

UNIVERSIDAD PRIVADA DOMINGO SAVIO



PROYECTO SOCIOFORMATIVO LOGISTICA DE TRANSPORTE OPTIMIZANDO COSTOS MEDIANTE SISTEMAS DE ECUACIONES

NOMBRE DEL ESTUDIANTE

Roy Danyeri Camacho Torrez

Douglas Flor Hernández

Jorge Luis Justiniano Justiniano

Julio César Sánchez Burgos

DOCENTE

Esther Veizaga Gonzales

Santa Cruz de la Sierra – Bolivia

2025

abril, 2025

INDICE

INTRODUCCIÓN	4
DEDICATORIA	5
AGRADECIMIENTOS	5
CAPITULO I	6
ANTECEDENTES	6
1.1 PLANTEAMIENTO DEL PROBLEMA	6
1.2 JUSTIFICACION	6
1.3 TIPO Y NIVEL DE INVESTIGACION	7
1.4 OBJETIVOS DEL PROYECTO	7
1.4.1 Objetivo General	7
1.4.2 Objetivos específicos	7
CAPITULO II	8
METODOLOGÍA DEL TRABAJO	8
2.1 ORGANIGRAMA DE LA ORGANIZACIÓN GRUPAL	8
2.2 DESCRIPCIÓN DEL ÁMBITO DE LA INVESTIGACIÓN	9
CAPITULO III	10
MARCO CONCEPTUAL	10
3.1 VISUAL STUDIO CODE (VSCODE)	10
3.2 HYPERTEXT MARKUP LANGUAGE	11
3.3 CASCADING STYLE SHEETS	12
3.4 PHP: Lenguaje de Programación para Desarrollo Web	13
3.5 MySQL: Sistema de Gestión de Bases de Datos Relacionales	14
3.6 XAMPP: Entorno de Desarrollo para Servidores Web	15

CAPITULO IV	16
MODELADO DE LOS ARTEFACTOS DEL SISTEMA	16
4.1 LISTA DE REQUERIMIENTOS.....	17
4.2 MODELADO ESTRUCTURAL.....	19
4.2.1 Diagrama de clases.....	19
4.2.2 Diagrama de objetos.....	20
4.2.3 Diagrama de paquetes.....	22
4.3 MODELADO DE COMPORTAMIENTO	23
4.3.1 Diagrama de casos de uso	23
4.3.2 Diagrama de secuencia	26
CAPITULO V.....	28
DESARROLLO DE LA SISTEMA FUNCIONAL	28
CAPITULO VI	53
CONCLUSIONES DEL PROYECTO.....	53
CAPITULO VII.....	54
RECOMENDACIONES GRUPALES	54
CAPITULO VIII	55
REFERENCIAS BIBLIOGRAFICAS.....	55

INTRODUCCIÓN

El desarrollo de modelos de optimización logística ha evolucionado significativamente en respuesta a las crecientes necesidades de eficiencia en el transporte de productos. Desde los primeros métodos manuales hasta las soluciones matemáticas actuales, la tecnología ha permitido optimizar los procesos de distribución, facilitando la minimización de costos y mejorando la eficiencia operativa.

En este contexto, el presente proyecto tiene como objetivo desarrollar un modelo matemático para optimizar las rutas de transporte entre las ciudades de La Paz, Cochabamba y Tarija, enfocándose en minimizar los costos de transporte mediante una distribución eficiente de 1,500 productos.

El modelo propuesto permitirá realizar cálculos precisos para determinar la cantidad óptima de productos a transportar entre cada par de ciudades, considerando factores como distancia y costo por kilómetro. Además, el proyecto demostrará la aplicación de técnicas matemáticas avanzadas en la resolución de problemas logísticos reales.

DEDICATORIA

Este trabajo está dedicado a nuestras familias, quienes nos han brindado constante apoyo y motivación durante nuestra formación académica. Su paciencia, comprensión y amor han sido fundamentales en nuestro proceso educativo.

A nuestros padres, que siempre nos han inspirado a perseguir nuestros sueños y a nunca rendirnos ante los desafíos.

AGRADECIMIENTOS

Expresamos nuestro más sincero agradecimiento:

- Al Ing. **Esther Veizaga Gonzales**, docente de la asignatura Álgebra Lineal, por su compromiso, orientación y apoyo constante durante todo el proceso de desarrollo del proyecto.
- A nuestras familias, por su apoyo incondicional, comprensión y motivación a lo largo de todo el proceso.
- A nuestros compañeros de equipo, por su compromiso, esfuerzo y colaboración en cada fase del proyecto.

ANTECEDENTES

1.1 PLANTEAMIENTO DEL PROBLEMA

El sector logístico requiere soluciones eficientes para la distribución de productos entre múltiples ciudades. Aunque existen métodos tradicionales de planificación de rutas, su desarrollo e implementación suelen estar limitados a grandes empresas, dejando pocas oportunidades para el análisis académico y la experimentación.

Ante esta realidad, se propone el desarrollo de un modelo matemático propio de optimización logística que incluya un método de cálculo para la distribución óptima de productos. A diferencia de las soluciones comerciales, este proyecto se enfocará en la creación de un modelo con código estructurado y documentación detallada, permitiendo a estudiantes y profesionales comprender su funcionamiento y replicarlo en distintos entornos.

1.2 JUSTIFICACION

Este proyecto se justifica desde una doble perspectiva: técnica y académica. Desde el punto de vista técnico, busca desarrollar un modelo matemático que optimice la distribución de productos mediante un método de cálculo preciso y escalable. Esto permitirá la minimización de costos de transporte, reduciendo errores y mejorando la eficiencia operativa.

Desde la perspectiva académica, este desarrollo representa una oportunidad para aplicar conocimientos en modelado matemático, optimización de sistemas y análisis de operaciones. A diferencia de las soluciones comerciales cerradas, este modelo estará acompañado de documentación detallada, facilitando su comprensión y reutilización por parte de otros investigadores y desarrolladores.

1.3 TIPO Y NIVEL DE INVESTIGACION

La investigación se divide en dos enfoques: teórico, que abarcó el estudio de principios matemáticos para optimización de rutas, y experimental, que consistió en la implementación y prueba práctica del modelo.

- **Descriptivo:** Se exploraron las técnicas matemáticas para optimización de rutas de transporte, enfocándose en aspectos como minimización de costos, distribución eficiente de productos y análisis de restricciones.
- **Experimental:** Basado en la investigación teórica, se desarrolló el modelo utilizando herramientas matemáticas. Se implementaron métodos de resolución de sistemas de ecuaciones, realizando pruebas de validación para garantizar la precisión del modelo.

1.4 OBJETIVOS DEL PROYECTO

En esta sección se detallan los objetivos que guían el desarrollo del proyecto. Se presenta el objetivo general que define el propósito principal del proyecto, seguido de los objetivos específicos que abordan las metas concretas y las acciones necesarias para alcanzar el resultado esperado.

1.4.1 Objetivo General

Desarrollar un modelo matemático de optimización para la distribución de productos entre La Paz, Cochabamba y Tarija, garantizando la minimización de costos de transporte y la eficiencia en la distribución de 1,500 unidades.

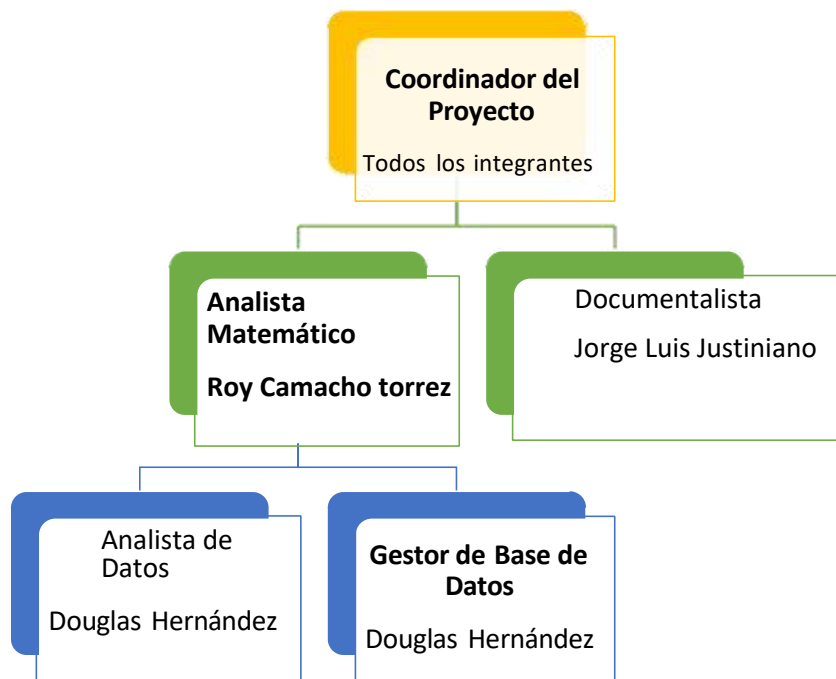
1.4.2 Objetivos específicos

- Modelar matemáticamente las restricciones y variables del problema de distribución de productos.
- Desarrollar un método de cálculo que determine la cantidad óptima de productos a transportar entre cada par de ciudades.
- Implementar un modelo que permita calcular el costo mínimo de transporte.
- Validar el modelo mediante la resolución del sistema de ecuaciones y análisis de resultados.

METODOLOGÍA DEL TRABAJO

Este capítulo presenta la organización del grupo mediante un organigrama visual, detallando la estructura interna de coordinación. Además, describe los entornos claves, como el laboratorio de informática, la biblioteca universitaria y la casa epicentro, que fueron fundamentales para la planificación, el desarrollo y la culminación exitosa del trabajo.

2. ORGANIGRAMA DE LA ORGANIZACIÓN GRUPAL



En la presentación del organigrama de posición organizacional, enfatizamos el importante papel de cada miembro del equipo, reconociendo sus aportes específicos en áreas clave del proyecto. Este enfoque permite una visualización clara de cómo cada participante contribuyó con sus habilidades y conocimientos para el logro exitoso de los objetivos de la investigación. Por otro lado, la transparencia y el detalle no solo resaltan la diversidad de talentos en el equipo, sino que también resaltan la colaboración efectiva que fue clave para el éxito general del proyecto.

2.1 PLANTEAMIENTO DEL PROBLEMA

Tenemos una empresa de logística que necesita transportar productos entre tres ciudades: La Paz, Cochabamba y Tarija. La empresa busca minimizar los costos de transporte, y el costo por kilómetro es de \$25 por producto.

Tenemos las siguientes condiciones:

El total de productos a transportar es de 1500 unidades.

Las distancias entre las ciudades son:

- La Paz → Cochabamba: 610 km
- La Paz → Tarija: 1050 km
- Cochabamba → Tarija: 700 km

El objetivo es determinar cuántos productos deben enviarse directamente entre las ciudades para minimizar el costo total.

Paso 1: Organizamos los datos en forma de sistema de ecuaciones.

Denotamos las cantidades de productos enviados entre las ciudades con las siguientes variables:

- x_1 : Cantidad de productos enviados de La Paz a Cochabamba.
- x_2 : Cantidad de productos enviados de La Paz a Tarija.
- x_3 : Cantidad de productos enviados de Cochabamba a Tarija.

