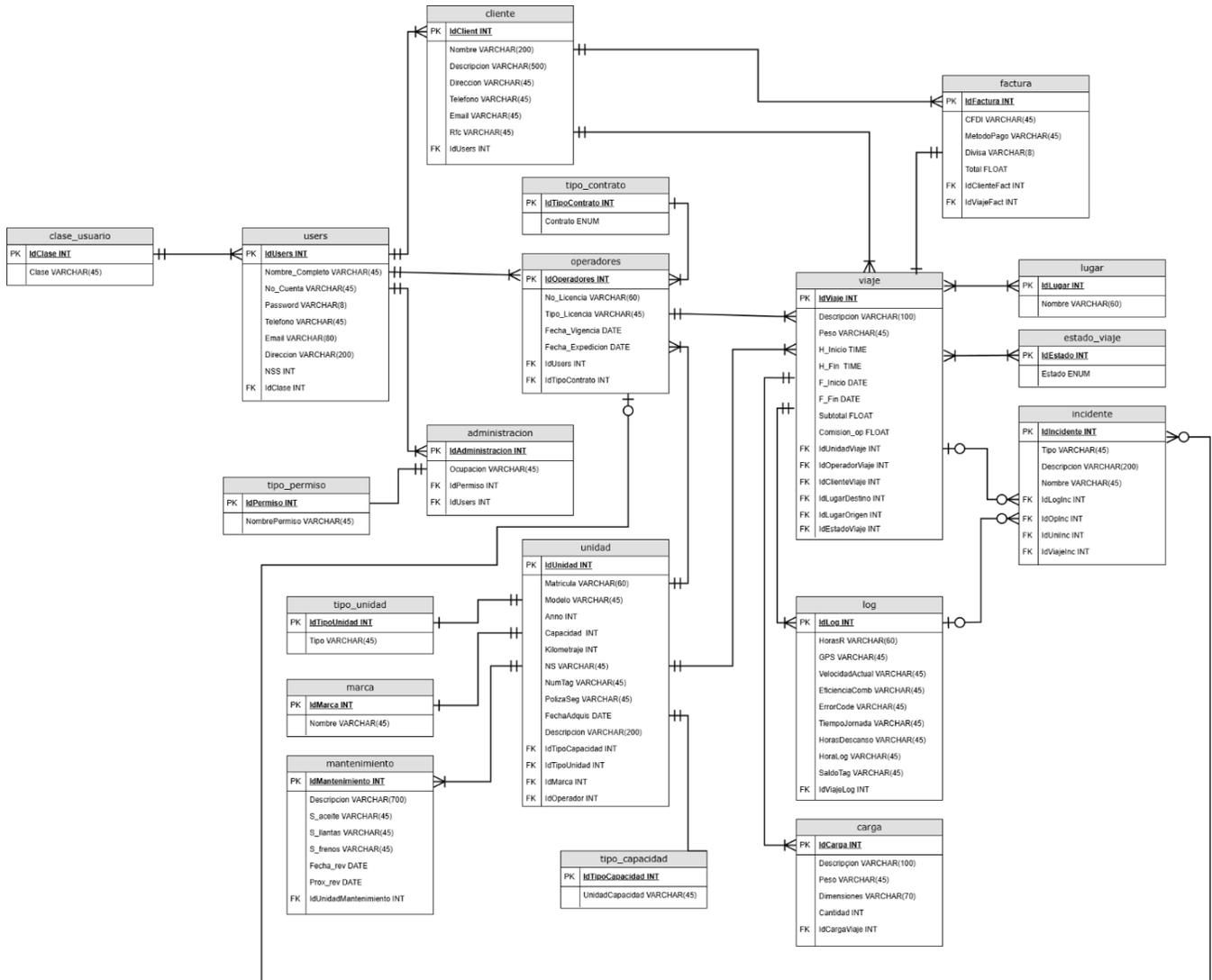


Imagen 12: Diagrama Relacional.

El modelo ER, al ser una representación tangible de las ideas propuestas, permitió a los directivos visualizar la dirección del proyecto y proporcionar comentarios valiosos. La

interacción activa durante esta fase no solo fortaleció el diseño, también sentó las bases para un proceso iterativo y colaborativo.

Habiendo diseñado el modelo Relacional se continuo con la creación de la base datos donde las principales tablas de la base de datos son:

- **Log:** Se encarga de registrar los eventos ocurridos en el transcurso del viaje, almacenando información como lo es la hora, la velocidad actual, las coordenadas de GPS actuales, etc. Permitiendo un historial completo del estado de la unidad a lo largo de su viaje.
- **Viaje:** Almacena información sobre cada viaje en específico; cumpliendo una función similar a una carta porta que permite donde se guarda de donde sale la unidad a la que fue asignada el viaje, a donde va, su costo, quien maneja la unidad, etc.
- **Carga:** Tabla encargada de detallar todas las características pertinentes de las cargas que las unidades llevan en cada viaje.
- **Operadores:** Se encarga de almacenar toda la información pertinente a los operadores de las unidades asignándole a ciertos usuarios el rol del mismo nombre que en una versión más completa del sistema les permitiría acceso a funciones necesarias únicamente para los operadores como lo es su nómina, unidades asignadas a ellos, modificar sus tiempos en los viajes, etc.
- **Factura:** Guarda la información relacionada al costo del viaje comisionado vinculando al cliente junto con el viaje.
- **Mantenimiento:** Tabla encargada de almacenar toda la información relacionada al mantenimiento de las unidades debido a que por cuestiones legales se debe de tener un registro constante del mantenimiento de frenos, llantas, aceite, etc. Para hacer valido el seguro en cada unidad.
- **Unidad:** hace referencia a los camiones que conforman la flotilla, mejor denominados como unidades, se encarga de almacenar información acerca de cada uno de estos camiones para poder tener una idea constante de las capacidades de la flota de la empresa, también genera un vínculo con los usuarios a los que son asignados estas unidades para poder tener un registro de responsabilidad.

- **Incidente:** Es una tabla que almacena cualquier suceso que afronte la unidad a lo largo de su viaje variando desde un frenado demasiado rápido hasta la pérdida total de la unidad.
- **Users:** Tabla generada para el almacenamiento de la información básica que todo usuario del sistema debe tener. La tabla se creó como una plantilla básica con la cual se generan a los usuarios y que posteriormente son asignados a determinados roles.
- **Ciente:** Tabla con la cual se vincula a ciertos usuarios al rol de clientes en donde se busca almacenar información discal y de facturación de cada usuario cliente.
- **Administración:** Su principal propósito en esta versión es el proveer una tabla con acceso maestro a todo el sistema para facilitar el desarrollo, pero igualmente se piensa como una tabla que presta el rol de administrador a usuarios como lo son los directivos de la empresa para generar una manera conveniente de analizar todos los datos que el sistema genera, la tabla también se generó con la idea prestar a subdividirla para determinar solo ciertos permisos de administrador al usuario en caso de ser necesario.

### 5.5.- Diseño de la interfaz de usuario(UI).

El diseño de la interfaz de usuario fue desarrollado con un enfoque profesional, limpio y funcional utilizando la herramienta Figma. Esta plataforma permitió crear maquetas interactivas y colaborativas para simular de forma precisa el comportamiento de la aplicación final. A continuación, se describen los elementos clave que componen la estructura visual [\(Imagen 13\)](#):

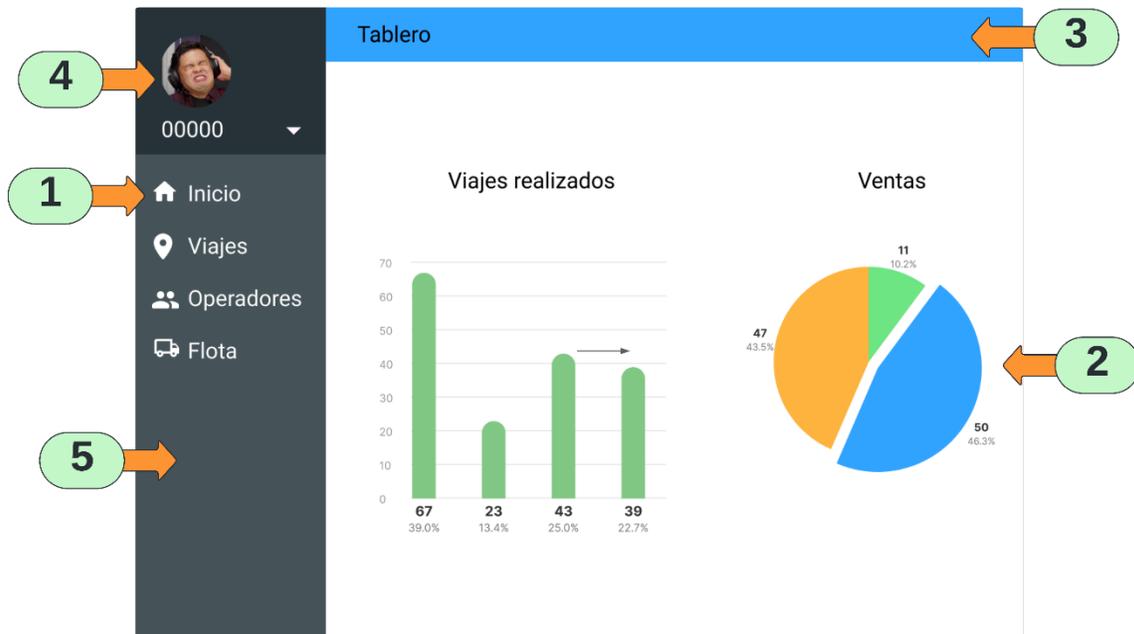


Imagen 13: Diseño de la UI.

1. **Menú lateral de navegación:** Este panel oscuro a la izquierda actúa como el eje central de navegación. Se utilizó un fondo gris oscuro para dar contraste y destacar los iconos blancos, que representan las secciones principales como "Inicio", "Viajes", "Operadores" y "Flota". Este diseño facilita el acceso rápido y claro a las funciones principales de la plataforma. La fuente empleada es Roboto, que aporta un estilo moderno, limpio y legible.
2. **Área de contenido dinámico:** Es la sección principal donde se muestra la información relevante según la opción seleccionada en el menú. En este caso, se visualizan gráficos de barras y de pastel que representan datos sobre viajes y ventas. Esta zona fue diseñada para ser clara y sin elementos distractores, permitiendo una interpretación visual rápida. Los gráficos utilizan colores vibrantes como verde, naranja y azul para distinguir bien cada categoría.
3. **Encabezado o barra superior:** La barra azul superior funciona como un título o encabezado de la pantalla activa. Ayuda a contextualizar al usuario sobre la sección que está visualizando, en este caso, "Tablero". El color azul fue seleccionado por su asociación con confianza y profesionalismo. La tipografía blanca sobre azul mantiene un buen contraste visual.

4. **Perfil del usuario:** En la parte superior izquierda se encuentra el avatar del usuario, acompañado de su ID. Este pequeño bloque personaliza la experiencia y permite una identificación rápida. El diseño circular del avatar suaviza visualmente el espacio y le da un toque más amigable al entorno de trabajo.
5. **Distribución general y estilo visual:** La disposición en columnas, el uso de íconos junto a texto, y una paleta de colores sobria pero contrastante fueron cuidadosamente seleccionados para transmitir orden y eficiencia. Se buscó una estética profesional con jerarquía visual clara, guiando al usuario de manera intuitiva por la plataforma.

## Capítulo 6.- Desarrollo de la aplicación.

En este capítulo se aborda el desarrollo del TMS, detallando su estructura, funcionalidades y componentes principales. A lo largo del contenido se explica la implementación de las distintas partes del sistema. Todo el código fuente relacionado con el desarrollo del TMS está disponible en el repositorio de GitHub: <https://github.com/Tlacuache13/TMS>.