Sabemos que la cantidad total de productos es 1500 unidades, por lo que tenemos la siguiente restricción:

$$x_1 + x_2 + x_3 = 1500$$

Paso 2: Modelar el Costo

Para cada ruta, calculamos el costo por kilómetro multiplicado por la distancia y el número de productos.

Costo LP→CB = 610 km×25 USD =15250 \$ por producto. Entonces, el costo de enviar x_1 productos es $15250 \times x_1$

Costo LP→TA=1050 km×25 USD =26250 \$ por producto. Entonces, el costo de enviar x_2 productos es $26250 \times x_2$

Costo TA→OR=700 km×25 USD=17500 \$ por producto. Entonces, el costo de enviar x_3 productos es $17500 \times x_3$

Paso 3: Plantear el Sistema de Ecuaciones

El sistema de ecuaciones queda planteado de la siguiente manera:

$$x_1 + x_2 + x_3 = 1500$$

$$15250x_1 + 0x_2 + 0x_3 = 15250 \times 1500$$

$$0x_1 + 26250x_2 + 0x_3 = 26250 \times 1500$$

Simplificamos:

$$x_1 + x_2 + x_3 = 1500$$

$$15250x_1 = 22875000$$

$$26250x_2 + 17500x_3 = 39375000$$

**Paso 4: Sistema Matricial**

Representamos el sistema en forma matricial:

$$\begin{bmatrix} 1 & 1 & 1 \\ 15250 & 0 & 0 \\ 0 & 26250 & 17500 \end{bmatrix} \begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix} = \begin{pmatrix} 1500 \\ 22875000 \\ 39375000 \end{pmatrix}$$

La matriz de coeficientes A es:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 15250 & 0 & 0 \\ 0 & 26250 & 17500 \end{bmatrix}$$

El vector de resultados b es:

$$b = \begin{pmatrix} 1500 \\ 22875000 \\ 39375000 \end{pmatrix}$$

Paso 5: Encontrar la Matriz Inversa

Calculamos la matriz inversa de A, denotada A^{-1} :

$$A^{-1} = \begin{bmatrix} 0 & -\frac{1}{15250} & -\frac{1}{15250} \\ \frac{2625}{8750} & \frac{875}{8750} & \frac{1}{8750} \\ \frac{1575}{8750} & \frac{2625}{8750} & \frac{8750}{8750} \end{bmatrix}$$

paso 6: Resolver el Sistema

Multiplicamos la matriz inversa A^{-1} por el vector b:

$$\begin{pmatrix} X1 \\ X2 \\ X3 \end{pmatrix} = A^{-1} \cdot b$$

Hacemos las multiplicaciones:

x1 = 1500 productos van de La Paz a Cochabamba.

x2 = 300 productos van de La Paz a Tarija.

x3 = 1200 productos van de Cochabamba a Tarija.

Interpretación:

Los valores obtenidos minimizan el costo total de transporte:

x1 = 1500 productos van de La Paz a Cochabamba

x2 = 300 productos van de La Paz a Tarija.

x3 = 1200 productos van de Cochabamba a Tarija.

Para minimizar el costo total de transporte:

- Enviarás 1500 productos desde La Paz a Cochabamba.
- Enviarás 300 productos desde La Paz a Tarija.
- Enviarás 1200 productos desde Cochabamba a Tarija.

Esta solución minimiza el costo total del transporte dado que se basa en las distancias y el precio por kilómetro.

2.2 DESCRIPCIÓN DEL ÁMBITO DE LA INVESTIGACIÓN

Durante el desarrollo del proyecto, se identificaron tres entornos clave fundamentales para su implementación:

- **Laboratorio de Informática:** Espacio crucial para la planificación, modelado matemático y desarrollo de cálculos. Aquí se definieron las estrategias de resolución, se implementaron los algoritmos de optimización y se realizaron las pruebas iniciales del modelo.
- **Biblioteca Universitaria:** Entorno esencial para la investigación teórica y el análisis de metodologías de optimización. Se revisaron referencias matemáticas, modelos de optimización de rutas y técnicas de resolución de sistemas de ecuaciones.
- **Espacio de Trabajo Remoto:** El entorno de trabajo individual se convirtió en un espacio clave para la integración final, revisión de cálculos y preparación de la documentación. Aquí se realizaron las pruebas finales, se ajustó el modelo y se preparó la presentación del proyecto.



MARCO CONCEPTUAL

Sinergia entre Ingeniería Industrial e Ingeniería de Sistemas

Optimización de Costos desde la Perspectiva de Ingeniería Industrial

La Ingeniería Industrial juega un papel fundamental en la optimización de procesos logísticos, aportando metodologías y herramientas que permiten:

- **Análisis de Procesos**
 1. Estudio detallado de las rutas de transporte
 2. Identificación de puntos de ineficiencia
 3. Evaluación de costos operativos

- **Técnicas de Optimización**
 1. Modelado matemático de problemas de distribución
 2. Aplicación de métodos de investigación de operaciones
 3. Desarrollo de modelos de minimización de costos

- **Herramientas de Análisis**
 1. Diagramas de flujo de procesos
 2. Análisis de tiempos y movimientos
 3. Técnicas de muestreo y medición de eficiencia

Contribución de Ingeniería de Sistemas en la Automatización

La Ingeniería de Sistemas complementa el proceso de optimización mediante:

- **Automatización de Procesos**
 1. Desarrollo de sistemas de gestión logística
 2. Implementación de algoritmos de optimización
 3. Creación de herramientas de cálculo y simulación
- **Tecnologías de Soporte**
 1. Sistemas de información geográfica (GIS)
 2. Aplicaciones de optimización de rutas
 3. Herramientas de análisis de datos big data
- **Soluciones Tecnológicas**
 1. Desarrollo de software especializado
 2. Implementación de sistemas de seguimiento en tiempo real
 3. Creación de interfaces para toma de decisiones
 - **Metodología Interdisciplinaria de Optimización**
 - *Proceso Integrado*
- **Fase de Análisis Industrial**
 1. Identificación de restricciones
 2. Modelado matemático inicial
 3. Evaluación de parámetros de costos
- **Fase de Desarrollo Tecnológico**
 1. Traducción del modelo matemático a algoritmos
 2. Desarrollo de herramientas de cálculo
 3. Implementación de sistemas de simulación
- **Fase de Implementación y Mejora Continua**
 1. Pruebas y validación del modelo
 2. Ajuste de parámetros
 3. Desarrollo de sistemas de monitoreo y mejora continua
 - **Beneficios de la Colaboración Interdisciplinaria**
- **Optimización Integral**
 1. Enfoque holístico de los problemas
 2. Combinación de rigor matemático y soluciones tecnológicas
- **Reducción de Costos**
 1. Minimización de ineficiencias
 2. Optimización de recursos

3.1 VISUAL STUDIO CODE (VSCODE)

Visual Studio Code, lanzado en 2015 por Microsoft, se ha convertido en uno de los editores de código más utilizados debido a su ligereza, flexibilidad y la enorme comunidad que lo respalda.

VSCoode es un editor de código fuente gratuito y abierto que proporciona soporte para una amplia variedad de lenguajes de programación. Ofrece una interfaz simple pero poderosa para desarrollar sitios web, y su enfoque en la personalización lo hace ideal para proyectos individuales.

Características clave:

- **Extensiones y plugins:** VSCoode permite la instalación de extensiones para lenguajes específicos, herramientas de depuración, integración con bases de datos, y más.
- **Integración con Git:** Permite gestionar versiones de código de manera eficiente, facilitando el seguimiento de los cambios en el proyecto.
- **Personalización de la interfaz:** Los usuarios pueden ajustar la apariencia y funcionalidad del editor según sus necesidades. (Cuadrado, 2022) (reclut, 2022)



3.2 HYPERTEXT MARKUP LANGUAGE

Fue creado por Tim Berners-Lee en 1991 como el lenguaje de marcado estándar para estructurar contenido en la web.

HTML es el lenguaje de marcado utilizado para estructurar páginas web. Permite definir la jerarquía y los elementos dentro de una página, como encabezados, párrafos, enlaces y tablas.

Características clave:

- **Estructuración del contenido:** Utiliza etiquetas para definir elementos como texto, imágenes, formularios, y tablas.
- **Enlaces y navegación:** Permite crear vínculos a otras páginas o sitios web.
- **Multimedia:** Soporta la inserción de contenido multimedia como imágenes, videos y audios.
- **Uso en el proyecto:** HTML fue utilizado para estructurar todo el contenido del sitio web, desde los encabezados y párrafos hasta las tablas y formularios necesarios para la interacción del usuario.

En este proyecto, HTML se utilizó para la estructura y organización del contenido del sistema, permitiendo la correcta visualización de la interfaz de usuario en distintos dispositivos. Gracias a su simplicidad y compatibilidad con otras tecnologías, se logró desarrollar una base sólida para la presentación de la información. (Delgado, 2024) (Vadavo, 2024)



3.3 CASCADING STYLE SHEETS

fue desarrollado en 1996 por Håkon Wium Lie y Bert Bos como una forma de separar el contenido (HTML) de la presentación visual en las páginas web.

CSS es un lenguaje de estilo utilizado para describir la presentación de un documento HTML. Permite controlar el diseño, el color, las fuentes y otros aspectos visuales de una página web.

Características clave:

- **Diseño visual:** Permite definir el aspecto de los elementos HTML, como el tamaño, color, y disposición.
- **Estilos responsivos:** Con CSS, se puede crear un diseño que se adapte a diferentes tamaños de pantalla utilizando técnicas como los medios de consulta (media queries).
- **Animaciones y transiciones:** Permite agregar efectos visuales para mejorar la experiencia del usuario.

En este proyecto, CSS se empleó para el diseño y la estilización de la interfaz, asegurando una experiencia visual atractiva y una disposición adecuada de los elementos en pantalla. Mediante el uso de estilos personalizados y técnicas de diseño responsivo, se garantizó una correcta adaptación del sistema a diferentes dispositivos y tamaños de pantalla. (Code, 2022) (Zermeño, 2022)



3.4 PHP: Lenguaje de Programación para Desarrollo Web

Es un lenguaje de programación de código abierto diseñado para el desarrollo web. Fue creado en 1994 por Rasmus Lerdorf y, desde entonces, se ha convertido en una de las tecnologías más utilizadas para la creación de sitios dinámicos y aplicaciones web.

Este lenguaje se ejecuta en el servidor, lo que permite generar contenido dinámico antes de enviarlo al navegador del usuario. Se integra fácilmente con bases de datos como MySQL, facilitando la creación de aplicaciones interactivas y sistemas de gestión de contenido (CMS).

Características Clave:

- **Código abierto y multiplataforma:** Compatible con sistemas como Windows, Linux y macOS.
- **Ejecución en el servidor:** Procesa el código antes de enviarlo al navegador, permitiendo la generación de contenido dinámico.
- **Interacción con bases de datos:** Soporte para MySQL, PostgreSQL, SQLite, entre otras.
- **Compatibilidad con HTML y JavaScript:** Se puede combinar con tecnologías frontend para mejorar la experiencia del usuario.

En este proyecto, PHP se utilizó para la creación del backend, permitiendo la conexión con la base de datos y la gestión de la lógica del sitio web. Gracias a su flexibilidad y eficiencia, se logró desarrollar un sistema dinámico y funcional. (Maldonado, 2021) (Argenis, 2024)



3.5 MySQL: Sistema de Gestión de Bases de Datos Relacionales

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto que permite el almacenamiento, manipulación y recuperación eficiente de datos estructurados. Fue desarrollado en 1995 por Michael Widenius, David Axmark y Allan Larsson y es ampliamente utilizado en aplicaciones web, sistemas empresariales y plataformas en la nube.

Este sistema es compatible con SQL (Structured Query Language), lo que facilita la creación, modificación y consulta de bases de datos. Gracias a su escalabilidad y rendimiento, MySQL es una opción preferida para sitios web dinámicos y aplicaciones de gran demanda.

Características Clave:

- **Alta velocidad y eficiencia:** Optimizado para manejar grandes volúmenes de datos con un rendimiento superior.
- **Soporte para múltiples motores de almacenamiento:** InnoDB (con soporte para transacciones y claves foráneas), MyISAM (más rápido en lecturas), entre otros.
- **Seguridad y control de acceso:** Permite la gestión de permisos mediante usuarios y autenticación segura.
- **Integración con múltiples lenguajes:** Compatible con PHP, Python, Java, entre otros, facilitando el desarrollo de aplicaciones web.

En este proyecto, MySQL se utilizó para el almacenamiento y gestión de los datos del sistema, permitiendo consultas optimizadas y una integración eficiente con el backend desarrollado en PHP. Su estabilidad y facilidad de uso fueron clave para garantizar el correcto funcionamiento de la base de datos. (Robledano, openwebinars, 2019) (Luna, 2024)



3.6 XAMPP: Entorno de Desarrollo para Servidores Web

Es un paquete de software de código abierto que proporciona un entorno de desarrollo completo para la creación y prueba de aplicaciones web. Fue desarrollado por Apache Friends y está disponible para Windows, Linux y macOS.

El nombre XAMPP proviene de la combinación de los principales componentes que incluye:

- **X:** Multiplataforma (compatible con varios sistemas operativos).
- **A:** Apache, el servidor web más utilizado.
- **M:** MySQL o MariaDB, sistemas de gestión de bases de datos.
- **P:** PHP, lenguaje de programación para desarrollo web.
- **P:** Perl, otro lenguaje de scripting compatible.

Características Clave:

- **Fácil instalación y configuración:** Permite desplegar rápidamente un entorno de desarrollo.
- **Incluye Apache, MySQL/MariaDB, PHP y Perl:** Todo lo necesario para ejecutar aplicaciones web.
- **Panel de control intuitivo:** Facilita la gestión de servicios como el servidor web y la base de datos.
- **Compatible con phpMyAdmin:** Permite la administración de bases de datos a través de una interfaz gráfica.
- **Útil para pruebas locales:** Permite desarrollar y probar proyectos antes de su implementación en un servidor en línea.

En este proyecto, XAMPP se utilizó para crear un entorno de desarrollo local, permitiendo la ejecución del servidor web, la base de datos y los scripts en PHP. Gracias a su facilidad de uso, se pudo realizar pruebas y configuraciones sin necesidad de un servidor en producción. (Peña, 2022) (Cyberstream, 2024)



MODELADO DE LOS ARTEFACTOS DEL SISTEMA

Este capítulo aborda el desarrollo del modelo estructural y de comportamiento UML, permitiendo una representación clara y detallada de la arquitectura y funcionalidad del sistema. Se incluyen diagramas que describen tanto la estructura del sistema como la interacción entre los diferentes elementos.

Gran parte del desarrollo de este capítulo se realizó utilizando Draw.io, una herramienta en línea que permite la creación de diagramas de manera intuitiva y colaborativa. Se utilizó para diseñar el modelo estructural UML, que incluye el diagrama de clases, diagrama de objetos y diagrama de paquetes, facilitando la visualización de la estructura y organización del sistema. Además, se elaboró el modelado de comportamiento, representado por el diagrama de casos de uso y el diagrama de secuencia, los cuales describen la interacción de los usuarios con el sistema y el flujo de ejecución de los procesos principales.

Draw.io fue elegida por su interfaz intuitiva, la posibilidad de trabajo colaborativo en tiempo real y su capacidad para generar diagramas detallados con facilidad. Esto permitió organizar de manera clara los elementos del sistema, asegurando un diseño preciso y bien documentado.

A continuación, se adjunta el logo de la herramienta Draw.io, la cual fue utilizada en el desarrollo de este capítulo para crear los diagramas y modelos relacionados con la estructura y funcionamiento del sistema.



Para finalizar, se deja el enlace de la herramienta Draw.io para que pueda darle un vistazo y utilizarla según sus necesidades: <https://app.diagrams.net/>

CONCLUSIONES DEL PROYECTO

La colaboración entre Ingeniería Industrial e Ingeniería de Sistemas ofrece una solución integral para problemas de optimización logística. Mediante la combinación de análisis matemático, técnicas de optimización y herramientas tecnológicas avanzadas, es posible desarrollar soluciones más eficientes, precisas y adaptables.

El modelo presentado demuestra el potencial de un enfoque interdisciplinario, donde las fortalezas de cada disciplina se complementan para crear soluciones innovadoras y de alto impacto en la optimización de procesos logísticos.

RECOMENDACIONES GRUPALES

Durante el desarrollo de este proyecto, hemos adquirido valiosas lecciones que pueden servir como guía para aquellos que deseen iniciarse en el área de desarrollo o replicar proyectos similares. A continuación, compartimos nuestras recomendaciones basadas en nuestros aciertos y errores:

- **Planificación Estratégica:** Es esencial establecer objetivos claros, asignar tareas de manera equitativa y fijar un cronograma realista. Esto ayuda a gestionar el tiempo y los recursos de forma eficiente, reduciendo posibles retrasos y asegurando que cada fase del proyecto se complete con éxito.
 - **Dominio de Herramientas Tecnológicas:** Es importante tener un buen manejo de herramientas como HTML y CSS. Aunque no utilizamos frameworks, conocer plataformas como Bootstrap y Angular puede ser muy útil. Practicar con ejemplos sencillos acelera el aprendizaje y mejora la calidad del trabajo, evitando errores en la implementación.
 - **Pruebas y Optimización:** Realizar pruebas de forma regular y mejorar el código, la velocidad de carga y la compatibilidad con diversos dispositivos es clave para ofrecer una experiencia de usuario fluida, permitiendo detectar problemas antes de que se conviertan en dificultades más grandes.
 - **Documentación Detallada:** Mantener una documentación exhaustiva sobre decisiones tomadas, configuraciones y cambios realizados facilita la solución de problemas y permite la integración de nuevos desarrolladores, mejorando la capacidad de mantenimiento y futuras actualizaciones del sistema.
 - **Modelado UML:** Aunque al principio puede parecer innecesario o complicado, el modelado UML (estructural y de comportamiento) es vital para orientar el desarrollo del proyecto. Los diagramas proporcionan claridad y coherencia entre los diferentes módulos, evitando errores y facilitando la resolución de problemas a largo plazo.
 - **Iteración y Retroalimentación:** Adoptar un enfoque flexible y ajustar el proyecto según la retroalimentación recibida permite identificar áreas de mejora y hacer ajustes antes de la finalización, asegurando que el producto cumpla con las expectativas y requisitos establecidos.
- Al aplicar estas recomendaciones, el proceso de desarrollo será más ágil, eficiente y exitoso en futuros sistemas web u otros proyectos de desarrollo.

REFERENCIAS BIBLIOGRÁFICAS

- Hillier, F. S., & Lieberman, G. J. (2010). Investigación de Operaciones. McGraw-Hill.
- Winston, W. L. (2004). Investigación de Operaciones: Aplicaciones y Algoritmos. Thomson.
- Taha, H. A. (2012). Investigación de Operaciones. Pearson Educación.
- Baker, K. R. (1995). Optimization Modeling with Spreadsheets. South-Western College Publishing.
- Argenis. (2024, Abril 05). *webempresa*. Obtenido de PHP, ¿Qué es y cómo funciona?: <https://www.webempresa.com/blog/php-que-es-y-como-funciona.html>
- Code, L. (2022, Junio 22). *dev.to*. Obtenido de La Cascada en CSS: Qué es y cómo funciona: <https://dev.to/lupitacode/la-cascada-en-css-que-es-y-como-funciona-31cd>
- Cuadrado, G. C. (2022, Julio 22). *openwebinars*. Obtenido de Visual Studio Code: Editor de código para desarrolladores: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Cyberstream. (2024, Mayo 30). *byronvargas*. Obtenido de XAMPP: Guía completa sobre qué es y para qué sirve: <https://www.byronvargas.com/web/que-es-y-para-que-sirve-el-xampp/>
- Delgado, H. (2024, Abril 25). *disenowebakus*. Obtenido de Historia de HTML - Origen y evolución del hipertexto Web: <https://disenowebakus.net/historia-html.php>
- Luna, J. C. (2024, Septiembre 11). *datacamp*. Obtenido de Tutorial de MySQL: Guía completa para principiantes: <https://www.datacamp.com/es/tutorial/my-sql-tutorial>
- Maldonado, J. (2021, Junio 01). *joystick*. Obtenido de Historia de PHP sus ventajas, desventajas y para que se usa actualmente.: <https://www.joystick.com.mx/historia-de-php-y-sus-ventajas-y-desventajas-y-para-que-se-usa-actualmente/>
- Peña, V. (2022, Julio 19). *norvicsoftware*. Obtenido de ¿Qué es Xampp?: <https://norvicsoftware.com/que-es-xampp/>
- recluit. (2022, Abril 12). *recluit.com*. Obtenido de ¿Qué es Visual Studio Code?: <https://recluit.com/que-es-visual-studio-code/>
- Robledano, A. (2019, Diciembre 24). *openwebinars*. Obtenido de Qué es MySQL: Características y ventajas: <https://openwebinars.net/blog/que-es-mysql/>
- Robledano, A. (2020, Mayo 14). *openwebinars*. Obtenido de Qué es Github: <https://openwebinars.net/blog/que-es-github/>
- Saavedra, J. A. (2023, Junio 04). *ebac*. Obtenido de Qué es Github y para qué sirve: una guía para principiantes: <https://ebac.mx/blog/que-es-github>
- vadavo. (2024, Noviembre 18). *vadavo.com*. Obtenido de Conceptos básicos de HTML: Guía para principiantes: <https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/>
- Zermeño, R. M. (2022, Agosto 26). *azulschool.net*. Obtenido de CSS, historia de los estilos y el diseño web.: <https://azulschool.net/css-historia-de-los-estilos-y-el-diseno-web/>

SCRIPTS DE LA BASES DE DATOS

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 05-04-2025 a las 02:48:13
-- Versión del servidor: 10.4.32-MariaDB
-- Versión de PHP: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `transporte_optimizacion`
--

--
-- Estructura de tabla para la tabla `ciudades`
--

CREATE TABLE `ciudades` (
  `id` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `ciudades`
--

INSERT INTO `ciudades` (`id`, `nombre`, `created_at`) VALUES
(1, 'La Paz', '2025-04-04 04:21:22'),
(2, 'Cochabamba', '2025-04-04 04:21:22'),
(3, 'Tarija', '2025-04-04 04:21:22');
```