### 6.1.- Creación de la Base de Datos en MySQL Server.

Con la aprobación consolidada del diseño de la base de datos mediante el modelo ER, se dio paso a la fase de implementación. La transición del diseño teórico a la realidad práctica es en MySQL Server, utilizando la herramienta visual MySQL Workbench. Dada la restricción de tiempo en este proyecto, la elección de herramientas eficientes es fundamental para mantener un flujo de trabajo efectivo y cumplir con los plazos establecidos.

MySQL Workbench al ofrecer una interfaz visual intuitiva, simplifica la creación y gestión de la base de datos. Acelerando el proceso de implementación, proporcionando una visión clara y accesible de la estructura de la base de datos mediante su GUI que permite alterar y observar la DB en constante evolución.

### 6.2.- Configuración de Flask y Python.

La instalación de Python en la PC designada para el desarrollo fue el primer paso en la configuración del entorno de desarrollo. Lo que fue seguido por la instalación del compilador necesario para VSCode, asegurando una integración suave con el entorno de desarrollo elegido. Posteriormente, se procedió con la instalación de Flask, un framework de desarrollo web para Python que facilitaría la creación de la interfaz de usuario ([Imagen 14](#)).

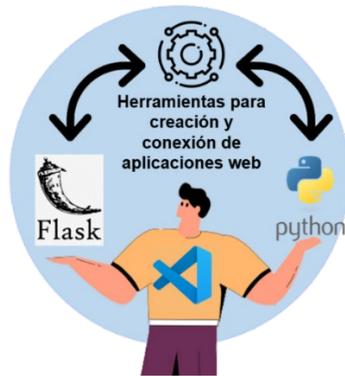


Imagen 14: Diagrama de la conexión de Flask y Python.

Para ejecutar el código desarrollado en Python, se instalan las extensiones necesarias dentro de VSCode (Imagen 15), permitiendo interpretar y ejecutar los scripts escritos. Asimismo, se incorpora el framework Flask mediante el comando `$ pip install Flask` dentro de la terminal de VSCode, lo que habilita la creación de aplicaciones web y la gestión de rutas en el backend.

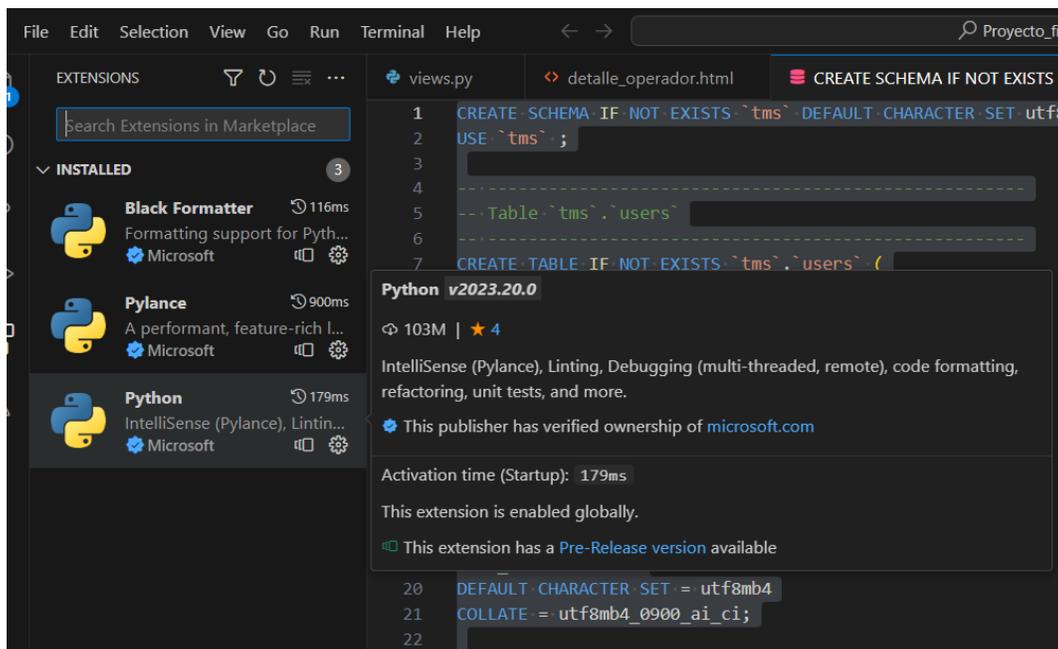


Imagen 15: Extensiones de Python instaladas en Visual Studio Code.

El proceso de configuración se realiza con un enfoque meticuloso, garantizando la integración coherente de cada componente. La elección de Visual Studio Code y Flask se fundamenta en su funcionalidad, su capacidad de agilizar el flujo de trabajo y su aporte en la construcción de una base sólida que permite el desarrollo continuo.

### **6.3.- Desarrollo del código.**

En la presente etapa del progreso del proyecto, se llevó a cabo la tarea de establecer el proceso para la creación de una página web en Flask. La implementación de la página web requirió únicamente la creación de dos archivos con extensión “.py”. El primero actúa como el archivo central, desde el cual se genera la aplicación web, mientras que el segundo se encarga de gestionar todas las funciones y vistas en el marco de la página web. Con un enfoque bifurcado en archivos se facilita la modularidad en el desarrollo del código, permitiendo una estructura organizada de la aplicación web en Flask.

El sistema sigue una arquitectura cliente-servidor, donde distintos usuarios acceden al servidor encargado de ejecutar la aplicación y gestionar las peticiones de cada uno. La arquitectura resulta indispensable, al permitir centralizar la gestión de datos, garantizando la integridad de la información y mantener un control eficiente sobre los procesos operativos. Además, facilita el acceso simultáneo de múltiples usuarios desde diferentes ubicaciones, manteniendo la sincronización en tiempo real de las operaciones. La [imagen 16](#) ilustra la arquitectura de la aplicación web. La interfaz de usuario se compone de diversas páginas HTML (Inicio.html, Operadores.html, Carga.html, entre otras), gestionadas por el archivo *views.py*, que actúa como controlador y enlaza las vistas con la lógica de la aplicación. El controlador establece la conexión con la base de datos mediante los módulos *conexion.py* y *pool.py*, los cuales administran las conexiones y garantizan un acceso seguro y eficiente a la información almacenada. Por su parte, el archivo *app.py* ejecuta la aplicación web, integrando todos los componentes y generando seguridad a través de certificados que permiten operar sobre el protocolo HTTPS.

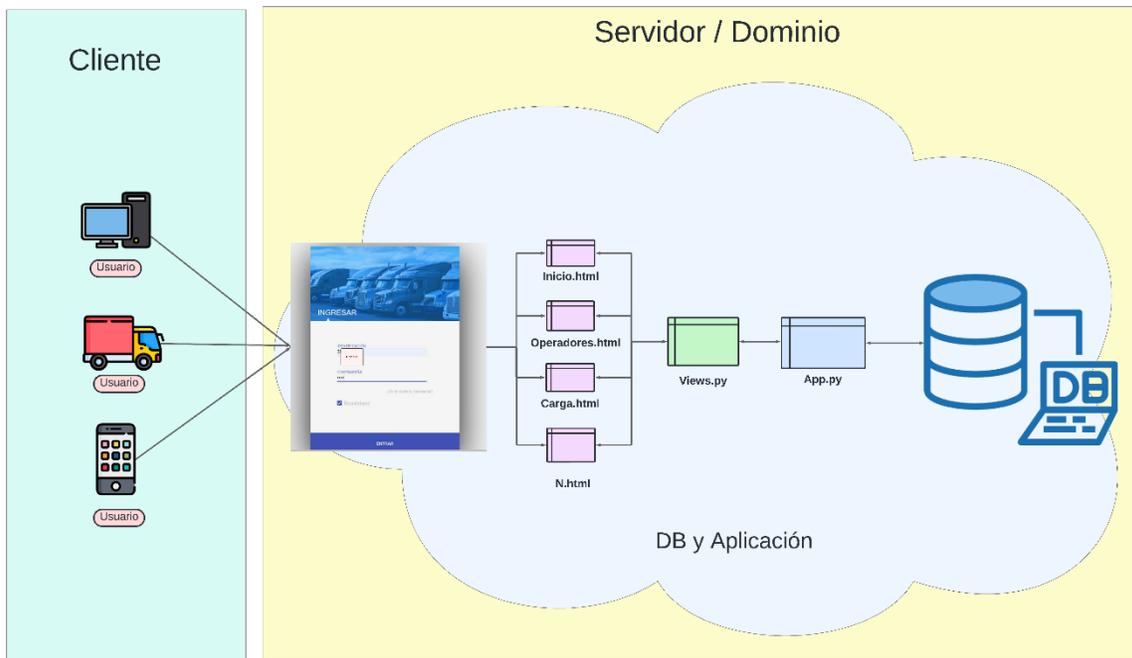


Imagen 16: Diagrama conceptual del funcionamiento del sistema.

El acceso al sistema se realiza a través de diferentes plataformas y dispositivos, como aplicaciones web ejecutadas desde computadoras o dispositivos móviles. En el caso de los usuarios operadores, inicialmente se consideró el desarrollo de una aplicación de escritorio para ser ejecutada en las computadoras instaladas en las unidades. Sin embargo, esta opción fue descartada al identificarse posibles problemas de compatibilidad y seguridad, derivados de las características de los equipos utilizados, los cuales suelen ser computadoras de bajo costo, como las Chromebooks, que operan principalmente con conexión a internet mediante tarjetas SIM. Por esta razón, se optó por implementar una aplicación web con diseño responsivo, ya que representa la alternativa más adecuada, flexible y funcional para el desarrollo del TMS.

#### 6.4.- Construcción de la UI.

Una vez completada la programación integral del framework, se procedió con el diseño y desarrollo de las interfaces visuales de la aplicación. Este proceso se realizó utilizando HTML, CSS, JavaScript y el framework de diseño Material Design Lite, seleccionados por su capacidad para cumplir con los requerimientos funcionales y de experiencia de usuario.

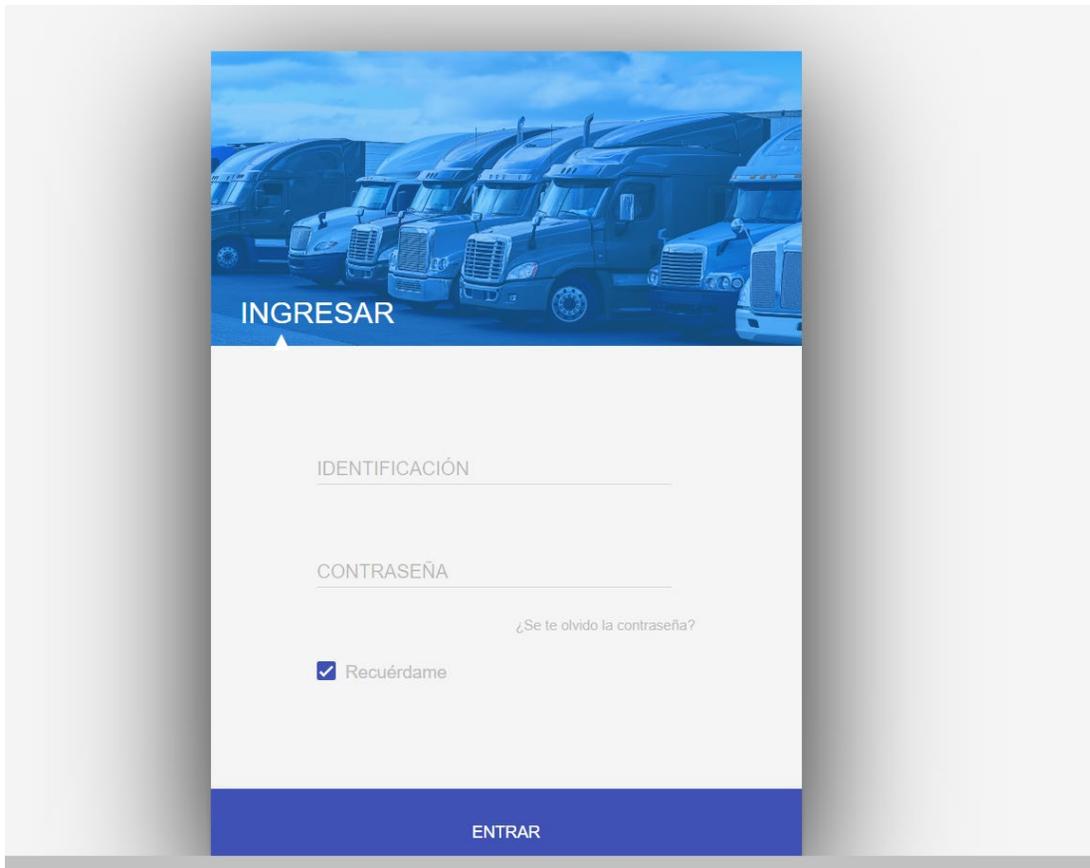
La elección de HTML respondió a su capacidad para estructurar de manera lógica y semántica los elementos del contenido, facilitando una organización clara de la información y estableciendo la base para una navegación intuitiva dentro del sistema.

Por su parte, CSS permitió definir la apariencia visual de cada elemento, gestionando aspectos como la disposición, los colores, las tipografías y el comportamiento de los elementos visuales en distintos tamaños de pantalla.

La incorporación de JavaScript proporcionó la capacidad de dotar de dinamismo e interactividad a las interfaces, permitiendo que la aplicación respondiera de manera eficiente a las acciones del usuario, mejorando así la experiencia de uso.

Adicionalmente, se integró Material Design Lite como guía visual y conjunto de componentes prediseñados. Esta herramienta permitió mantener un diseño coherente, estético y profesional, alineado con estándares modernos de usabilidad y experiencia de usuario, además de reducir tiempos en la creación de estilos personalizados.

Esta combinación de tecnologías facilitó el desarrollo de una interfaz adaptable, intuitiva y funcional, capaz de ajustarse a distintos dispositivos y tamaños de pantalla, optimizando así la usabilidad del sistema. A continuación, se presentan capturas de pantalla que ilustran la apariencia y funcionamiento del sistema, como se muestra en la [Imagen 17](#), correspondiente a la página de inicio para el ingreso de credenciales, y en las imágenes siguientes, donde se observan distintas vistas de las secciones disponibles para los usuarios.



*Imagen 17: Página de inicio de sesión de la aplicación.*

En la [imagen 18](#) se puede observar la página de inicio después de iniciar sesión disponible para todos los tipos de usuarios, pero de uso esencial para los administradores que genera una página con todas las gráficas principales que el usuario desea visualizar, útiles para la toma de decisiones. El acceso al sistema sigue las políticas de seguridad establecidas por la empresa, las cuales requieren que el nombre de usuario sea una identificación numérica de seis dígitos, mientras que la contraseña debe contener al menos doce caracteres, incluyendo una letra mayúscula, un número y un símbolo, garantizando así un nivel adecuado de seguridad en el control de acceso.

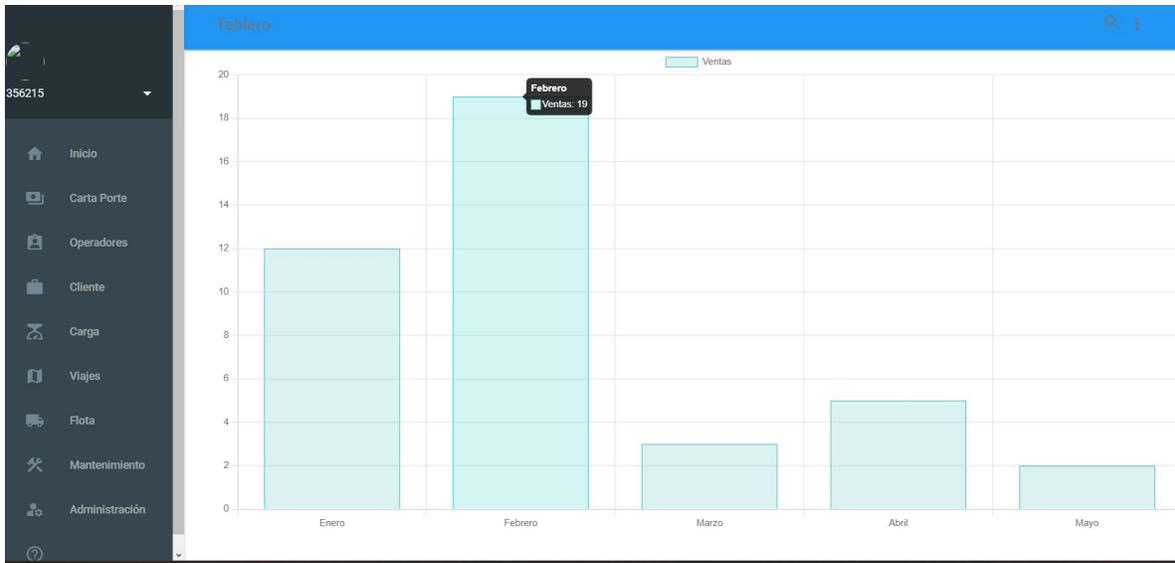


Imagen 18: Sección de inicio donde se muestran graficas.

La [imagen 19](#) muestra los elementos contenidos dentro de la tabla de viajes, donde se observa un registro con estado terminado y otro con estado activo, además de las acciones disponibles para gestionar cada uno de estos viajes.

<input type="checkbox"/>	IdViaje	Destino	Origen	Estado	Acciones
<input type="checkbox"/>	1	Pachuca	Tulancingo	TERMINADO	
<input type="checkbox"/>	3	CDMX	Huasca	ACTIVO	

Imagen 19: Sección de viajes donde se observan todos los registros de viajes.

La [imagen 20](#) muestra la información detallada de un viaje, la cual corresponde a los datos que deben presentarse en la carta porte. Esta vista forma parte de la sección de viajes disponible para los administradores y para aquellos usuarios que, de acuerdo con los permisos asignados, tienen autorizado su acceso y consulta.

### Información de Viaje

Folio: 1 Estado:TERMINADO  
Descripción: Cuarzo  
Origen: Pachuca Destino: Tulancingo  
Fecha de Inicio: 2023-09-26 Fecha de Termino:2023-09-27  
Hora de Inicio: 7:47:00 Hora de Termino:14:00:00  
Cliente: Kira  
Unidad: 1  
Operador: 1

### Detalle de Carga

Descripción de la Carga	Peso (kg)	Dimensiones (m)	Cantidad
Electrodomesticos		3x2x1	1

Peso: 1 Tonelada

[VER LOG](#)

Imagen 20: Ventana de detalles de un viaje en específico.

## 6.5.- Conexión con la base de datos.

El desarrollo de la aplicación se basa en una arquitectura cliente-servidor de dos niveles, que permite separar la interfaz de usuario del procesamiento centralizado. Esta separación contribuye a un acceso seguro y eficiente a la información, optimizando la gestión de datos y facilitando la escalabilidad del sistema. La conexión con la base de datos MySQL se establece mediante esta arquitectura, garantizando la centralización del almacenamiento y procesamiento de información crítica como rutas, inventarios, pedidos y seguimiento de envíos.

La seguridad en la comunicación se asegura mediante el uso del protocolo HTTPS, implementado a través de la librería flask\_sslify, que cifra los datos transmitidos entre el cliente y el servidor para protegerlos contra accesos no autorizados. Además, se emplea la encriptación SHA-256 para salvaguardar la integridad y confidencialidad de datos sensibles, especialmente en la gestión de credenciales. Complementariamente, se utilizan consultas parametrizadas para prevenir ataques de inyección SQL, controlando la forma en que los datos ingresan a las consultas y evitando manipulaciones maliciosas. La gestión de sesiones y la encriptación de credenciales refuerzan la autenticación de usuarios, protegiendo la información sensible.

El modelo de dos niveles facilita la escalabilidad, permite acceso simultáneo a múltiples usuarios y garantiza que los datos estén siempre actualizados, proporcionando una vista en tiempo real esencial para la toma de decisiones en logística y gestión de transporte. El servidor centraliza el procesamiento y almacenamiento, mientras que las interfaces web o aplicaciones cliente permiten a los usuarios interactuar con el sistema desde cualquier ubicación, optimizando la operación y el mantenimiento sin interrupciones.

El flujo de funcionamiento simplificado dentro de esta arquitectura es el siguiente: los usuarios envían solicitudes para realizar tareas específicas; el servidor procesa estas solicitudes, consulta la base de datos y ejecuta las operaciones requeridas; finalmente, el servidor devuelve los resultados para su presentación en las interfaces de usuario.

Tras diseñar las plantillas para las vistas de la aplicación web, se inició la fase de conexión con la base de datos, implementando dos archivos Python con funciones específicas para este propósito. El archivo `conexión.py` establece la comunicación con la base de datos a través del puerto 3306. Sin embargo, esta conexión presenta limitaciones para manejar múltiples solicitudes simultáneas, lo que afecta la estabilidad y rendimiento del sistema cuando varios usuarios acceden concurrentemente.

Para superar esta limitación, se desarrolló un pool de conexiones mediante el archivo `pool.py`, el cual administra un conjunto de conexiones activas y reutilizables. Este mecanismo evita la creación y cierre constante de conexiones, optimizando el uso de recursos y garantizando un flujo de datos eficiente. El pool de conexiones mejora la escalabilidad, reduce errores por saturación y permite una operación estable en entornos con múltiples usuarios, considerando que las empresas de transporte suelen operar con flotillas de 50 a 200 unidades, generando numerosas solicitudes simultáneas.

El archivo `pool.py` utiliza la librería `mysql.connector.pooling` para crear un grupo de conexiones llamado *mypool*, con una capacidad de hasta diez conexiones simultáneas. A través de la función `get_connection_from_pool()`, la aplicación solicita y reutiliza conexiones según la demanda, solucionando las limitaciones del archivo `conexión.py`, que establece una única conexión directa adecuada solo para escenarios con bajo volumen de usuarios.

Ambos archivos, `conexión.py` y `pool.py`, trabajan en conjunto para gestionar de forma eficiente la comunicación con la base de datos, asegurando estabilidad y flexibilidad en el acceso a la información necesaria para el funcionamiento del sistema.

Una vez establecida la conexión, se definieron los queries, setters y getters en el archivo `views.py`, que administra las rutas y solicitudes entre la interfaz de usuario y el servidor. Este archivo ejecuta las operaciones para interactuar con la base de datos, facilitando la creación, consulta, actualización y eliminación de registros. Cada ruta está asociada a una función que procesa las solicitudes según los permisos asignados a cada usuario. Por ejemplo, el personal administrativo posee acceso completo para gestionar todos los datos, mientras que los operadores pueden solo generar registros en las tablas relacionadas con sus actividades, como la tabla “viaje”. Esta fase es una de las más extensas y se mantiene en constante revisión para garantizar la integridad y seguridad del sistema.

Al alcanzar este punto, con la aplicación web conectada y realizando peticiones, se resalta la importancia de la seguridad en la comunicación y gestión de datos. La utilización de `flask_sslify` para redirigir el tráfico a HTTPS, junto con la encriptación SHA-256, asegura la protección y confidencialidad de la información sensible, especialmente credenciales y datos críticos. Asimismo, el uso de consultas parametrizadas previene inyecciones SQL, fortaleciendo la seguridad general del sistema. Estas medidas, combinadas con la gestión adecuada de sesiones y autenticación, protegen la aplicación contra accesos no autorizados y posibles vulnerabilidades. Se reconoce que la seguridad es un proceso continuo, por lo que se planifican pruebas periódicas para detectar y corregir posibles riesgos.

Finalmente, se añadieron funciones para mejorar la utilidad del sistema, como la exportación de datos en formatos estándar como PDF y Excel. Esto facilita la integración con otras herramientas empresariales y el intercambio de información con usuarios externos al sistema central. La incorporación gradual de estas funciones se planifica en iteraciones futuras, basadas en la retroalimentación de los usuarios y las necesidades cambiantes del entorno empresarial.