```

-----

--
-- Estructura de tabla para la tabla `optimizaciones`
--

CREATE TABLE `optimizaciones` (
  `id` int(11) NOT NULL,
  `total_productos` int(11) NOT NULL,
  `fecha_calculo` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `optimizaciones`
--

INSERT INTO `optimizaciones` (`id`, `total_productos`, `fecha_calculo`) VALUES
(1, 1000, '2025-04-04 05:24:02'),
(2, 500, '2025-04-04 05:26:45'),
(3, 1500, '2025-04-04 05:34:46'),
(4, 100, '2025-04-04 05:37:43'),
(5, 1000, '2025-04-04 05:59:47'),
(6, 3000, '2025-04-04 06:02:20'),
(7, 1500, '2025-04-04 10:31:21'),
(8, 2000, '2025-04-04 11:45:29'),
(9, 500, '2025-04-04 11:58:04'),
(10, 1000, '2025-04-04 12:46:40');

-----

--
-- Estructura de tabla para la tabla `resultados_optimizacion`
--

CREATE TABLE `resultados_optimizacion` (
  `id` int(11) NOT NULL,
  `optimizacion_id` int(11) NOT NULL,
  `ruta_id` int(11) NOT NULL,
  `cantidad_productos` int(11) NOT NULL,
  `costo_total` decimal(15,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `resultados_optimizacion`
--

INSERT INTO `resultados_optimizacion` (`id`, `optimizacion_id`, `ruta_id`, `cantidad_productos`,
`costo_total`) VALUES
(1, 1, 1, 1000, 15250000.00),
(2, 1, 2, 5000, 131250000.00),
(3, 1, 3, -5000, -87500000.00),
(4, 2, 1, 500, 7625000.00),
(5, 2, 2, 2500, 65625000.00),
(6, 2, 3, -2500, -43750000.00),

```

```
(7, 3, 1, 1500, 22875000.00),
(8, 3, 2, 7500, 196875000.00),
(9, 3, 3, -7500, -131250000.00),
(10, 4, 1, 100, 1525000.00),
(11, 4, 2, 500, 13125000.00),
(12, 4, 3, -500, -8750000.00),
(13, 5, 1, 1000, 15250000.00),
(14, 5, 1, 0, 0.00),
(15, 5, 2, 0, 0.00),
(16, 6, 1, 3000, 45750000.00),
(17, 6, 1, 0, 0.00),
(18, 6, 2, 0, 0.00),
(19, 7, 1, 1500, 22875000.00),
(20, 7, 1, 0, 0.00),
(21, 7, 2, 0, 0.00),
(22, 8, 1, 2000, 30500000.00),
(23, 8, 1, 0, 0.00),
(24, 8, 2, 0, 0.00),
(25, 9, 1, 500, 7625000.00),
(26, 9, 1, 0, 0.00),
(27, 9, 2, 0, 0.00),
(28, 10, 1, 1000, 15250000.00),
(29, 10, 1, 0, 0.00),
(30, 10, 2, 0, 0.00);
```

```
-- -----
```

```
--
```

```
-- Estructura de tabla para la tabla `rutas`
```

```
--
```

```
CREATE TABLE `rutas` (
  `id` int(11) NOT NULL,
  `origen_id` int(11) NOT NULL,
  `destino_id` int(11) NOT NULL,
  `distancia` int(11) NOT NULL,
  `costo_km` decimal(10,2) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--
```

```
-- Volcado de datos para la tabla `rutas`
```

```
--
```

```
INSERT INTO `rutas` (`id`, `origen_id`, `destino_id`, `distancia`, `costo_km`, `created_at`)
VALUES
(1, 1, 2, 610, 25.00, '2025-04-04 04:21:22'),
(2, 1, 3, 1050, 25.00, '2025-04-04 04:21:22'),
(3, 2, 3, 700, 25.00, '2025-04-04 04:21:22');
```

```
--
```

```
-- Índices para tablas volcadas
```

```
--
```

```

--
-- Indices de la tabla `ciudades`
--
ALTER TABLE `ciudades`
  ADD PRIMARY KEY (`id`);

--
-- Indices de la tabla `optimizaciones`
--
ALTER TABLE `optimizaciones`
  ADD PRIMARY KEY (`id`);

--
-- Indices de la tabla `resultados_optimizacion`
--
ALTER TABLE `resultados_optimizacion`
  ADD PRIMARY KEY (`id`),
  ADD KEY `optimizacion_id` (`optimizacion_id`),
  ADD KEY `ruta_id` (`ruta_id`);

--
-- Indices de la tabla `rutas`
--
ALTER TABLE `rutas`
  ADD PRIMARY KEY (`id`),
  ADD KEY `origen_id` (`origen_id`),
  ADD KEY `destino_id` (`destino_id`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `ciudades`
--
ALTER TABLE `ciudades`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT de la tabla `optimizaciones`
--
ALTER TABLE `optimizaciones`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

--
-- AUTO_INCREMENT de la tabla `resultados_optimizacion`
--
ALTER TABLE `resultados_optimizacion`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;

--
-- AUTO_INCREMENT de la tabla `rutas`
--
ALTER TABLE `rutas`

```

```
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `resultados_optimizacion`
--
ALTER TABLE `resultados_optimizacion`
  ADD CONSTRAINT `resultados_optimizacion_ibfk_1` FOREIGN KEY (`optimizacion_id`) REFERENCES `optimizaciones` (`id`),
  ADD CONSTRAINT `resultados_optimizacion_ibfk_2` FOREIGN KEY (`ruta_id`) REFERENCES `rutas` (`id`);

--
-- Filtros para la tabla `rutas`
--
ALTER TABLE `rutas`
  ADD CONSTRAINT `rutas_ibfk_1` FOREIGN KEY (`origen_id`) REFERENCES `ciudades` (`id`),
  ADD CONSTRAINT `rutas_ibfk_2` FOREIGN KEY (`destino_id`) REFERENCES `ciudades` (`id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

CODIGO MODELADO EN MVC

CIUDAD CONTROLADOR

```
<?php
// Usar rutas absolutas completas
$serverRoot = $_SERVER['DOCUMENT_ROOT'];
$projectFolder = '/proyecto_transporte';
$projectRoot = $serverRoot . $projectFolder;

// Definir la ruta base si no está definida
if (!defined('BASE_PATH')) {
    define('BASE_PATH', $projectRoot);
}

require_once $projectRoot . '/models/CiudadModel.php';

class CiudadController {
    private $ciudadModel;

    public function __construct() {
        $this->ciudadModel = new CiudadModel();
    }

    // Mostrar lista de ciudades
    public function index() {
        // Obtener todas las ciudades
        $ciudades = $this->ciudadModel->getAll();

        // Cargar la vista
        include_once BASE_PATH . '/views/layouts/main.php';
        include_once BASE_PATH . '/views/ciudad/index.php';
    }

    // Mostrar formulario para crear nueva ciudad
    public function create() {
        // Cargar la vista del formulario
        include_once BASE_PATH . '/views/layouts/main.php';
        include_once BASE_PATH . '/views/ciudad/create.php';
    }

    // Procesar la creación de una nueva ciudad
    public function store() {
        // Validar datos recibidos del formulario
        if (!isset($_POST['nombre']) || empty($_POST['nombre'])) {
            $_SESSION['error'] = "El nombre de la ciudad es requerido.";
            header("Location: index.php?controller=ciudad&action=create");
            exit;
        }
    }
}
```

```

    }

    // Verificar que no exista una ciudad con el mismo nombre
    $nombre = trim($_POST['nombre']);
    if ($this->ciudadModel->existeCiudad($nombre)) {
        $_SESSION['error'] = "Ya existe una ciudad con este nombre.";
        header("Location: index.php?controller=ciudad&action=create");
        exit;
    }

    // Asignar valores al modelo
    $this->ciudadModel->nombre = $nombre;

    // Guardar la ciudad
    if ($this->ciudadModel->create()) {
        $_SESSION['success'] = "Ciudad creada correctamente.";
        header("Location: index.php?controller=ciudad");
    } else {
        $_SESSION['error'] = "Error al crear la ciudad.";
        header("Location: index.php?controller=ciudad&action=create");
    }
    exit;
}

// Mostrar formulario para editar ciudad
public function edit() {
    if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
        $_SESSION['error'] = "ID de ciudad inválido.";
        header("Location: index.php?controller=ciudad");
        exit;
    }

    $id = $_GET['id'];

    // Obtener datos de la ciudad
    if (!$this->ciudadModel->getById($id)) {
        $_SESSION['error'] = "Ciudad no encontrada.";
        header("Location: index.php?controller=ciudad");
        exit;
    }

    // Preparar datos para la vista
    $ciudad = [
        'id' => $this->ciudadModel->id,
        'nombre' => $this->ciudadModel->nombre
    ];

    // Cargar la vista del formulario
    include_once BASE_PATH . '/views/layouts/main.php';
    include_once BASE_PATH . '/views/ciudad/edit.php';
}

// Procesar la actualización de una ciudad
public function update() {

```

```

// Validar datos recibidos del formulario
if (!isset($_POST['id']) || !is_numeric($_POST['id']) ||
    !isset($_POST['nombre']) || empty($_POST['nombre'])) {
    $_SESSION['error'] = "Todos los campos son requeridos.";
    header("Location: index.php?controller=ciudad");
    exit;
}

$id = $_POST['id'];
$nombre = trim($_POST['nombre']);

// Verificar que no exista otra ciudad con el mismo nombre (excluyendo la actual)
if ($this->ciudadModel->existeCiudadExcluyendo($nombre, $id)) {
    $_SESSION['error'] = "Ya existe otra ciudad con este nombre.";
    header("Location: index.php?controller=ciudad&action=edit&id=$id");
    exit;
}

// Asignar valores al modelo
$this->ciudadModel->id = $id;
$this->ciudadModel->nombre = $nombre;

// Actualizar la ciudad
if ($this->ciudadModel->update()) {
    $_SESSION['success'] = "Ciudad actualizada correctamente.";
    header("Location: index.php?controller=ciudad");
} else {
    $_SESSION['error'] = "Error al actualizar la ciudad.";
    header("Location: index.php?controller=ciudad&action=edit&id=$id");
}
exit;
}

// Eliminar una ciudad
public function delete() {
    if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
        $_SESSION['error'] = "ID de ciudad inválido.";
        header("Location: index.php?controller=ciudad");
        exit;
    }

    $id = $_GET['id'];

    // Verificar si la ciudad está siendo utilizada en rutas
    if ($this->ciudadModel->ciudadEnUso($id)) {
        $_SESSION['error'] = "No se puede eliminar la ciudad porque está siendo utilizada en una o más rutas.";
        header("Location: index.php?controller=ciudad");
        exit;
    }

    // Asignar el ID al modelo
    $this->ciudadModel->id = $id;

```

```
// Eliminar la ciudad
if ($this->ciudadModel->delete()) {
    $_SESSION['success'] = "Ciudad eliminada correctamente.";
} else {
    $_SESSION['error'] = "Error al eliminar la ciudad.";
}

header("Location: index.php?controller=ciudad");
exit;
}
}
?>
```


RUTA CONTROLADOR

```
<?php
// Usar rutas absolutas completas
$serverRoot = $_SERVER['DOCUMENT_ROOT'];
$projectFolder = '/proyecto_transporte';
$projectRoot = $serverRoot . $projectFolder;

// Definir la ruta base si no está definida
if (!defined('BASE_PATH')) {
    define('BASE_PATH', $projectRoot);
}

require_once $projectRoot . '../models/RutaModel.php';
require_once $projectRoot . '../models/CiudadModel.php';

class RutaController {
    private $rutaModel;
    private $ciudadModel;

    public function __construct() {
        $this->rutaModel = new RutaModel();
        $this->ciudadModel = new CiudadModel();
    }

    // Mostrar lista de rutas
    public function index() {
        // Obtener todas las rutas
        $rutas = $this->rutaModel->getAll();

        // Cargar la vista
        include_once BASE_PATH . '/views/layouts/main.php';
        include_once BASE_PATH . '/views/ruta/index.php';
    }

    // Mostrar formulario para crear nueva ruta
    public function create() {
        // Obtener todas las ciudades para los selectores
        $ciudades = $this->ciudadModel->getAll();

        // Cargar la vista del formulario
        include_once BASE_PATH . '/views/layouts/main.php';
        include_once BASE_PATH . '/views/ruta/create.php';
    }

    // Procesar la creación de una nueva ruta
    public function store() {
        // Validar datos recibidos del formulario
        if (!isset($_POST['origen_id']) || !isset($_POST['destino_id']) ||
            !isset($_POST['distancia']) || !isset($_POST['costo_km'])) {
```

```

        $_SESSION['error'] = "Todos los campos son requeridos.";
        header("Location: index.php?controller=ruta&action=create");
        exit;
    }

    // Validar que origen y destino sean diferentes
    if ($_POST['origen_id'] == $_POST['destino_id']) {
        $_SESSION['error'] = "El origen y destino deben ser diferentes.";
        header("Location: index.php?controller=ruta&action=create");
        exit;
    }

    // Validar valores numéricos
    if (!is_numeric($_POST['distancia']) || $_POST['distancia'] <= 0 ||
        !is_numeric($_POST['costo_km']) || $_POST['costo_km'] <= 0) {
        $_SESSION['error'] = "La distancia y el costo por kilómetro deben ser valores
numéricos positivos.";
        header("Location: index.php?controller=ruta&action=create");
        exit;
    }

    // Verificar que no exista una ruta con el mismo origen y destino
    if ($this->rutaModel->existeRuta($_POST['origen_id'], $_POST['destino_id'])) {
        $_SESSION['error'] = "Ya existe una ruta con este origen y destino.";
        header("Location: index.php?controller=ruta&action=create");
        exit;
    }

    // Asignar valores al modelo
    $this->rutaModel->origen_id = $_POST['origen_id'];
    $this->rutaModel->destino_id = $_POST['destino_id'];
    $this->rutaModel->distancia = $_POST['distancia'];
    $this->rutaModel->costo_km = $_POST['costo_km'];

    // Guardar la ruta
    if ($this->rutaModel->create()) {
        $_SESSION['success'] = "Ruta creada correctamente.";
        header("Location: index.php?controller=ruta");
    } else {
        $_SESSION['error'] = "Error al crear la ruta.";
        header("Location: index.php?controller=ruta&action=create");
    }
    exit;
}

// Mostrar formulario para editar ruta
public function edit() {
    if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
        $_SESSION['error'] = "ID de ruta inválido.";
        header("Location: index.php?controller=ruta");
        exit;
    }

    $id = $_GET['id'];

```

```

// Obtener datos de la ruta
if (!$this->rutaModel->getById($id)) {
    $_SESSION['error'] = "Ruta no encontrada.";
    header("Location: index.php?controller=ruta");
    exit;
}

// Obtener todas las ciudades para los selectores
$ciudades = $this->ciudadModel->getAll();

// Preparar datos para la vista
$ruta = [
    'id' => $this->rutaModel->id,
    'origen_id' => $this->rutaModel->origen_id,
    'destino_id' => $this->rutaModel->destino_id,
    'distancia' => $this->rutaModel->distancia,
    'costo_km' => $this->rutaModel->costo_km
];

// Cargar la vista del formulario
include_once BASE_PATH . '/views/layouts/main.php';
include_once BASE_PATH . '/views/ruta/edit.php';
}

// Procesar la actualización de una ruta
public function update() {
    // Validar datos recibidos del formulario
    if (!isset($_POST['id']) || !is_numeric($_POST['id']) ||
        !isset($_POST['origen_id']) || !isset($_POST['destino_id']) ||
        !isset($_POST['distancia']) || !isset($_POST['costo_km'])) {
        $_SESSION['error'] = "Todos los campos son requeridos.";
        header("Location: index.php?controller=ruta");
        exit;
    }

    $id = $_POST['id'];

    // Validar que origen y destino sean diferentes
    if ($_POST['origen_id'] == $_POST['destino_id']) {
        $_SESSION['error'] = "El origen y destino deben ser diferentes.";
        header("Location: index.php?controller=ruta&action=edit&id=$id");
        exit;
    }

    // Validar valores numéricos
    if (!is_numeric($_POST['distancia']) || $_POST['distancia'] <= 0 ||
        !is_numeric($_POST['costo_km']) || $_POST['costo_km'] <= 0) {
        $_SESSION['error'] = "La distancia y el costo por kilómetro deben ser valores
numéricos positivos.";
        header("Location: index.php?controller=ruta&action=edit&id=$id");
        exit;
    }
}

```

```

    // Verificar que no exista otra ruta con el mismo origen y destino (excluyendo la actual)
    if ($this->rutaModel->existeRutaExcluyendo($_POST['origen_id'], $_POST['destino_id'],
$id)) {
        $_SESSION['error'] = "Ya existe otra ruta con este origen y destino.";
        header("Location: index.php?controller=ruta&action=edit&id=$id");
        exit;
    }

    // Asignar valores al modelo
    $this->rutaModel->id = $id;
    $this->rutaModel->origen_id = $_POST['origen_id'];
    $this->rutaModel->destino_id = $_POST['destino_id'];
    $this->rutaModel->distancia = $_POST['distancia'];
    $this->rutaModel->costo_km = $_POST['costo_km'];

    // Actualizar la ruta
    if ($this->rutaModel->update()) {
        $_SESSION['success'] = "Ruta actualizada correctamente.";
        header("Location: index.php?controller=ruta");
    } else {
        $_SESSION['error'] = "Error al actualizar la ruta.";
        header("Location: index.php?controller=ruta&action=edit&id=$id");
    }
    exit;
}

// Eliminar una ruta
public function delete() {
    if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
        $_SESSION['error'] = "ID de ruta inválido.";
        header("Location: index.php?controller=ruta");
        exit;
    }

    $id = $_GET['id'];

    // Asignar el ID al modelo
    $this->rutaModel->id = $id;

    // Eliminar la ruta
    if ($this->rutaModel->delete()) {
        $_SESSION['success'] = "Ruta eliminada correctamente.";
    } else {
        $_SESSION['error'] = "Error al eliminar la ruta.";
    }

    header("Location: index.php?controller=ruta");
    exit;
}
}
?>

```

TRANSPORTE CONTROLADOR

```
<?php
// Definir la ruta base del proyecto
$base_path = dirname(__DIR__);

require_once $base_path . '/models/TransporteModel.php';
require_once $base_path . '/models/CiudadModel.php';
require_once $base_path . '/models/RutaModel.php';

class TransporteController {
    private $transporteModel;
    private $ciudadModel;
    private $rutaModel;
    private $base_path;

    public function __construct() {
        $this->transporteModel = new TransporteModel();
        $this->ciudadModel = new CiudadModel();
        $this->rutaModel = new RutaModel();
        // Usar la constante BASE_PATH definida en index.php
    }
    if (!defined('BASE_PATH')) {
        define('BASE_PATH', dirname(__DIR__));
    }
    $this->base_path = BASE_PATH;
}

// Mostrar página principal de transporte
public function index() {
    $optimizaciones = $this->transporteModel->getAllOptimizaciones();

    include_once $this->base_path . '/views/layouts/main.php';
    include_once $this->base_path . '/views/transporte/index.php';
}

// Mostrar formulario de optimización
public function optimizar() {
    $rutas = $this->transporteModel->getRutasOptimizacion();

    include_once $this->base_path . '/views/layouts/main.php';
    include_once $this->base_path . '/views/transporte/optimizar.php';
}

// Procesar la optimización
public function procesarOptimizacion() {
    $total_productos = isset($_POST['total_productos']) ? intval($_POST['total_productos']) :
0;
```

```

if ($total_productos <= 0) {
    $_SESSION['error'] = "La cantidad de productos debe ser mayor a cero.";
    header("Location: index.php?controller=transporte&action=optimizar");
    exit;
}

try {
    // Realizar cálculo de optimización
    $resultados = $this->transporteModel->optimizarDistribucion($total_productos);

    // Guardar resultados en la base de datos
    $resultados_db = [];
    foreach ($resultados as $resultado) {
        $resultados_db[] = [
            'ruta_id' => $resultado['ruta_id'],
            'cantidad_productos' => $resultado['cantidad_productos'],
            'costo_total' => $resultado['costo_total']
        ];
    }

    $optimizacion_id = $this->transporteModel->saveOptimizacion($total_productos,
$resultados_db);

    // Redirigir a la página de resultados
    header("Location: index.php?controller=transporte&action=resultados&id=" .
$optimizacion_id);
    exit;
} catch (Exception $e) {
    $_SESSION['error'] = "Error al realizar la optimización: " . $e->getMessage();
    header("Location: index.php?controller=transporte&action=optimizar");
    exit;
}
}

// Mostrar resultados de optimización
public function resultados() {
    $optimizacion_id = isset($_GET['id']) ? intval($_GET['id']) : 0;

    if ($optimizacion_id <= 0) {
        $_SESSION['error'] = "ID de optimización inválido.";
        header("Location: index.php?controller=transporte");
        exit;
    }

    $stmt = $this->transporteModel->getResultadosOptimizacion($optimizacion_id);
    $resultados = [];
    $costo_total = 0;

    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        $resultados[] = [
            'id' => $row['id'],
            'ruta_id' => $row['ruta_id'],
            'origen' => $row['origen_nombre'],

```

```
        'destino' => $row['destino_nombre'],
        'distancia' => $row['distancia'],
        'costo_km' => $row['costo_km'],
        'cantidad_productos' => $row['cantidad_productos'],
        'costo_total' => $row['costo_total']
    ];

    $costo_total += $row['costo_total'];
}

include_once $this->base_path . '/views/layouts/main.php';
include_once $this->base_path . '/views/transporte/resultados.php';
}
```

```
// Mostrar historia de optimizaciones
```

```
public function historia() {
    $optimizaciones = $this->transporteModel->getAllOptimizaciones();

    include_once $this->base_path . '/views/layouts/main.php';
    include_once $this->base_path . '/views/transporte/historia.php';
}
```

```
}
?>
```

HOME CONTROLADOR

```
<?php
class HomeController {
    public function index() {
        // Usar la constante BASE_PATH definida en index.php
        if (!defined('BASE_PATH')) {
            define('BASE_PATH', dirname(__DIR__));
        }

        // Renderizar la vista principal usando rutas absolutas
        include_once BASE_PATH . '/views/layouts/main.php';
        include_once BASE_PATH . '/views/home/index.php';

        // Comprobar si existen los archivos antes de incluirlos
        if (file_exists(BASE_PATH . '/views/layouts/footer.php')) {
            include_once BASE_PATH . '/views/layouts/footer.php';
        }

        if (file_exists(BASE_PATH . '/views/layouts/scripts.php')) {
            include_once BASE_PATH . '/views/layouts/scripts.php';
        }

        if (file_exists(BASE_PATH . '/views/layouts/alertas.php')) {
            include_once BASE_PATH . '/views/layouts/alertas.php';
        }
    }
}
?>
```