## 6.6.- Conexión con el dominio público.

Para permitir que la aplicación desarrollada con Flask sea accesible desde Internet, se llevó a cabo la adquisición del dominio `tlacu.org`. El proceso de despliegue inició con la obtención de la dirección IP pública del servidor, la cual, por razones de privacidad y siguiendo criterios comunes en documentación técnica —tal como lo sugieren organismos como el Internet Engineering Task Force (IETF) y las recomendaciones en materia de ciberseguridad de la Open Worldwide Application Security Project (OWASP)—, se representa utilizando valores en cero.

Posteriormente, se configuró un registro tipo A dentro del sistema de nombres de dominio (DNS) del dominio adquirido, con el fin de redirigir todas las solicitudes entrantes hacia la dirección IP pública del servidor donde se aloja la aplicación. De este modo, cualquier usuario que ingrese al dominio es direccionado directamente al entorno del servidor.

Dentro del código de la aplicación, se habilitó la recepción de conexiones desde cualquier interfaz de red mediante la instrucción: `app.run(host="0.0.0.0", port=000)`. Este ajuste permite que el servidor escuche solicitudes provenientes de cualquier dispositivo con conexión a Internet, sin restricciones por dirección IP específica.

En una primera etapa, la seguridad de las comunicaciones se gestionó mediante la implementación de la librería Flask-SSLify, cuyo objetivo es forzar la redirección de todo el tráfico HTTP hacia conexiones HTTPS. Esta solución fue empleada como mecanismo provisional para verificar la correcta gestión de conexiones seguras durante las pruebas iniciales del entorno.

Posteriormente, con el objetivo de fortalecer la seguridad y cumplir con estándares adecuados para entornos de producción, se generó un certificado SSL/TLS gratuito mediante Let's Encrypt, utilizando la herramienta Certbot. Dicho certificado se integró en la configuración del servidor, permitiendo que todas las conexiones sean cifradas y verificadas.

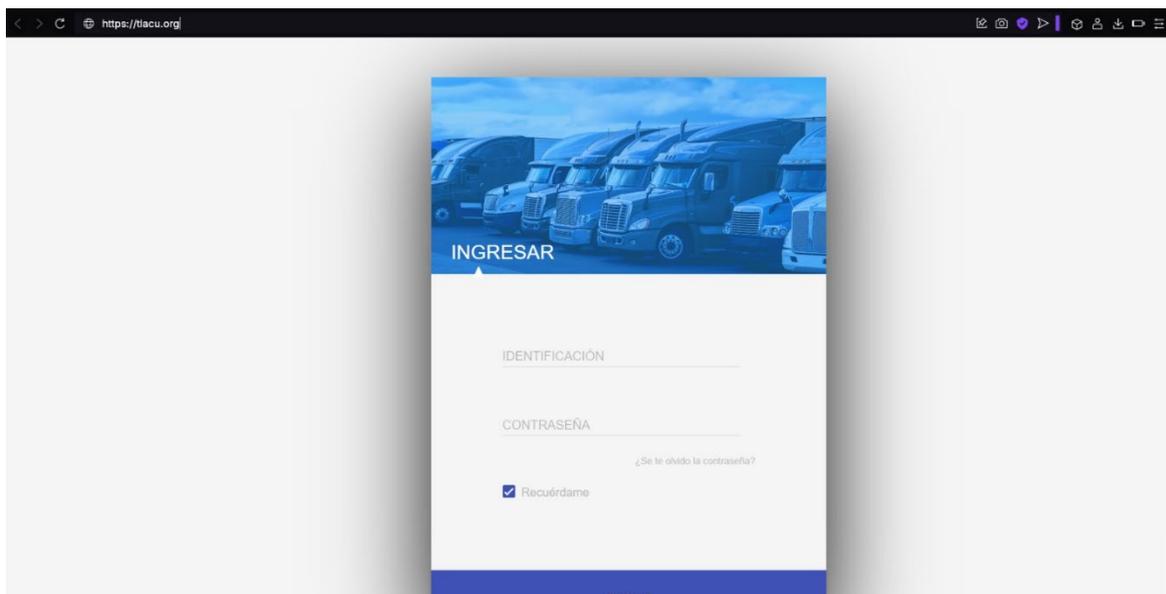
El protocolo SSL/TLS empleado en el entorno garantiza la confidencialidad, autenticación y la integridad de los datos transmitidos. Lograndolo mediante una combinación de algoritmos criptográficos robustos, donde se utiliza Advanced Encryption Standard (AES) para el cifrado simétrico de los datos y funciones hash como Secure Hash Algorithm 256 (SHA-256)

para asegurar la integridad de la información. En este contexto, SHA-256 permite validar que los datos no han sido alterados durante su transmisión, mientras que la negociación de claves seguras se realiza mediante algoritmos como Elliptic Curve Diffie-Hellman Ephemeral (ECDHE), y la autenticación del servidor se verifica mediante certificados digitales.

La ejecución de la aplicación con soporte SSL se realizó con la siguiente instrucción:  
`app.run(host="0.0.0.0", port=000, ssl_context=("ruta/fullchain.pem", "ruta/privkey.pem"))`

Finalmente, con el fin de garantizar la disponibilidad continua y mejorar la gestión de múltiples conexiones, se incorpora Gunicorn (Green Unicorn), siendo una biblioteca de Flask permitiendo servir como servidor de aplicaciones compatible con Web Server Gateway Interface (WSGI). Su función es actuar como un intermediario entre la aplicación desarrollada en Flask y el servidor, permitiendo manejar múltiples procesos simultáneos. Esto optimiza tanto la estabilidad como el rendimiento, distribuyendo eficientemente las solicitudes entrantes hacia los diferentes hilos o procesos que ejecutan la aplicación.

Adicionalmente, se configura un servicio en Systemd que permite que la aplicación se inicie automáticamente al arrancar el servidor, asegurando su funcionamiento constante sin necesidad de intervención manual [imagen 21](#).



*Imagen 21: Aplicación ejecutándose en el dominio tlacu.org*

## **6.7.- Beneficios observados para la organización.**

Después de vincular la aplicación al dominio público se formó a un grupo de prueba de usuarios dentro de la empresa CANACE Transportes permitiéndoles interactuar con el sistema para poder identificar sus opiniones, los beneficios que otorga y posibles áreas de mejora. Los comentarios señalan que la disponibilidad constante de datos genera una impresión de orden y facilita la consulta de información sin depender de registros dispersos. Esta sensación de acceso inmediato sugiere que las decisiones pueden tomarse con más agilidad y que la coordinación entre áreas resulta más fluida.

Algunos usuarios perciben que la plataforma ayuda a optimizar tareas administrativas, al centralizar procesos que antes se realizaban en distintos formatos, reduciendo pasos repetitivos. Otros mencionan que la posibilidad de anticipar incidentes y mantener un historial unificado favorece la organización de las operaciones diarias.

Basado en estos comentarios se lograron identificar 6 beneficios principales:

- **Sensación de control operativo**
- **Acceso centralizado a la información**
- **La posibilidad de mejor comunicación entre áreas**
- **Mayor seguridad al momento de manejar información**
- **Un uso más eficiente de los recursos**

De manera general, la retroalimentación del grupo de prueba apunta a que el sistema contribuye a una mayor claridad en la administración de datos, a un uso más eficiente de los recursos y a un ambiente de trabajo que se siente más seguro, basándose únicamente en la percepción de quienes participaron en la evaluación. Las opiniones completas de los usuarios se encuentran recopiladas en el Anexo 2.- Cuestionario para su consulta detallada.

## Trabajos futuros.

Enfocándose a la mejora continúa buscando la adaptación de las necesidades cambiantes del entorno logístico, se identifican diversas áreas de desarrollo y expansión para futuros trabajos en el sistema de gestión de transporte (TMS). Algunas de estas posibles mejoras que se plantean dándose el caso de continuar este proyecto son las siguientes:

- **Replantear y mejorar la exportación a plataformas externas:** Actualmente la exportación a archivos .pdf y .xlsx solo se encuentra presente en la vista general de los registros de cada apartado debido a problemas al momento de generar los archivos, junto con la situación donde la empresa aún está considerando si necesita exportar cada registro individualmente. Proponiéndose, en caso de que se continúe el desarrollo, el mejorar la exportación de registros para que no solo exporte los datos de la pantalla, sino que también exporte otros campos de los registros todo basándose en si lo desea el usuario en un panel con filtros u opciones dedicado a generar estos archivos.
- **Integración con sistemas externos:** En el estado presente del proyecto solo se cuenta con una parte del sistema como tal que es la base de datos en la donde se almacenan todos los registros útiles y la interfaz de usuario, mediante la cual permite la interacción. Se espera, en caso de continuar este proyecto, vincular con los otros sistemas necesarios para obtener los datos de manera automática como es su ubicación a través del GPS de las unidades, las computadoras de las diferentes áreas de la empresa e incluso posiblemente los clientes o proveedores.
- **Importación de archivos:** En la versión que se entrega del prototipo solo se cuenta con un botón placeholder de importación de archivos .XML para facturación. Pero en un futuro se busca solucionar los errores de importación y compatibilidad que impiden que se habilite esta función.
- **Mejoras en la experiencia del usuario (UX):** Realizar estudios de usabilidad adicionales, así como recopilar retroalimentación de los usuarios para identificar áreas específicas de la interfaz de usuario susceptibles a mejoras, con el objetivo de hacer el sistema más intuitivo y fácil de usar.

- **Habilitar la creación de diagramas en base a los registros:** En la página principal solo se cuenta con un ejemplo de cómo se planea que luzcan los diagramas basados en los registros, un trabajo futuro es el corregir los errores que impiden que se generen diagramas dinámicos en base a los registros.
- **Expansión de capacidades de informes:** Ampliar las capacidades de generación de informes, incluyendo métricas específicas del sector, análisis de tendencias a largo plazo y herramientas visuales más avanzadas para facilitar la interpretación de datos.

## **Conclusiones.**

La ejecución exitosa del TMS ha validado de manera concluyente la viabilidad de dicho concepto, destacando mejoras sustanciales en eficiencia, visibilidad, así como seguridad de los procesos logísticos para la empresa CANACE Transportes S.A. de C.V. La escalabilidad junto con la flexibilidad del sistema ha sido probada, mostrando su capacidad para adaptarse al crecimiento futuro de diversas empresas, además de personalizarse según necesidades específicas. Los resultados subrayan el potencial sustancial del TMS en la transformación y potenciación de las operaciones logísticas, estableciendo una base sólida en su implementación completa con posibles expansiones futuras.

Aunque el prototipo actual representa solo una fase inicial de un sistema más complejo en desarrollo, se puede concluir de manera afirmativa que constituye un paso crucial en la dirección correcta hacia la creación de soluciones integrales destinadas a empresas dedicadas al autotransporte de carga. El enfoque en la integridad, seguridad e intereses tanto de la empresa como de sus clientes y empleados se refleja claramente en el diseño junto con la funcionalidad del prototipo. La atención dedicada a dichas consideraciones resulta fundamental en la garantía del éxito junto con la aceptación del sistema a medida que evoluciona hacia su implementación completa. El prototipo sirve como un sólido fundamento para el desarrollo continuo, demostrando el compromiso de la solución con la mejora de las operaciones logísticas además de la satisfacción integral de todas las partes involucradas en el proceso de transporte de carga.

## Anexos

### Anexo 1.- Manual de usuario.

En el estado actual del sistema primero se debe de iniciar el servidor de la aplicación y acceder a la dirección que esta marca.

Para garantizar un funcionamiento estable, seguro y eficiente del TMS, es indispensable contar con un servidor que cumpla con las características mínimas recomendadas de hardware. Estas especificaciones permiten gestionar correctamente la base de datos, atender múltiples conexiones simultáneas y mantener la integridad de las operaciones logísticas.