CIUDADMODELO.PHP

```
<?php
// Usar rutas absolutas completas
$serverRoot = $_SERVER['DOCUMENT_ROOT'];
$projectFolder = '/proyecto_transporte';
$projectRoot = $serverRoot . $projectFolder;

// Definir la ruta base si no está definida
if (!defined('BASE_PATH')) {
    define('BASE_PATH', $projectRoot);
}

require_once $projectRoot . '/config/database.php';

class CiudadModel {
    private $conn;
    private $table_name = "ciudades";

    public $id;
    public $nombre;
    public $created_at;

    public function __construct() {
        $database = new Database();
        $this->conn = $database->getConnection();
    }

    // Obtener todas las ciudades
    public function getAll() {
        $query = "SELECT id, nombre, created_at FROM " . $this->table_name . " ORDER BY nombre";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();

        return $stmt;
    }

    // Obtener una ciudad por su ID
    public function getById($id) {
        $query = "SELECT id, nombre, created_at FROM " . $this->table_name . " WHERE id = ?";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(1, $id);
        $stmt->execute();

        $row = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($row) {
            $this->id = $row['id'];
            $this->nombre = $row['nombre'];
            $this->created_at = $row['created_at'];
        }
    }
}
```

```

        return true;
    }

    return false;
}

// Verificar si existe una ciudad con el mismo nombre
public function existeCiudad($nombre) {
    $query = "SELECT COUNT(*) as count FROM " . $this->table_name . " WHERE nombre = ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $nombre);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    return $row['count'] > 0;
}

// Verificar si existe otra ciudad con el mismo nombre, excluyendo la ciudad actual
public function existeCiudadExcluyendo($nombre, $id) {
    $query = "SELECT COUNT(*) as count FROM " . $this->table_name . " WHERE nombre = ? AND
id != ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $nombre);
    $stmt->bindParam(2, $id);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    return $row['count'] > 0;
}

// Verificar si una ciudad está siendo utilizada en rutas
public function ciudadEnUso($id) {
    $query = "SELECT COUNT(*) as count FROM rutas WHERE origen_id = ? OR destino_id = ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $id);
    $stmt->bindParam(2, $id);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    return $row['count'] > 0;
}

// Crear una nueva ciudad
public function create() {
    $query = "INSERT INTO " . $this->table_name . " (nombre) VALUES (?)";
    $stmt = $this->conn->prepare($query);

    // Sanitizar entrada
    $this->nombre = htmlspecialchars(strip_tags($this->nombre));

    // Bind parametros

```

```

$stmt->bindParam(1, $this->nombre);

if ($stmt->execute()) {
    return true;
}

return false;
}

// Actualizar una ciudad
public function update() {
    $query = "UPDATE " . $this->table_name . " SET nombre = ? WHERE id = ?";
    $stmt = $this->conn->prepare($query);

    // Sanitizar entrada
    $this->nombre = htmlspecialchars(strip_tags($this->nombre));
    $this->id = htmlspecialchars(strip_tags($this->id));

    // Bind parametros
    $stmt->bindParam(1, $this->nombre);
    $stmt->bindParam(2, $this->id);

    if ($stmt->execute()) {
        return true;
    }

    return false;
}

// Eliminar una ciudad
public function delete() {
    $query = "DELETE FROM " . $this->table_name . " WHERE id = ?";
    $stmt = $this->conn->prepare($query);

    // Sanitizar entrada
    $this->id = htmlspecialchars(strip_tags($this->id));

    // Bind parametro
    $stmt->bindParam(1, $this->id);

    if ($stmt->execute()) {
        return true;
    }

    return false;
}
}
?>

```

RUTAMODELO.PHP

```
<?php
// Usar rutas absolutas completas
$serverRoot = $_SERVER['DOCUMENT_ROOT'];
$projectFolder = '/proyecto_transporte';
$projectRoot = $serverRoot . $projectFolder;

// Definir la ruta base si no está definida
if (!defined('BASE_PATH')) {
    define('BASE_PATH', $projectRoot);
}

require_once $projectRoot . '/config/database.php';

class RutaModel {
    private $conn;
    private $table_name = "rutas";

    public $id;
    public $origen_id;
    public $destino_id;
    public $distancia;
    public $costo_km;
    public $created_at;

    // Propiedades para JOIN
    public $origen_nombre;
    public $destino_nombre;

    public function __construct() {
        $database = new Database();
        $this->conn = $database->getConnection();
    }

    // Obtener todas las rutas con nombres de ciudades
    public function getAll() {
        $query = "SELECT r.id, r.origen_id, r.destino_id, r.distancia, r.costo_km, r.created_at,
                        o.nombre as origen_nombre, d.nombre as destino_nombre
                FROM " . $this->table_name . " r
                LEFT JOIN ciudades o ON r.origen_id = o.id
                LEFT JOIN ciudades d ON r.destino_id = d.id
                ORDER BY r.id";

        $stmt = $this->conn->prepare($query);
        $stmt->execute();

        return $stmt;
    }
}
```

```

// Obtener una ruta por su ID
public function getById($id) {
    $query = "SELECT r.id, r.origen_id, r.destino_id, r.distancia, r.costo_km, r.created_at,
                    o.nombre as origen_nombre, d.nombre as destino_nombre
                FROM " . $this->table_name . " r
                LEFT JOIN ciudades o ON r.origen_id = o.id
                LEFT JOIN ciudades d ON r.destino_id = d.id
                WHERE r.id = ?";

    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $id);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($row) {
        $this->id = $row['id'];
        $this->origen_id = $row['origen_id'];
        $this->destino_id = $row['destino_id'];
        $this->distancia = $row['distancia'];
        $this->costo_km = $row['costo_km'];
        $this->created_at = $row['created_at'];
        $this->origen_nombre = $row['origen_nombre'];
        $this->destino_nombre = $row['destino_nombre'];
        return true;
    }

    return false;
}

// Obtener ruta por origen y destino
public function getByOrigenDestino($origen_id, $destino_id) {
    $query = "SELECT id, origen_id, destino_id, distancia, costo_km, created_at
                FROM " . $this->table_name . "
                WHERE origen_id = ? AND destino_id = ?";

    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $origen_id);
    $stmt->bindParam(2, $destino_id);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($row) {
        $this->id = $row['id'];
        $this->origen_id = $row['origen_id'];
        $this->destino_id = $row['destino_id'];
        $this->distancia = $row['distancia'];
        $this->costo_km = $row['costo_km'];
        $this->created_at = $row['created_at'];
        return true;
    }

    return false;
}

```

```

}

// Verificar si existe una ruta con el mismo origen y destino
public function existeRuta($origen_id, $destino_id) {
    $query = "SELECT COUNT(*) as count FROM " . $this->table_name . "
        WHERE origen_id = ? AND destino_id = ?";

    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $origen_id);
    $stmt->bindParam(2, $destino_id);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    return $row['count'] > 0;
}

// Verificar si existe otra ruta con el mismo origen y destino, excluyendo la ruta actual
public function existeRutaExcluyendo($origen_id, $destino_id, $id) {
    $query = "SELECT COUNT(*) as count FROM " . $this->table_name . "
        WHERE origen_id = ? AND destino_id = ? AND id != ?";

    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $origen_id);
    $stmt->bindParam(2, $destino_id);
    $stmt->bindParam(3, $id);
    $stmt->execute();

    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    return $row['count'] > 0;
}

// Crear una nueva ruta
public function create() {
    $query = "INSERT INTO " . $this->table_name . "
        (origen_id, destino_id, distancia, costo_km)
        VALUES (?, ?, ?, ?)";

    $stmt = $this->conn->prepare($query);

    // Sanitizar entrada
    $this->origen_id = htmlspecialchars(strip_tags($this->origen_id));
    $this->destino_id = htmlspecialchars(strip_tags($this->destino_id));
    $this->distancia = htmlspecialchars(strip_tags($this->distancia));
    $this->costo_km = htmlspecialchars(strip_tags($this->costo_km));

    // Bind parametros
    $stmt->bindParam(1, $this->origen_id);
    $stmt->bindParam(2, $this->destino_id);
    $stmt->bindParam(3, $this->distancia);
    $stmt->bindParam(4, $this->costo_km);

    if ($stmt->execute()) {

```

```

        return true;
    }

    return false;
}

// Actualizar una ruta
public function update() {
    $query = "UPDATE " . $this->table_name . "
        SET origen_id = ?, destino_id = ?, distancia = ?, costo_km = ?
        WHERE id = ?";

    $stmt = $this->conn->prepare($query);

    // Sanitizar entrada
    $this->origen_id = htmlspecialchars(strip_tags($this->origen_id));
    $this->destino_id = htmlspecialchars(strip_tags($this->destino_id));
    $this->distancia = htmlspecialchars(strip_tags($this->distancia));
    $this->costo_km = htmlspecialchars(strip_tags($this->costo_km));
    $this->id = htmlspecialchars(strip_tags($this->id));

    // Bind parametros
    $stmt->bindParam(1, $this->origen_id);
    $stmt->bindParam(2, $this->destino_id);
    $stmt->bindParam(3, $this->distancia);
    $stmt->bindParam(4, $this->costo_km);
    $stmt->bindParam(5, $this->id);

    if ($stmt->execute()) {
        return true;
    }

    return false;
}

// Eliminar una ruta
public function delete() {
    // Primero verificar si la ruta está siendo utilizada en optimizaciones
    $query = "SELECT COUNT(*) as count FROM resultados_optimizacion WHERE ruta_id = ?";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(1, $this->id);
    $stmt->execute();
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($row['count'] > 0) {
        // La ruta está siendo utilizada, no se puede eliminar
        return false;
    }

    // Si no está siendo utilizada, proceder con la eliminación
    $query = "DELETE FROM " . $this->table_name . " WHERE id = ?";

    $stmt = $this->conn->prepare($query);

```

```
// Sanitizar entrada
$this->id = htmlspecialchars(strip_tags($this->id));

// Bind parametro
$stmt->bindParam(1, $this->id);

if ($stmt->execute()) {
    return true;
}

return false;
}
}
?>
```