Los requerimientos básicos de hardware para el servidor son los siguientes:

- Procesador: Intel i5 de 8va generación o equivalente (AMD Ryzen 5 o superior).
- Memoria RAM: 8 GB como mínimo (recomendado 16 GB para mayor estabilidad).
- Almacenamiento: 120 GB SSD como mínimo para garantizar velocidad en el acceso a datos.
- Conectividad: Red Ethernet con mínimo 100 Mbps de subida y bajada, preferentemente conexión simétrica.
- Fuente de energía: Estabilizador o respaldo UPS para proteger contra fallos eléctricos.
- Sistema de respaldo: Unidad externa o almacenamiento en la nube para copias de seguridad periódicas.

Estas especificaciones son suficientes para desplegar el sistema en un entorno de producción de baja a media carga, soportando entre 20 y 100 conexiones simultáneas según el volumen de datos y el uso operativo.

Una vez dentro se puede observar la pestaña de inicio de sesión ([Imagen 22](#)), proceda a ingresar su número de cuenta y contraseña registrada la cual se limita conforme a las políticas internas de la empresa.

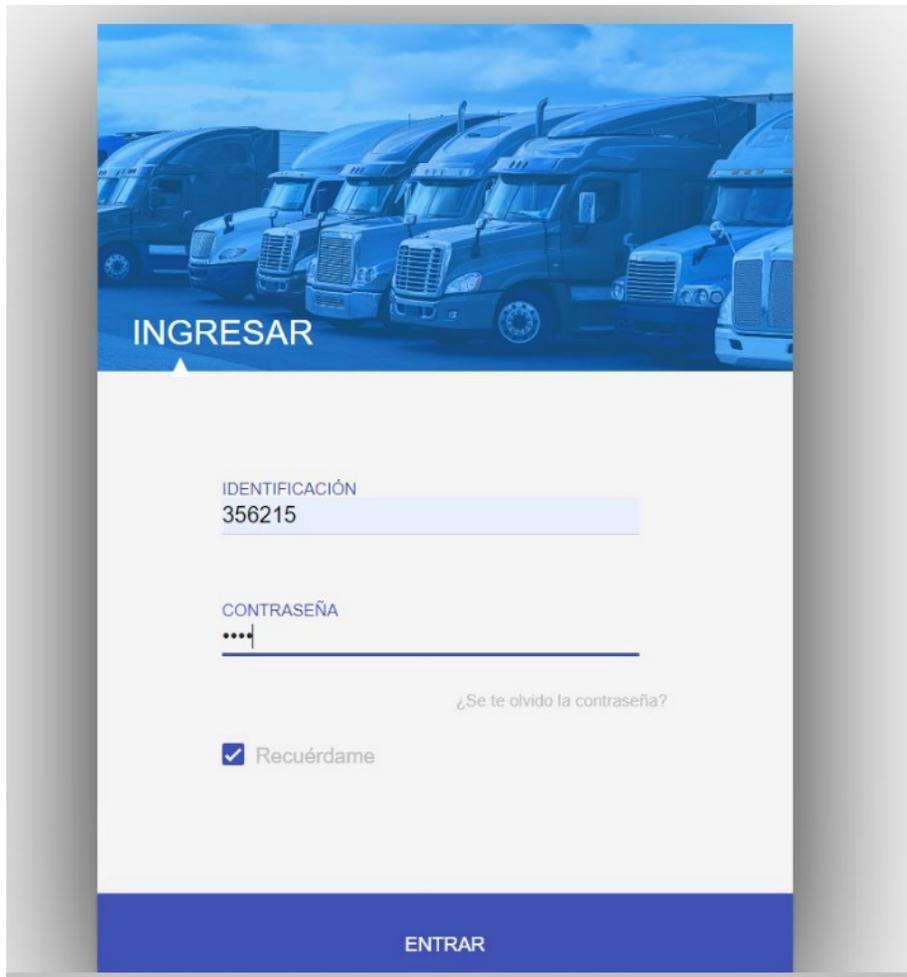


Imagen 22: Página de inicio de sesión.

Al ingresar lo primero que le aparecerá será un panel de datos (Imagen 23) donde encontrará información relevante al usuario dependiendo de sus permisos, que le permiten monitorear rendimiento, tomar decisiones basadas en los datos, etc. y en la parte izquierda habrá una columna con 9 filas, siendo las siguientes:

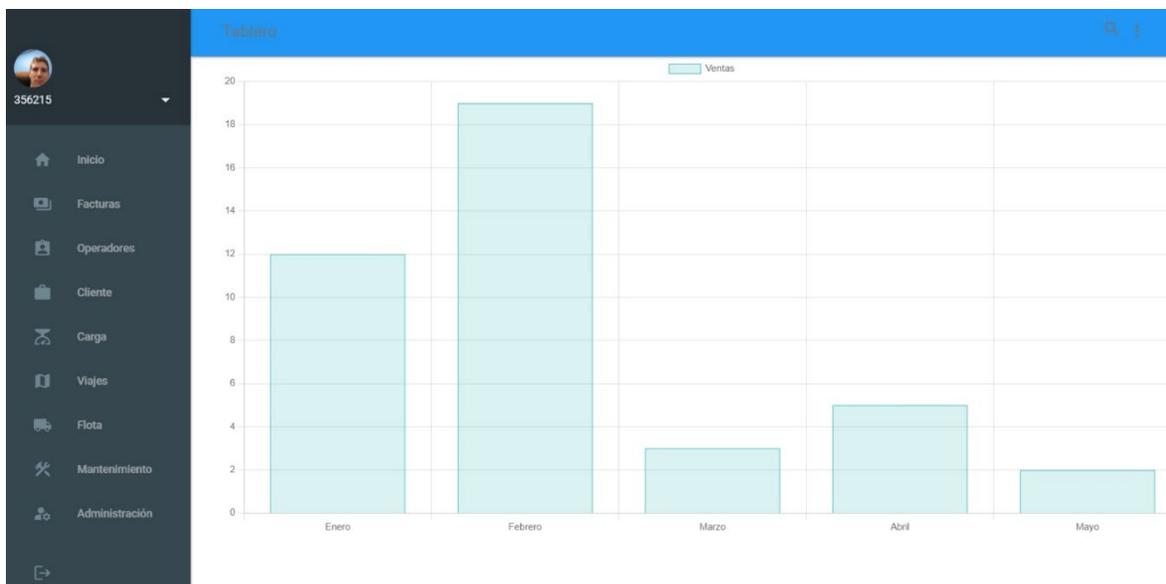


Imagen 23: Panel de datos.

**Inicio:** Esta sección muestra los gráficos generados a base de los registros de otras secciones para poder obtener una representación gráfica de los datos deseados para la toma de decisiones, evaluación de rendimiento o seguimiento de pedidos.

**Facturas:** Muestra una tabla con los datos principales de las facturas (Imagen 24) y en la parte superior tiene un buscador para encontrar registros específicos, debajo de este hay 3 los dos primeros exportan a Excel y a PDF mientras que el tercero importa archivos a XML. En la parte inferior derecha hay un círculo con un signo “+” para crear nuevos registros. Se puede hacer clic en cada uno de los registros para observar todos los datos en detalle, al final de la fila hay dos iconos uno de un lápiz y otro un basurero, el lápiz te envía a una pestaña para editar el registro mientras el basurero lo borra.

IdFactura	IdViajeFact	IdClienteFact	Total	Acciones
1	1	1	10.0	[Icono de lápiz] [Icono de basurero]

Imagen 24: Página de facturas.

**Operadores:** Muestra una tabla con los datos principales de los operadores (Imagen 25) y en la parte superior tiene un buscador para encontrar registros específicos, debajo hay dos botones encargados de exportar a Excel y a PDF respectivamente. En la parte inferior derecha hay un círculo con un signo “+” para crear nuevos registros. Se puede hacer clic en cada uno de los registros permitiendo observar todos los datos en detalle, al final de la fila hay dos iconos: uno de un lápiz y otro un basurero, el lápiz te envía a una pestaña con función de editar el registro mientras el basurero lo borra.

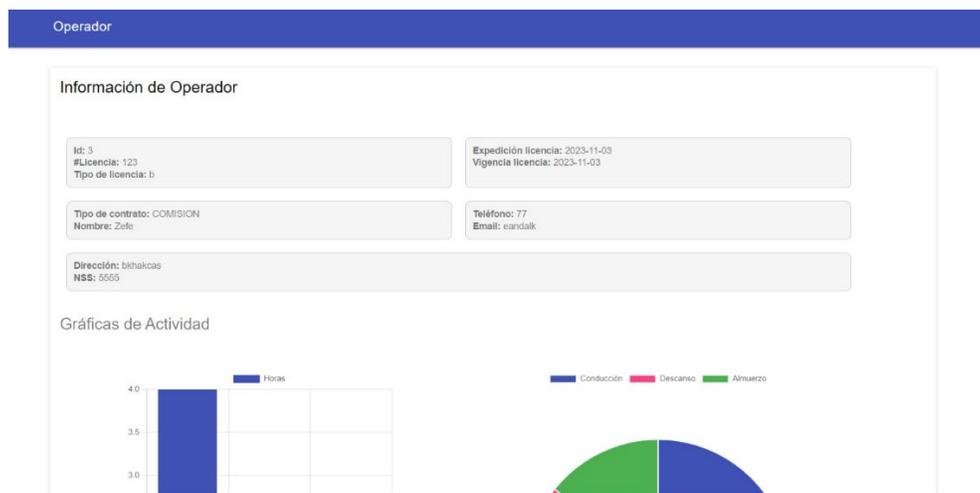


Imagen 25: Información detallada de los operadores.

Al desplazarse hacia abajo en la tabla de operadores puede observar un par de gráficos que representan la actividad del operador (Imagen 26), esto se muestra como un elemento temporal de lo que podría presentarse con los datos del operador para tener una representación gráfica de sus tiempos de actividad.

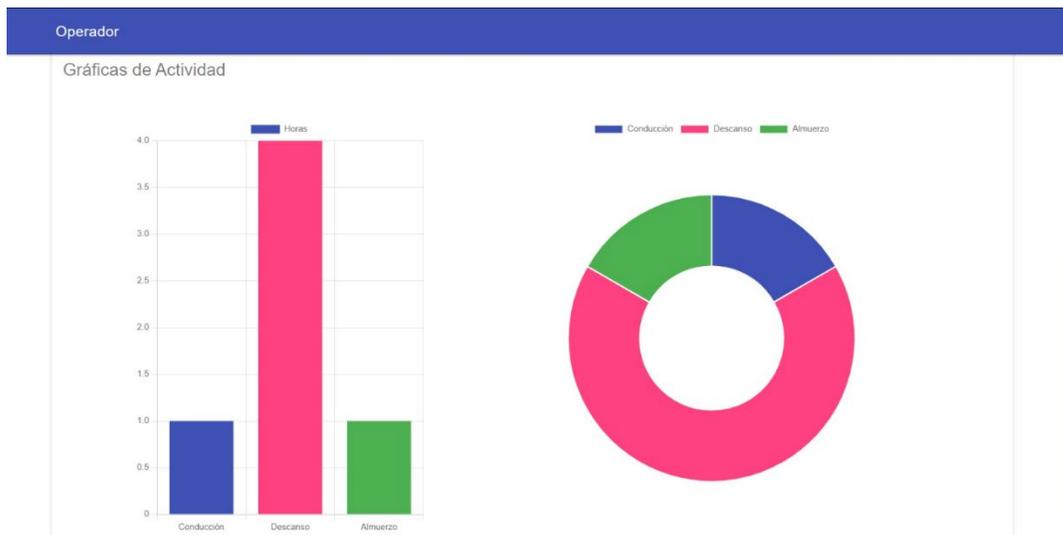


Imagen 26: Gráficas de actividad de los operadores.

**Ciente:** Muestra una tabla con los datos principales de los clientes y en la parte superior tiene un buscador para encontrar registros específicos, debajo hay tres botones los dos primeros exportan a Excel y a PDF mientras que el tercero importa archivos a XML. En la parte inferior derecha hay un círculo con un signo “+”, permitiendo crear nuevos registros. Se puede hacer clic en cada uno de los registros para observar todos los datos en detalle, al final de la fila hay dos iconos uno de un lápiz seguido de un basurero, el lápiz te envía a una pestaña para editar el registro mientras que el basurero lo borra.

**Carga:** Muestra una tabla con los datos principales de las cargas y en la parte superior tiene un buscador con el que se pueden encontrar registros específicos, debajo de dicho buscador hay tres botones los dos primeros exportan a Excel y a PDF mientras que el tercero importa archivos a XML. En la parte inferior derecha hay un círculo con un signo “+” permitiendo la creación nuevos registros. Se puede hacer clic en cada uno de los registros permitiendo observar todos los datos en detalle, al final de la fila hay dos iconos unos de un lápiz y otro un basurero, el lápiz te envía a una pestaña para editar el registro mientras el basurero lo borra.

**Viajes:** Muestra una tabla con los datos principales de los viajes (Imagen 27) y en la parte superior tiene un buscador con el que se pueden encontrar registros específicos; debajo de dicho buscador hay tres botones: los dos primeros exportan a Excel y a PDF, mientras que el tercero importa archivos a XML. En la parte inferior derecha hay un círculo con un signo “+” sirve para crear nuevos registros. Al final de la fila hay dos iconos: uno de un lápiz y otro un basurero. El lápiz te envía a una pestaña desde la que se puede editar el registro, mientras el basurero lo borra.

<input type="checkbox"/>	idViaje	Destino	Origen	Estado	Acciones
<input type="checkbox"/>	1	Pachuca	Tulancingo	TERMINADO	
<input type="checkbox"/>	3	CDMX	Huasca	ACTIVO	

Imagen 27: Página de viajes.

Se puede hacer clic en cada uno de los registros para observar todos los datos en detalle lo cual muestra toda la información registrada en el sistema acerca de los viajes. En la parte inferior central existe un botón con el nombre “ver log” el cual permite ver el log del viaje (Imagen 28) con las métricas actuales de la unidad a lo largo del viaje.

## Log de incidencias

Idlog	HorasR	GPS	VelocidadActual	EficienciaCom	ErrorCode	TiempoJornada	HorasDescanso	HorasLog	SaldoTag	IdViajeLog
1	8	Lat: -22.209457, Lon: -89.167987	60 km/h	85%	0x00	8 horas	1 hora	11:05:55 p.m.	500	12345
2	8	Lat: -61.941754, Lon: -87.514337	60 km/h	85%	0x00	8 horas	1 hora	11:06:35 p.m.	500	12345

Imagen 28: Log del viaje.

**Flota:** Muestra una tabla con los datos principales de las unidades y en la parte superior tiene un buscador con el que se pueden encontrar registros específicos; debajo de dicho buscador hay tres botones: los dos primeros exportan a Excel y a PDF, mientras el tercero importa archivos XML. En la parte inferior derecha hay un círculo con un signo “+”, sirve para crear nuevos registros. Se puede hacer clic en cada uno de los registros para observar todos los datos en detalle; al final de la fila hay dos iconos: uno de un lápiz y otro de un basurero. El lápiz te envía a una pestaña donde se puede editar el registro, mientras el basurero lo borra.

**Mantenimiento:** Muestra una tabla con los datos principales de las unidades en mantenimiento y en la parte superior tiene un buscador facilitando encontrar registros específicos; debajo de dicho buscador hay tres botones: los dos primeros exportan a Excel y a PDF, mientras el tercero importa archivos XML (No es funcional). En la parte inferior derecha hay un círculo con un signo “+”, utilizado para crear nuevos registros. Se puede hacer clic en cada uno de los registros para observar todos los datos en detalle; al final de la fila hay dos iconos: uno de un lápiz y otro de un basurero. El lápiz te envía a una pestaña donde se puede editar el registro, mientras el basurero lo borra.

**Administración:** Muestra una tabla con los datos principales de los usuarios del sistema ya sea administradores u operadores. En la parte inferior derecha hay un círculo con un signo “+”, sirve para crear nuevos registros. Se puede hacer clic en cada uno de los registros para

observar todos los datos en detalle, al final de la fila hay dos iconos uno de un lápiz y otro de un basurero, el lápiz te envía a una pestaña para editar el registro mientras el basurero lo borra.

Cuando se crea un nuevo registro dentro de cualquier tabla se cargará una página similar a la siguiente (Imagen 29), en el cual se podrá poner toda la información necesaria para un nuevo registro y nada más lo que faltaría sería presionar crear:

Carta Porte

**Información de Viaje**

Folio: Estado:

Descripción:

Origen:  Destino:

Fecha de Inicio:  Fecha de Termino:

Hora de Inicio:  Hora de Termino:

Cliente:

Unidad:

Operador:

Peso total:

**Detalle de Carga**

Descripción de la Carga	Peso (kg)	Dimensiones (m)	Cantidad
-------------------------	-----------	-----------------	----------

Subtotal:

**CREAR**

Imagen 29: Página crear viaje.

Al momento de observar los datos en detalle de un registro aparecerá una ventana similar a esta (Imagen 30) con toda la información:

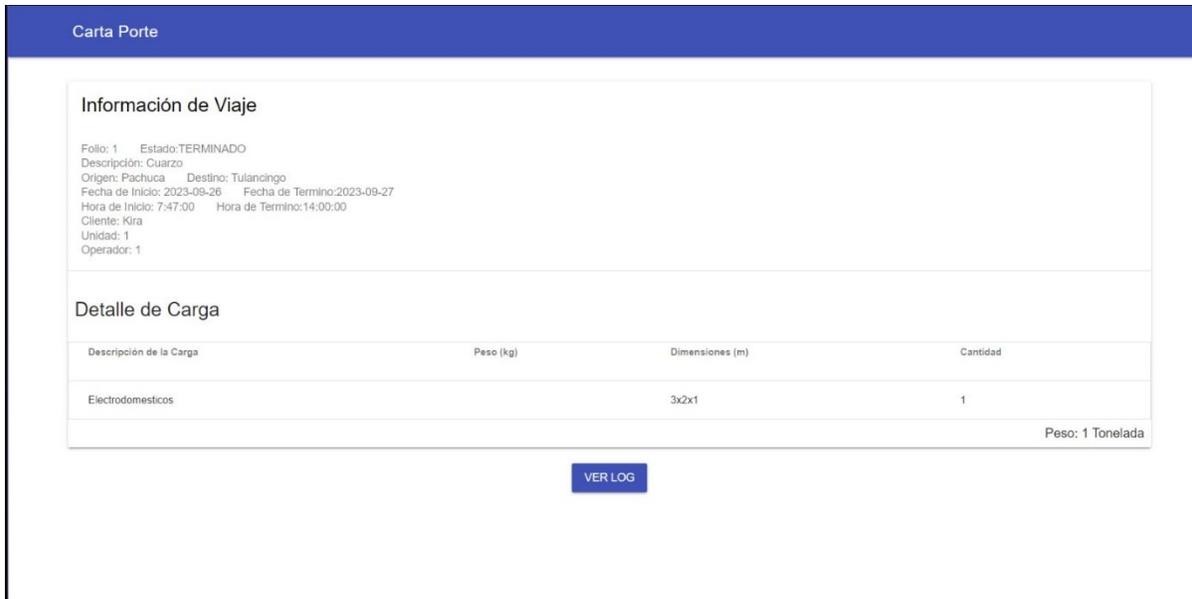


Imagen 30: Página detalle viaje.

Para editar los registros, se debe presionar el icono de lápiz, lo cual abrirá la vista en detalle de los datos con campos editables (Imagen 31):

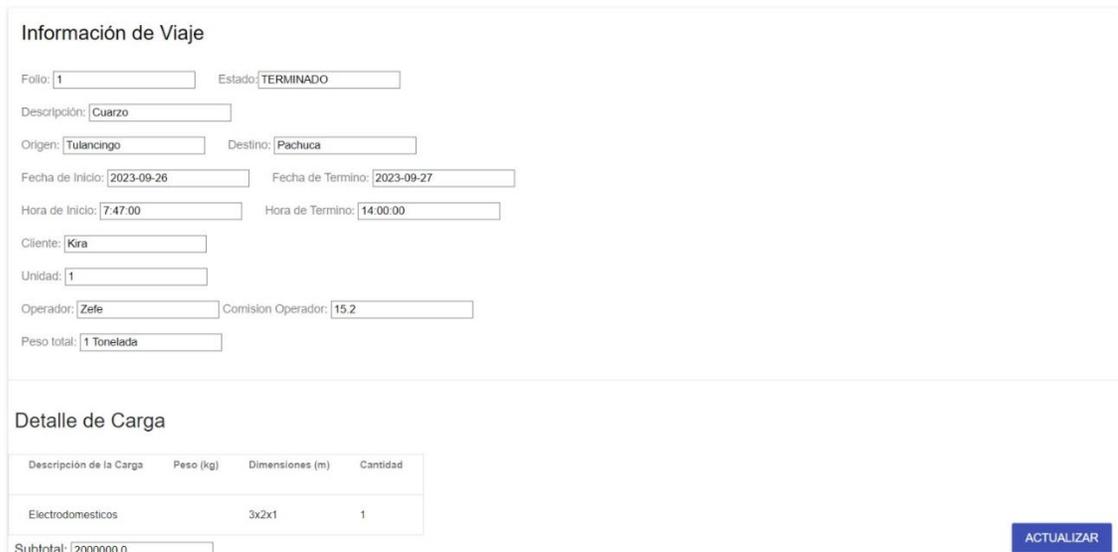


Imagen 31: Página editar viaje.

Para salir del sistema lo único que se tiene que hacer es presionar la flecha a lado de la imagen de usuario, lo cual desplegará una lista con la opción de salir y dar clic en ella llevará a la ventana de inicio de sesión (Imagen 32):

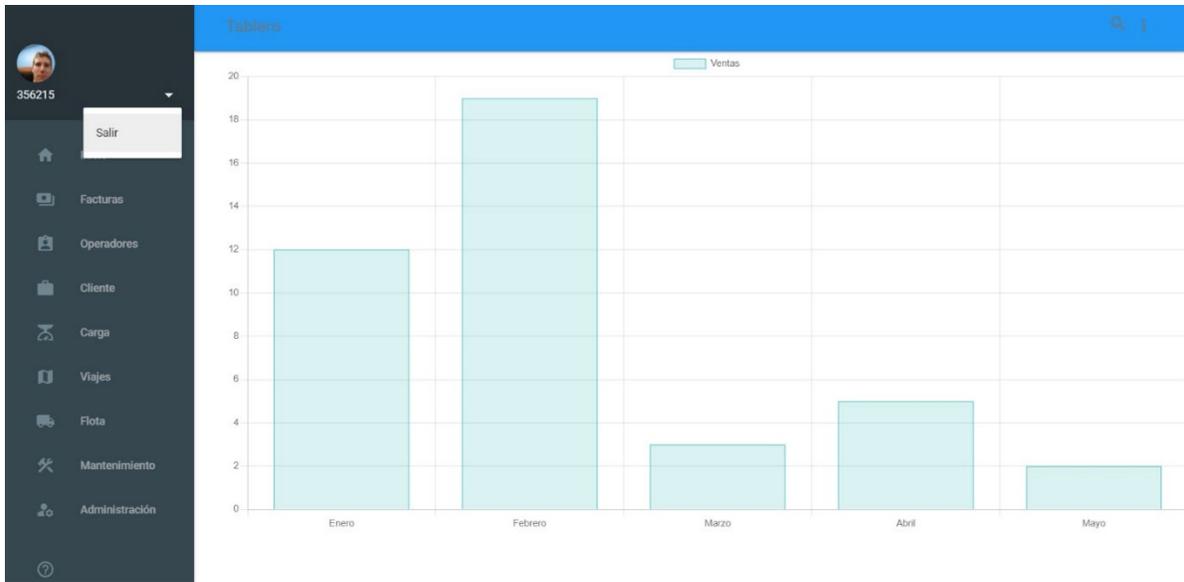


Imagen 32: Página de inicio con botón salir a la vista.