TRANSPORTEMODELO.PHP

```
<?php

// Usar rutas absolutas completas
// Obtener la ruta raíz del servidor
$serverRoot = $_SERVER['DOCUMENT_ROOT'];
$projectFolder = '/proyecto_transporte'; // Ajusta esto si tu carpeta tiene otro nombre
$projectRoot = $serverRoot . $projectFolder;

// Definir la ruta base si no está definida
if (!defined('BASE_PATH')) {
    define('BASE_PATH', $projectRoot);
}

// Imprimir rutas para depuración
//echo "Server Root: " . $serverRoot . "<br>";
//echo "Project Root: " . $projectRoot . "<br>";

// Definir rutas a los archivos
$databaseFile = $projectRoot . '/config/database.php';
$matrizInversaFile = $projectRoot . '/utils/MatrizInversa.php';
$rutaModelFile = $projectRoot . '/models/RutaModel.php';

//echo "Intentando cargar database.php desde: " . $databaseFile . "<br>";

// Comprobar si los archivos existen
if (!file_exists($databaseFile)) {
    // Intentar encontrar el archivo en otras ubicaciones posibles
    $alternativeLocations = [
        $projectRoot . '/config/database.php',
        $projectRoot . '/database.php',
        dirname($projectRoot) . '/config/database.php',
        $serverRoot . '/config/database.php'
    ];

    foreach ($alternativeLocations as $location) {
        if (file_exists($location)) {
            $databaseFile = $location;
            echo "Archivo encontrado en ubicación alternativa: " . $databaseFile . "<br>";
            break;
        }
    }

    if (!file_exists($databaseFile)) {
        die("Error: No se encontró el archivo database.php en ninguna ubicación conocida");
    }
}

// Incluir los archivos necesarios
```

```

require_once $databaseFile;

// Solo incluir estos archivos si existen
if (file_exists($matrizInversaFile)) {
    require_once $matrizInversaFile;
} else {
    echo "Advertencia: No se pudo encontrar MatrizInversa.php<br>";
}

if (file_exists($rutaModelFile)) {
    require_once $rutaModelFile;
} else {
    echo "Advertencia: No se pudo encontrar RutaModel.php<br>";
}

class TransporteModel {
    private $conn;
    private $tabla_optimizaciones = "optimizaciones";
    private $tabla_resultados = "resultados_optimizacion";

    public $id;
    public $total_productos;
    public $fecha_calculo;

    public function __construct() {
        $database = new Database();
        $this->conn = $database->getConnection();
    }

    // Obtener todas las optimizaciones
    public function getAllOptimizaciones() {
        $query = "SELECT id, total_productos, fecha_calculo FROM " . $this->tabla_optimizaciones . " ORDER BY fecha_calculo DESC";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();

        return $stmt;
    }

    // Obtener resultados de una optimización
    public function getResultadosOptimizacion($optimizacion_id) {
        $query = "SELECT ro.id, ro.ruta_id, ro.cantidad_productos, ro.costo_total,
                        r.origen_id, r.destino_id, r.distancia, r.costo_km,
                        o.nombre as origen_nombre, d.nombre as destino_nombre
                FROM " . $this->tabla_resultados . " ro
                JOIN rutas r ON ro.ruta_id = r.id
                JOIN ciudades o ON r.origen_id = o.id
                JOIN ciudades d ON r.destino_id = d.id
                WHERE ro.optimizacion_id = ?";

        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(1, $optimizacion_id);
    }
}

```

```

$stmt->execute();

return $stmt;
}

// Guardar una nueva optimización
public function saveOptimizacion($total_productos, $resultados) {
    try {
        $this->conn->beginTransaction();

        // Crear registro de optimización
        $query = "INSERT INTO " . $this->tabla_optimizaciones . " (total_productos) VALUES
(?)";

        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(1, $total_productos);
        $stmt->execute();

        $optimizacion_id = $this->conn->lastInsertId();

        // Guardar resultados de la optimización
        foreach ($resultados as $resultado) {
            $query = "INSERT INTO " . $this->tabla_resultados . "
                (optimizacion_id, ruta_id, cantidad_productos, costo_total)
                VALUES (?, ?, ?, ?)";

            $stmt = $this->conn->prepare($query);
            $stmt->bindParam(1, $optimizacion_id);
            $stmt->bindParam(2, $resultado['ruta_id']);
            $stmt->bindParam(3, $resultado['cantidad_productos']);
            $stmt->bindParam(4, $resultado['costo_total']);
            $stmt->execute();
        }

        $this->conn->commit();
        return $optimizacion_id;
    } catch (Exception $e) {
        $this->conn->rollBack();
        throw $e;
    }
}

// Obtener rutas para el cálculo de optimización
public function getRutasOptimizacion() {
    $rutaModel = new RutaModel();
    $stmt = $rutaModel->getAll();
    $rutas = [];

    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        $rutas[] = [
            'id' => $row['id'],
            'origen_id' => $row['origen_id'],
            'destino_id' => $row['destino_id'],
            'origen_nombre' => $row['origen_nombre'],
            'destino_nombre' => $row['destino_nombre'],

```

```

        'distancia' => $row['distancia'],
        'costo_km' => $row['costo_km']
    ];
}

return $rutas;
}

/**
 * MÉTODO OPTIMIZADO: Realizar el cálculo de optimización usando el enfoque de programación
lineal
 * Este método implementa la estrategia de asignar todos los productos a la ruta con menor
costo
 * cuando no hay restricciones adicionales de distribución.
 */
public function optimizarDistribucion($total_productos) {
    // Obtener todas las rutas disponibles
    $rutas = $this->getRutasOptimizacion();

    if (count($rutas) < 1) {
        throw new Exception("Se requieren al menos una ruta para realizar la optimización.");
    }

    // Calcular el costo total por producto para cada ruta
    foreach ($rutas as &$ruta) {
        $ruta['costo_total_por_producto'] = $ruta['distancia'] * $ruta['costo_km'];
    }

    // Ordenar las rutas por costo total (de menor a mayor)
    usort($rutas, function($a, $b) {
        return $a['costo_total_por_producto'] - $b['costo_total_por_producto'];
    });

    // Inicializar el array de resultados con todas las rutas disponibles
    $resultados = [];
    foreach ($rutas as $ruta) {
        $resultados[] = [
            'ruta_id' => $ruta['id'],
            'origen' => $ruta['origen_nombre'],
            'destino' => $ruta['destino_nombre'],
            'distancia' => $ruta['distancia'],
            'costo_km' => $ruta['costo_km'],
            'cantidad_productos' => 0, // Inicializar con 0 productos
            'costo_total' => 0, // Inicializar con costo 0
            'costo_unitario' => $ruta['costo_total_por_producto']
        ];
    }

    // Aplicar la estrategia de programación lineal:
    // Asignar todos los productos a la ruta con menor costo
    $resultados[0]['cantidad_productos'] = $total_productos;
    $resultados[0]['costo_total'] = $total_productos * $resultados[0]['costo_unitario'];

    // Registrar en el log del sistema para depuración

```

```
        error_log("Optimización realizada: Total productos: " . $total_productos);
        error_log("Ruta óptima: " . $resultados[0]['origen'] . " → " .
$resultados[0]['destino'] .
            " (Costo unitario: " . $resultados[0]['costo_unitario'] . ")");

        return $resultados;
    }
}
?>
```

MATRIZCENTRAL.PHP

```
<?php
class MatrizInversa {

    /**
     * Calcula el determinante de una matriz 3x3
     */
    public static function calcularDeterminante($matriz) {
        if (count($matriz) !== 3 || count($matriz[0]) !== 3) {
            throw new Exception("La matriz debe ser de 3x3 para calcular el determinante");
        }

        $det =
            $matriz[0][0] * ($matriz[1][1] * $matriz[2][2] - $matriz[1][2] * $matriz[2][1]) -
            $matriz[0][1] * ($matriz[1][0] * $matriz[2][2] - $matriz[1][2] * $matriz[2][0]) +
            $matriz[0][2] * ($matriz[1][0] * $matriz[2][1] - $matriz[1][1] * $matriz[2][0]);

        return $det;
    }

    /**
     * Calcula la matriz de cofactores
     */
    public static function calcularCofactores($matriz) {
        if (count($matriz) !== 3 || count($matriz[0]) !== 3) {
            throw new Exception("La matriz debe ser de 3x3 para calcular los cofactores");
        }

        $cofactores = [];

        for ($i = 0; $i < 3; $i++) {
            for ($j = 0; $j < 3; $j++) {
                // Extraemos la submatriz 2x2 eliminando la fila i y columna j
                $submatriz = [];
                for ($r = 0; $r < 3; $r++) {
                    if ($r === $i) continue;
                    $fila = [];
                    for ($c = 0; $c < 3; $c++) {
                        if ($c === $j) continue;
                        $fila[] = $matriz[$r][$c];
                    }
                    $submatriz[] = $fila;
                }

                // Calculamos el determinante de la submatriz 2x2
                $det_submatriz = $submatriz[0][0] * $submatriz[1][1] - $submatriz[0][1] *
$submatriz[1][0];

                // Aplicamos el signo correspondiente
```

```

        $cofactores[$i][$j] = pow(-1, $i + $j) * $det_submatriz;
    }
}

return $cofactores;
}

/**
 * Calcula la matriz adjunta (transpuesta de la matriz de cofactores)
 */
public static function calcularAdjunta($matriz) {
    $cofactores = self::calcularCofactores($matriz);
    $adjunta = [];

    for ($i = 0; $i < 3; $i++) {
        for ($j = 0; $j < 3; $j++) {
            $adjunta[$j][$i] = $cofactores[$i][$j];
        }
    }

    return $adjunta;
}

/**
 * Calcula la matriz inversa de una matriz 3x3
 */
public static function calcularInversa($matriz) {
    $det = self::calcularDeterminante($matriz);

    if ($det == 0) {
        throw new Exception("La matriz no tiene inversa (determinante = 0)");
    }

    $adjunta = self::calcularAdjunta($matriz);
    $inversa = [];

    for ($i = 0; $i < 3; $i++) {
        for ($j = 0; $j < 3; $j++) {
            $inversa[$i][$j] = $adjunta[$i][$j] / $det;
        }
    }

    return $inversa;
}

/**
 * Multiplica una matriz por un vector
 */
public static function multiplicarMatrizVector($matriz, $vector) {
    $resultado = [];
    $filas = count($matriz);
    $columnas = count($matriz[0]);

    if (count($vector) != $columnas) {

```

```

        throw new Exception("Dimensiones incompatibles para multiplicación matriz-vector");
    }

    for ($i = 0; $i < $filas; $i++) {
        $suma = 0;
        for ($j = 0; $j < $columnas; $j++) {
            $suma += $matriz[$i][$j] * $vector[$j];
        }
        $resultado[] = $suma;
    }

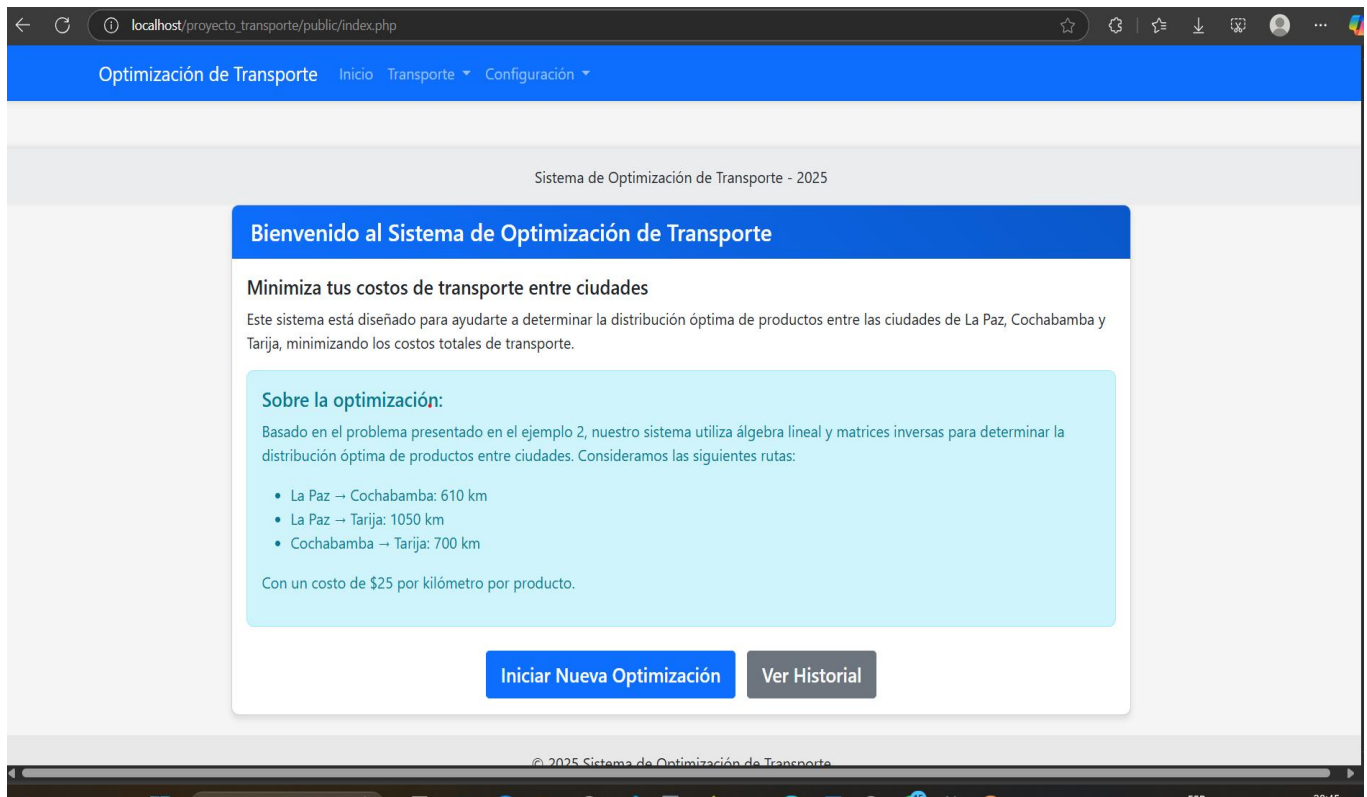
    return $resultado;
}

/**
 * Resuelve un sistema de ecuaciones lineales usando la inversa de la matriz de coeficientes
 * Ax = b, donde x = A^-1 * b
 */
public static function resolverSistema($coeficientes, $terminos_independientes) {
    $inversa = self::calcularInversa($coeficientes);
    $solucion = self::multiplicarMatrizVector($inversa, $terminos_independientes);

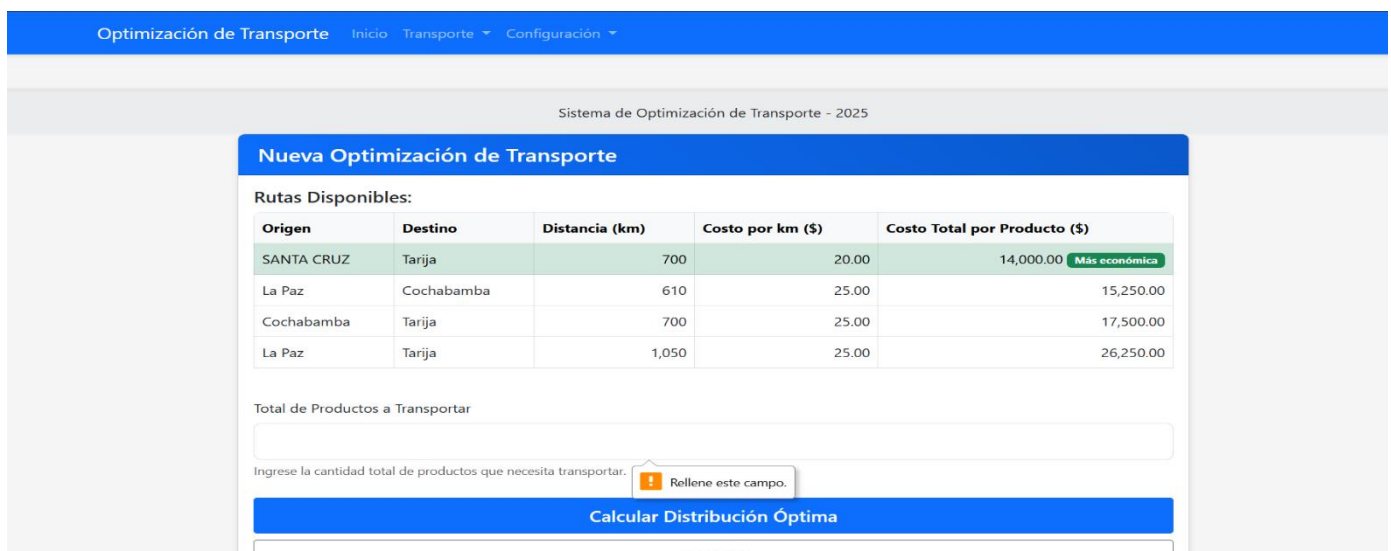
    return $solucion;
}
}
?>

```


VISTA PRINCIPAL DEL SISTEMA



VISTA CALCULO DE LA OPTIMIZACION



VISTA HISTORIAL

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Historial de Optimizaciones

Nueva Optimización

ID	Total de Productos	Fecha	Acciones
11	200 unidades	04/04/2025 21:13	Ver Detalles
10	1,000 unidades	04/04/2025 08:46	Ver Detalles
9	500 unidades	04/04/2025 07:58	Ver Detalles
8	2,000 unidades	04/04/2025 07:45	Ver Detalles
7	1,500 unidades	04/04/2025 06:31	Ver Detalles
6	3,000 unidades	04/04/2025 02:02	Ver Detalles
5	1,000 unidades	04/04/2025 01:59	Ver Detalles
4	100 unidades	04/04/2025 01:37	Ver Detalles
3	1,500 unidades	04/04/2025 01:34	Ver Detalles
2	500 unidades	04/04/2025 01:26	Ver Detalles
1	1,000 unidades	04/04/2025 01:24	Ver Detalles

Volver a Transporte

VISTA calculo de distribución optima

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Nueva Optimización de Transporte

Rutas Disponibles:

Origen	Destino	Distancia (km)	Costo por km (\$)	Costo Total por Producto (\$)
SANTA CRUZ	Tarija	700	20.00	14,000.00 Más económica
La Paz	Cochabamba	610	25.00	15,250.00
Cochabamba	Tarija	700	25.00	17,500.00
La Paz	Tarija	1,050	25.00	26,250.00

Total de Productos a Transportar:

Ingrese la cantidad total de productos que necesita transportar.

Calcular Distribución Óptima

Cancelar

VISTA GESTION DE CIUDADES

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Gestión de Ciudades

Nueva Ciudad

ID	Nombre	Fecha de Creación	Acciones
2	Cochabamba	04/04/2025 00:21	<div>EditarEliminar</div>
1	La Paz	04/04/2025 00:21	<div>EditarEliminar</div>
4	SANTA CRUZ	04/04/2025 21:11	<div>EditarEliminar</div>
3	Tarja	04/04/2025 00:21	<div>EditarEliminar</div>

Volver a Transporte

VISTA GESTION DE CIUDADES

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Nueva Ciudad

Nombre de la Ciudad

Ingrese el nombre completo de la ciudad.

Nota: Las ciudades se utilizarán como origen y destino en las rutas.

Asegúrese de ingresar el nombre correcto de la ciudad.

Cancelar

Guardar Ciudad

VISTA GESTION DE RUTAS

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Gestión de Rutas

Nueva Ruta

ID	Origen	Destino	Distancia (km)	Costo/km (\$)	Costo Total (\$)	Acciones
1	La Paz	Cochabamba	610	25.00	15,250.00	<div>Editar</div> <div>Eliminar</div>
2	La Paz	Tarija	1,050	25.00	26,250.00	<div>Editar</div> <div>Eliminar</div>
3	Cochabamba	Tarija	700	25.00	17,500.00	<div>Editar</div> <div>Eliminar</div>
4	SANTA CRUZ	Tarija	700	20.00	14,000.00	<div>Editar</div> <div>Eliminar</div>

Volver a Transporte

VISTA GESTION DE NUEVA RUTAS

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Nueva Ruta

Ciudad de Origen

Seleccione una ciudad

Ciudad de Destino

Seleccione una ciudad

Distancia (km)

Costo por Kilómetro (\$)

Cancelar

Guardar Ruta

VISTA PANEL DE TRANSPORTE

Optimización de Transporte

InicioTransporteConfiguración

Sistema de Optimización de Transporte - 2025

Gestión de Transporte

Nueva Optimización

Calcular la distribución óptima de productos entre ciudades

Iniciar Optimización

Historial de Optimizaciones

Ver todas las optimizaciones realizadas anteriormente

Ver Historial

Últimas Optimizaciones

ID	Total de Productos	Fecha	Acciones
11	200 unidades	04/04/2025 21:13	Ver Detalles
10	1,000 unidades	04/04/2025 08:46	Ver Detalles
9	500 unidades	04/04/2025 07:58	Ver Detalles
8	2,000 unidades	04/04/2025 07:45	Ver Detalles
7	1,500 unidades	04/04/2025 06:31	Ver Detalles

Ver todas las optimizaciones

VISTA PANEL EXPLICACION DEL CALCULO

Ruta	Origen	Destino	Distancia (km)	Costo/km (\$)	Productos	Costo Total (\$)
1	La Paz	Cochabamba	610	25.00	0	0.00
2	La Paz	Tarija	1,050	25.00	0	0.00
3	Cochabamba	Tarija	700	25.00	0	0.00
4	SANTA CRUZ	Tarija	700	20.00	10,000	140,000,000.00
			Total:		10,000	140,000,000.00

Explicación del Cálculo

El sistema ha resuelto el problema de optimización mediante programación lineal, siguiendo los pasos:

- Se definieron las variables de decisión: cantidad de productos a transportar por cada ruta
- Se estableció la función objetivo: minimizar el costo total de transporte
- Se identificaron las restricciones: transportar un total de 0 productos
- Se calculó el costo unitario para cada ruta multiplicando la distancia por el costo por kilómetro
- Se ordenaron las rutas de menor a mayor costo
- Se asignaron todos los productos a la ruta con menor costo unitario

Esta distribución garantiza el transporte de todos los productos minimizando el costo total.

VISTA INTERPRETACION

Interpretación

Beneficios

Esta solución optimizada ofrece los siguientes beneficios:

- Minimización del costo total de transporte mediante programación lineal
- Simplificación logística al utilizar una sola ruta
- Reducción de la complejidad operativa
- Máximo ahorro en costos para la empresa

Al implementar esta distribución, se asegura el uso óptimo de los recursos de transporte disponibles, aplicando principios matemáticos rigurosos.

Nueva Optimización

VISTA DE BENEFICIOS

Beneficios

Esta solución optimizada ofrece los siguientes beneficios:

- Minimización del costo total de transporte mediante programación lineal
- Simplificación logística al utilizar una sola ruta
- Reducción de la complejidad operativa
- Máximo ahorro en costos para la empresa

Al implementar esta distribución, se asegura el uso óptimo de los recursos de transporte disponibles, aplicando principios matemáticos rigurosos.

Nueva Optimización