## Anexo 2.- Cuestionario.

Una vez que se dispuso de un prototipo funcional de la aplicación, se llevó a cabo una evaluación práctica con un grupo compuesto por nueve personas pertenecientes a la empresa CANACE. Los individuos fueron seleccionados por su alta probabilidad de interactuar con el sistema en caso de alcanzar su plena completitud. Se le otorgó acceso a la versión actual del sistema con el propósito de realizar pruebas prácticas.

Posteriormente, se les facilitó un formulario diseñado con el objetivo de identificar áreas que requieren una mayor atención o incluso cambios sustanciales dentro del proyecto. La intención principal detrás de tal acción fue recoger de manera estructurada las opiniones de los usuarios respecto a la funcionalidad, usabilidad y cualquier otra dimensión relevante del sistema.

Dicho enfoque no solo permitió la validación práctica del prototipo, también proporcionó una valiosa retroalimentación directa de aquellos que probablemente serían usuarios finales del sistema en su implementación completa. La información que se presenta a continuación se considera esencial para realizar ajustes específicos y garantizar que la aplicación no solo cumpla con las expectativas, sino también se alinee de manera óptima con las necesidades y requerimientos reales de los usuarios

Las preguntas fueron las siguientes:

1. ¿Habías utilizado un TMS antes?
2. ¿Consideras que cumple con las necesidades de la empresa?
3. ¿Consideras que es posible adecuar el sistema a la estructura de la empresa?
4. ¿Fue más fácil de usar que los formatos de Excel?
5. ¿El sistema proporciona las funciones de control de acceso que necesitas para proteger la información sensible?
6. ¿Existen características adicionales que crees que serían beneficiosas para mejorar la productividad de la empresa?
7. En caso de que si ¿Cuáles?
8. Te has visto en la necesidad de recurrir a la documentación del TMS

9. ¿Considera que el proporcionar un log de los viajes ayuda a mantener seguro tanto al operador que maneja la unidad como a la empresa?
10. Por favor justifica tu respuesta anterior:
11. ¿Cómo describirías la facilidad de uso del TMS en términos generales?
12. ¿Cómo describirías la facilidad de uso del TMS en términos generales?
13. ¿Consideras que la forma en la que se presentan los datos permite que sean fácil de procesar?
14. ¿Consideras que las funciones como exportar PDF , Excel o importar archivos XML son útiles para la funcionalidad del sistema?

Las respuestas de la pregunta 1 ([imagen 33](#)) indican la realidad de que hoy en día aún se continua con el uso de software de paquetería como lo es Excel para administrar empresas de autotransporte generando que ineficiencia y grietas en la seguridad de la empresa:

Habías utilizado un TMS antes?  
8 respuestas

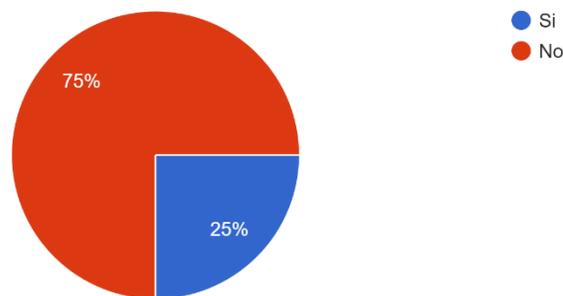


Imagen 33: Gráfico de la pregunta 1

En la siguiente pregunta, la pregunta número 2 ([Imagen 34](#)) se puede observar que la mayoría del grupo de prueba considera que un sistema como este cumple con las necesidades actuales de una empresa de autotransporte:

Consideras que cumple con las necesidades de la empresa?

8 respuestas

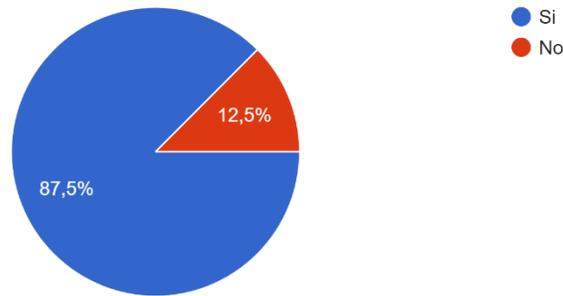


Imagen 34: Gráfico de la pregunta 2.

Sin embargo, a continuación, en la pregunta 3 (Imagen 35) se puede observar que cerca del 37% del grupo de prueba está en desacuerdo con la posibilidad de adecuación del sistema dentro de la empresa, por lo que sin duda este punto presenta una problemática con una necesidad replantear la implementación de este sistema y una mayor investigación acerca de las causas de la problemática:

Consideras que es posible adecuar el sistema a la estructura de la empresa?

8 respuestas

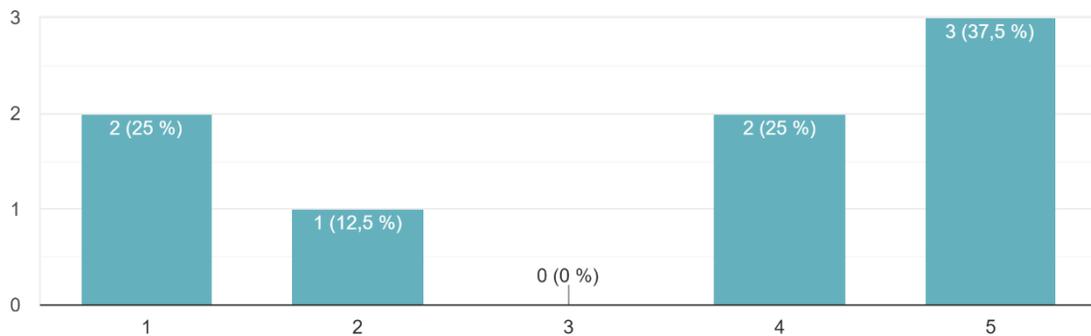


Imagen 35: Gráfico de la pregunta 3.

Un aspecto positivo es que de acuerdo a la pregunta 4 (Imagen 36) se logró un porcentaje de aprobación del 75% sobre la facilidad de su uso ante programas como Excel:

Fue más fácil de usar que los formatos de Excel?

8 respuestas

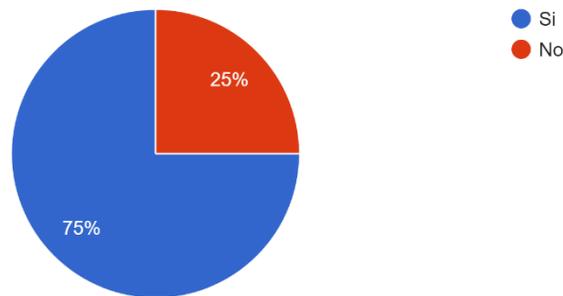


Imagen 36: Gráfico de la pregunta 4.

Aunque también genera un sentimiento de seguridad con los usuarios como se puede ver en la siguiente pregunta (Imagen 37):

El sistema proporciona las funciones de control de acceso que necesitas para proteger la información sensible?

8 respuestas

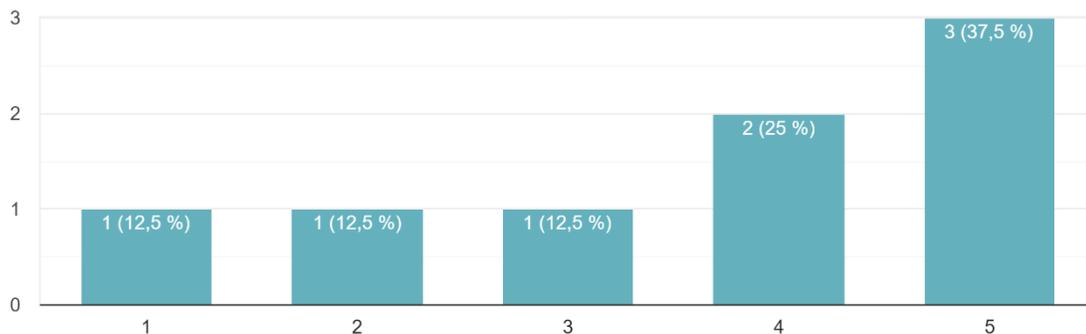


Imagen 37: Gráfico de la pregunta 5.

A pesar de que en la siguiente pregunta (Imagen 38) se consiguió un 75% de respuestas que indican que el sistema cumple con las características beneficiosas para la empresa, un 25% indica que existen otras características que podrían resultar útiles, sin embargo, no proporcionaron cuales:

Existen características adicionales que crees que serían beneficiosas para mejorar la productividad de la empresa?

8 respuestas

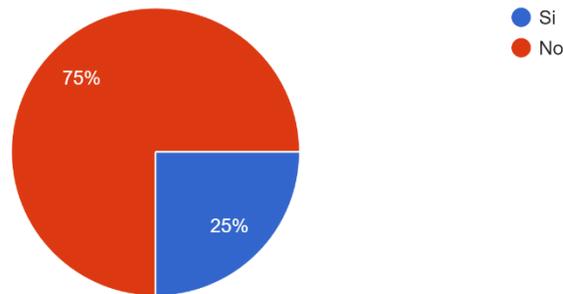


Imagen 38: Gráfico de la pregunta 6.

Las respuestas a la pregunta 8 (Imagen 39) indica que el sistema presenta una facilidad de uso debido a que la mayoría no necesita la documentación al momento de usar por primera vez el sistema:

Te has visto en la necesidad de recurrir a la documentación del TMS

8 respuestas

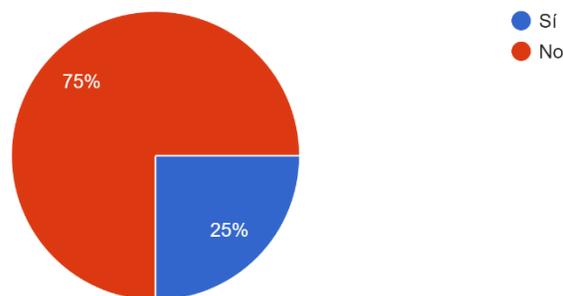


Imagen 39: Gráfico de la pregunta 8.

Los resultados a la pregunta 9 (Imagen 40) muestran que el sistema genera un sentimiento de confianza dentro del grupo de prueba respecto a los logs planteados para el desarrollo del sistema:

Considera que el proporcionar un log de los viajes ayuda a mantener seguro tanto al operador que maneja la unidad como a la empresa?

8 respuestas

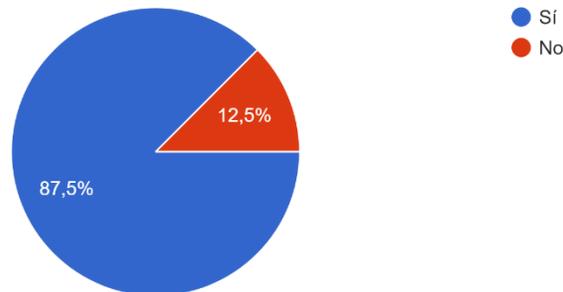


Imagen 40: Gráfico de la pregunta 9.

Estas fueron sus razones a la pregunta anterior:

- Para prevenir casos donde se pone en riesgo la integridad de ambos
- Para mantener un registro que puede servir en caso de incidencias
- Proporciona viabilidad ante cualquier circunstancia.
- Es muy funcional
- Lo hace más fácil
- Si, porque ayuda con la inseguridad
- Si debido que así se tiene mayor información acerca de las ocurrencias del viaje

Se realizaron una serie de preguntas relacionadas a la facilidad de uso del sistema por parte de los usuarios la primera fue la pregunta 11 que preguntó su facilidad de uso general (Imagen 41):

Cómo describirías la facilidad de uso del TMS en términos generales?

8 respuestas

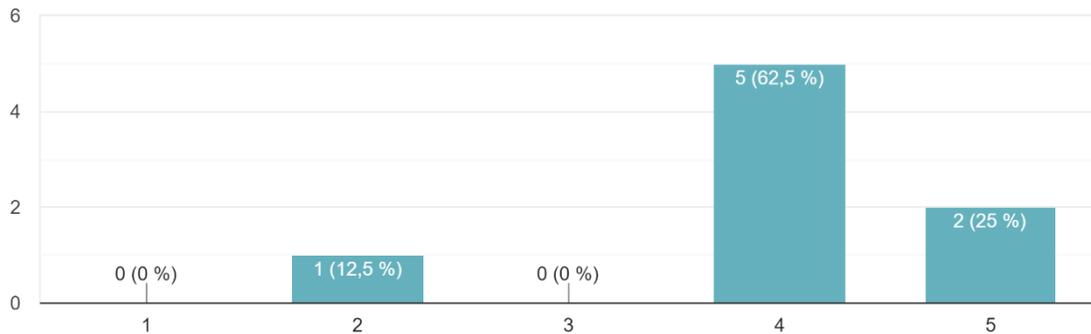


Imagen 41: Gráfico de la pregunta 11.

Posteriormente la pregunta 12 muestra la opinión de los usuarios acerca de la navegación dentro del sistema para desplazarse de una sección a otra (Imagen 42):

Consideras que la Interfaz de Usuario es fácil de navegar?

8 respuestas

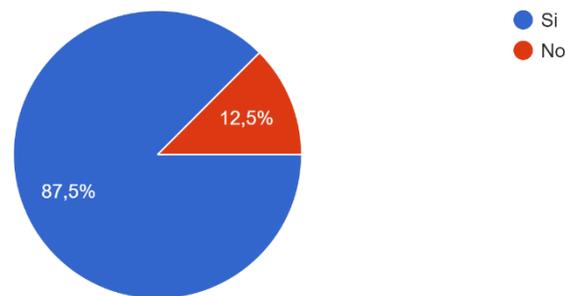


Imagen 42: Gráfico de la pregunta 12.

La pregunta 13 indica una alta favorabilidad por parte de los usuarios ante la distribución de datos dentro del sistema para su procesamiento (Imagen 43):

Consideras que la forma en la que se presentan los datos permite que sean fácil de procesar?  
8 respuestas

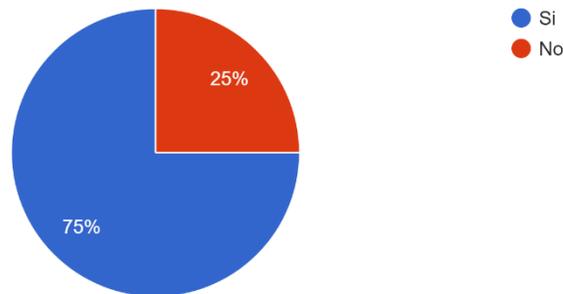


Imagen 43: Gráfico de la pregunta 13.

Por último, en la pregunta 14 (Imagen 44), algunas de las funciones que se encuentran activas dentro del presente sistema fueron vistas como útiles por casi el 90% de los entrevistados:

Consideras que las funciones como exportar a pdf, excel o importar archivos xml son útiles para la funcionalidad del sistema?  
8 respuestas

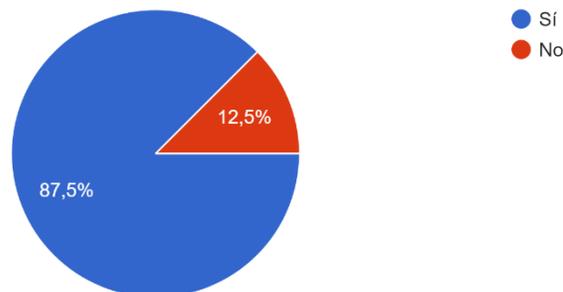


Imagen 44: Gráfico de la pregunta 14.

Mediante las evaluaciones se logró comprobar la funcionalidad percibida del sistema que de acuerdo a los usuarios cumple con los pilares de un TMS demostrando un desempeño eficiente de procesos logísticos, en parte gracias a la visibilidad en tiempo real de todos los datos eliminando errores comunes y permitiendo rastrear todos los pasos de un viaje desde el mantenimiento de la unidad hasta la entrega de un viaje junto con su facturación.

### Anexo 3.- Carta reconocimiento.



Canace Transportes SA de CV.  
Carretera Jaltepec Tulancingo  
Tulancingo, México  
Teléfono: 775 755 5513

A quien corresponda:

Por medio de la presente, me permito validar y reconocer el proyecto titulado "Sistema Administrativo de Autotransporte de Carga (TMS)" presentado por José Andrés Cruz Martínez, estudiante de la Licenciatura en Ciencias Computacionales con ID 356215 de la Universidad Autónoma del Estado de Hidalgo.

En Canace Transportes SA de CV, hemos revisado y evaluado dicho proyecto, el cual aborda de manera exhaustiva y profesional los aspectos críticos y operativos de un TMS. La investigación de José Andrés Cruz Martínez demuestra un profundo entendimiento de las necesidades y desafíos de la industria del transporte de carga, así como soluciones tecnológicas avanzadas para mejorar la eficiencia y eficacia en la gestión de flotas y operaciones logísticas.

Consideramos que el proyecto de José Andrés aporta valiosos conocimientos y propuestas innovadoras aplicables en el ámbito real del transporte de carga. Su trabajo es pertinente y relevante para el desarrollo de mejores prácticas en la gestión administrativa de autotransportes, y refleja un nivel de calidad académica y rigor investigativo digno de reconocimiento.

Por lo tanto, en Canace Transportes SA de CV expresamos nuestro respaldo y validación al proyecto de José Andrés Cruz Martínez, convencidos de que sus aportaciones serán de gran utilidad tanto para la comunidad académica como para el sector empresarial del transporte de carga.

Sin más por el momento, quedamos a su disposición para cualquier consulta adicional y extendemos nuestras más sinceras felicitaciones a José Andrés Cruz Martínez por su destacada labor y contribución significativa al campo de la gestión de autotransportes.

Atentamente,

  
\_\_\_\_\_  
Lic. Zeferino Cruz Rojas

## Referencias bibliográficas.

- A. Lushi, D. D. (2022). IoT-Based Public Transport Management System. *2022 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*. Egipto: IEEE. <https://doi.org/10.1109/GCAIoT57150.2022.10019029>
- Amazon Web Services. (7 de Junio de 2024). *AWS. ¿Cuál es la diferencia entre HTTP y HTTPS?*. <https://aws.amazon.com/es/compare/the-difference-between-https-and-http/#:~:text=Los%20mensajes%20HTTP%20son%20de,los%20datos%20de%20forma%20cifrada.>
- Amazon Web Services, Inc. (12 de Junio de 2024). *¿Qué es Python? - Explicación del lenguaje Python*. AWS. <https://aws.amazon.com/es/what-is/python/>
- beetrack. (17 de Julio de 2024). *Evolución del transporte logístico y futuro del sector*. beetrack. <https://www.beetrack.com/es/blog/evolucion-del-transporte-logistico>
- concepto. (13 de Febrero de 2024). *¿Qué es el transporte de carga?* concepto. <https://concepto.de/transporte-de-carga/>
- Dispatch Science. (28 de Febrero de 2023). *Dispatch Science | Get a demo and supercharge your business*. Dispatch Science. [https://www.dispatchscience.com/get-a-demo/?Utm\\_Capterra=GetApp](https://www.dispatchscience.com/get-a-demo/?Utm_Capterra=GetApp)
- Flotas GYT. (12 de Febrero de 2022). *Sistema de Gestión de Transporte TMS - 10 principales componentes*. GYT Flotas. <https://www.gytflotas.com/principales-componentes-de-un-buen-sistema-de-gestion-de-transporte-tms/>
- Gregorioiedge. (29 de Abril de 2019). *IEDGE - Sistemas para la gestión de la cadena de suministro*. IEDGE Business School. <https://www.iedge.eu/gregorio-cortes-sistemas-para-la-gestion-de-la-cadena-de-suministro>
- Huet, P. (18 de Abril de 2023). *Python: principales características*. OpenWebinars.net. <https://openwebinars.net/blog/python-principales-caracteristicas/>
- INEGI. (Diciembre de 2021). *Conociendo La Industria del Autotransporte de Carga*. INEGI. [https://www.inegi.org.mx/contenidos/productos/prod\\_serv/contenidos/espanol/bvin/egi/productos/nueva\\_estruc/889463903994.pdf](https://www.inegi.org.mx/contenidos/productos/prod_serv/contenidos/espanol/bvin/egi/productos/nueva_estruc/889463903994.pdf)
- IONOS, E. e. (2 de Noviembre de 2022). *Flask vs. Django: una comparativa de los frameworks de Python*. IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/flask-vs-django/>

- IONOS, E. e. (1 de Marzo de 2023). *¿Qué es Flask Python? un breve tutorial sobre este microframework*. IONOS Digital Guide. [¿Qué es Flask Python? un breve tutorial sobre este microframework](#)
- IONOS, E. e. (18 de Enero de 2023). *¿Qué es MySQL?* IONOS Digital Guide. <https://www.ionos.mx/digitalguide/servidores/know-how/que-es-mysql/>
- Issuu. (16 de Diciembre de 2021). *Conociendo la industria del autotransporte de carga*. Issuu. [https://issuu.com/canacarmx/docs/transporte\\_de\\_carga\\_13\\_dic](https://issuu.com/canacarmx/docs/transporte_de_carga_13_dic)
- Java. (16 de Diciembre de 2003). *WayBackMachine*. Distributed Application Architecture. <https://web.archive.org/web/20110406121920/http://java.sun.com/developer/Books/jdbc/ch07.pdf>
- Kuebix Transportation Management System. (10 de Noviembre de 2023). *Industry's ultimate TMS*. Kuebix. <https://www.kuebix.com/>
- Loogistas. (9 de Noviembre de 2023). *Freight forwarding software for logistics companies | Logistaas*. Logistaas - Soluciones de software de transporte de mercancías y envíos. <https://logistaas.com/es/>
- María Rosa López Mejía, S. M. (Noviembre de 2010). *Biblioteca digital de la UNAM*. Investigación y Ciencia. <https://biblat.unam.mx/hevila/InvestigacionycienciaUniversidadautonomadeaguascalientes/2010/vol18/no47/6.pdf>
- MDN. (18 de Julio de 2023). *Conceptos básicos de HTML - Aprende Desarrollo Web*. MDN Web Docs. [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics)
- MDN. (20 de Mayo de 2024). *¿Qué es JavaScript? - Aprende Desarrollo web*. MDN Web Docs. [https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- MDN. (10 de Julio de 2024). *CSS*. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/CSS>
- MDN. (13 de Febrero de 2024). *HTML: Lenguaje de etiquetas de hipertexto*. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTML>
- MercuryGate International. (18 de Mayo de 2021). *MercuryGate TMS | Soluciones de gestión de transporte*. MercuryGate International. <https://mercurysgate.com/latam/>

- MySQL. (2023). *MySQL Workbench*. MySQL.  
<https://www.mysql.com/products/workbench/>
- Oracle. (7 de Junio de 2024). *¿Qué es un sistema de gestión del transporte?* Oracle.  
<https://www.oracle.com/mx/scm/logistics/transportation-management/what-is-transportation-management-system/>
- Python. (17 de Marzo de 2024). *El tutorial de Python*. Python documentation.  
<https://docs.python.org/es/3/tutorial/>
- Red Hat. (20 de Enero de 2023). *¿Qué es el Internet de las cosas (IoT)?* Red Hat.  
<https://www.redhat.com/es/topics/internet-of-things/what-is-iot>
- Roger S. Pressman, B. R. (2020). *Ingeniería del software*. McGraw-Hill.  
<https://doi.org/9781456284756>
- Visual Studio Code. (3 de Abril de 2023). *Docs - Getting Started*. Visual Studio Code.  
<https://code.visualstudio.com/docs>
- Zúñiga, F. G. (01 de Julio de 2024). *Arsys*. Todo sobre la arquitectura cliente-servidor.  
<https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor>