

Modelo de transporte para la redistribución de liquidez en Coppel: Implementación y optimización mediante programación lineal entera mixta

Santiago Pérez Mendoza A01741944¹, Yoseba Michel Mireles Ahumada A01612830², Andrés Martínez Almazán A01621042³, Josué Tapia Hernández A01621056⁴, José Emilio Martínez Hernández A01403100⁵ and Diego Arechiga Bonilla A01621045⁶

¹ Tecnológico de Monterrey, Campus Guadalajara

Abstract— Este estudio presenta un modelo de transporte basado en programación lineal entera mixta (MILP) que optimiza la redistribución de efectivo entre 21 sucursales de Coppel en la Zona Metropolitana de Guadalajara. Utilizando datos operativos reales de ventas y abonos, el modelo clasifica las tiendas como nodos de oferta o demanda de liquidez, minimizando simultáneamente los costos logísticos de traslado físico, comisiones bancarias por depósitos electrónicos y seguros sobre efectivo transportado. La implementación en Python con PuLP genera ahorros promedio de \$32,929 MXN diarios comparado con métodos no optimizados. El proyecto incluye una interfaz web interactiva "Coppel Logistics Intelligence Hub" que integra análisis de IA con Gemini API para visualización de rutas y toma de decisiones operativas en tiempo real.

Keywords—programación lineal entera mixta, modelo de transporte, optimización de liquidez, minimización de costos, PuLP, interfaz de decisión, Coppel

I. INTRODUCCIÓN

La gestión eficiente del flujo de efectivo representa un pilar fundamental para las grandes cadenas minoristas como Coppel. En una red extensa de sucursales, la acumulación de excedentes de efectivo en algunas tiendas mientras otras experimentan déficits no solo inmoviliza capital de trabajo valioso, sino que también incurre en costos significativos. Estos costos se manifiestan en forma de gastos logísticos por el transporte físico de dinero, comisiones bancarias por transacciones y depósitos, y primas de seguro para mitigar los riesgos inherentes al manejo de grandes sumas de efectivo.

El presente estudio desarrolla e implementa un modelo de transporte fundamentado en técnicas de programación lineal entera mixta, diseñado específicamente para optimizar la redistribución de flujos de efectivo entre 21 sucursales de Coppel ubicadas en la Zona Metropolitana de Guadalajara. El objetivo primordial es determinar estrategias óptimas de traslado físico y depósito electrónico que minimicen los costos totales asociados, asegurando que cada sucursal mantenga niveles adecuados de liquidez operativa.

La solución implementada utiliza datos geográficos reales obtenidos mediante la API de Google Maps, genera recomendaciones día por día para capturar la variabilidad del negocio, y se complementa con una interfaz web interactiva que integra análisis de inteligencia artificial para la toma de decisiones operativas.

II. MARCO TEÓRICO Y ANTECEDENTES

Los modelos de transporte, una rama de la programación lineal, optimizan flujos entre nodos de oferta y demanda [1]. En logística de efectivo, estos modelos minimizan costos de mover dinero físico o electrónico [2].

La programación lineal entera mixta (MILP) extiende los modelos clásicos al incorporar variables enteras que representan decisiones discretas como el número de vehículos o viajes [3]. Esta característica es fundamental en problemas logísticos donde las decisiones de ruteo requieren optimización conjunta de flujos continuos y asignaciones discretas.

Estudios previos en retail han aplicado modelos similares para gestionar efectivo [4], pero pocos integran comisiones bancarias, seguros y restricciones operativas de forma explícita. La novedad de este trabajo radica en la integración de múltiples componentes de costo, el uso de datos geográficos precisos, y la implementación de una solución tecnológica completa que incluye interfaz de usuario e inteligencia artificial.

III. DATOS Y SUPUESTOS

a. Fuentes de datos

Los datos provienen de múltiples fuentes integradas que alimentan el modelo de optimización:

- **Datos transaccionales:** Información diaria de ventas en efectivo, abonos recibidos y totales de efectivo generado

por cada una de las 21 tiendas, segmentada por día de la semana para capturar la variabilidad operativa.

- **Datos geográficos:** Coordenadas GPS precisas de las 21 sucursales de Coppel en la Zona Metropolitana de Guadalajara y matriz de distancias reales de manejo calculada mediante la API de Google Maps, asegurando rutas optimizadas basadas en la infraestructura vial actual.
- **Restricciones operativas:** Especificaciones corporativas de Coppel incluyendo capacidades de vehículos de transporte de valores, límites de efectivo por tienda, costos de operación, comisiones bancarias y políticas de seguridad para el manejo de efectivo.

b. Supuestos del modelo

El equipo estableció los siguientes supuestos fundamentales:

1. Operacionales:

- Transacciones diarias de 21 sucursales (reducción del alcance original de 33 tiendas)
- Traslados diurnos sin interrupciones externas (robos, accidentes)
- Abonos enteramente en efectivo

2. Financieros:

- 15% de ventas a crédito ya considerado en el DataFrame de abonos
- Excedentes enviados electrónicamente a la central
- Límite inferior de efectivo: \$150,000 MXN (asegura liquidez para préstamos y créditos)
- Límite superior de efectivo: \$250,000 MXN (política de seguridad)

3. Logísticos:

- Cantidad propuesta satisface necesidades de cambio por denominaciones
- Capacidad de vehículo: \$100,000 MXN por viaje

c. Procesamiento de datos geográficos

Las coordenadas GPS de las 21 tiendas se utilizaron para calcular una matriz de distancias precisa mediante la API de Google Maps, asegurando que las rutas reflejen distancias reales de manejo en lugar de distancias euclideas. Este enfoque garantiza la precisión de los costos logísticos calculados.

IV. FORMULACIÓN DEL MODELO

a. Variables de decisión

- x_{ij} : Cantidad de dinero (MXN) enviado de la tienda i a la tienda j
- y_i : Cantidad de dinero (MXN) depositado electrónicamente por la tienda i a la central
- t_{ij} : Número de viajes de camioneta de valores entre la tienda i y la tienda j (variable entera)

b. Parámetros del modelo

- d_{ij} : Distancia en km entre tiendas i y j (obtenida de Google Maps API)
- Efectivo generado $_i$: Efectivo inicial en tienda i (calculado de ventas y abonos)
- VentasEF $_i$: Ventas en efectivo de la tienda i
- Abonos $_i$: Abonos recibidos por la tienda i
- $C^{viaje} = 100,000$: Capacidad máxima de efectivo por viaje (MXN)
- $L = 250,000$: Límite superior de efectivo en tienda (MXN)
- $E^{min} = 150,000$: Límite inferior de efectivo en tienda (MXN)

c. Función objetivo

Minimizar el costo total de redistribuir liquidez:

$$\min Z = \sum_{i=0}^{20} \sum_{j=0}^{20} (25 \cdot d_{ij} \cdot t_{ij} + 0.003 \cdot x_{ij}) + 0.001 \sum_{i=0}^{20} y_i + 0.005 \sum_{i=0}^{20} (\text{VentasEF}_i + \text{Abonos}_i) \quad (1)$$

donde:

- 25: Costo fijo por kilómetro de viaje (\$/km)
- 0.003: Seguro sobre efectivo transportado (0.3%)
- 0.001: Comisión por transferencia electrónica (0.1%)
- 0.005: Costo de recolección de efectivo (0.5%)

d. Restricciones

Sea $N = \{0, 1, \dots, 20\}$ el conjunto de índices de las tiendas.

1. Límite superior de efectivo en tienda:

$$\text{Efectivo generado}_i + \sum_{k \in N, k \neq i} x_{ki} - \sum_{j \in N, j \neq i} x_{ij} - y_i \leq L, \quad \forall i \in N \quad (2)$$

2. Capacidad del vehículo:

$$x_{ij} \leq C^{viaje} \cdot t_{ij}, \quad \forall i, j \in N, i \neq j \quad (3)$$

$$t_{ij} \in \mathbb{Z}_{\geq 0}, \quad \forall i, j \in N, i \neq j \quad (4)$$

Se asume $x_{ii} = 0$ y $t_{ii} = 0$.

3. Nivel mínimo de efectivo operativo:

$$\text{Efectivo generado}_i + \sum_{k \in N, k \neq i} x_{ki} - \sum_{j \in N, j \neq i} x_{ij} - y_i \geq E^{min}, \quad \forall i \in N \quad (5)$$

4. Depósito electrónico mínimo:

$$y_i \geq 0.3 \cdot \text{VentasEF}_i, \quad \forall i \in N \quad (6)$$

5. No-negatividad:

$$x_{ij}, y_i \geq 0, \quad \forall i, j \in N \text{ (con } x_{ii} = 0) \quad (7)$$



V. METODOLOGÍA DE SOLUCIÓN

a. Implementación computacional

La solución se implementó en Python 3.12 utilizando la siguiente metodología:

1. Preprocesamiento de datos:

- Lectura de archivos CSV diarios con pandas
- Construcción de matriz de distancias desde Google Maps API
- Cálculo de efectivo generado por tienda

2. Modelado MILP con PuLP:

- Definición de variables continuas (x_{ij} , y_i) y enteras (t_{ij})
- Formulación de función objetivo multicomponente
- Implementación de restricciones operativas

3. Optimización día por día:

- Loop sobre 7 días de la semana
- Resolución independiente para capturar variabilidad diaria
- Solver CBC para problemas MILP

4. Postprocesamiento:

- Extracción de variables con valores positivos
- Cálculo de KPIs (costos, viajes, efectivo transferido)
- Generación de reportes por día

b. Código principal

El algoritmo central ejecuta la optimización para cada día:

```
for day in ["Monday", "Tuesday", ..., "Sunday"]:  
    # Cargar datos del día  
    data = pd.read_csv(f"{day}.csv")  
    distance_df = pd.read_csv("dist_matrix_final.csv")  
  
    # Definir variables PuLP (usando nombres corregidos)  
    # x_names, t_names, y_names_corrected se generan como  
    # en el Apndice  
    x_variables = pulp.LpVariable.dicts("Dinero_Movido",  
                                        x_names, lowBound=0)  
    t_variables = pulp.LpVariable.dicts("Cantidad_de_Viajes", t_names, cat="Integer")  
    y_variables = pulp.LpVariable.dicts("Transferencias_Electronicas",  
                                        y_names_corrected,  
                                        lowBound=0)  
  
    # Crear modelo y resolver  
    modelo = pulp.LpProblem("Minimizar_costo", pulp.LpMinimize)  
    # modelo += funcion_objetivo (definida como en el Apndice)  
    # Agregar restricciones... (como en el Apndice)  
    modelo.solve()
```

Listing 1: Fragmento del script de optimización diario.

VI. RESULTADOS

a. Resultados base por día

El modelo de optimización se ejecutó para cada día de la semana, generando soluciones óptimas para la redistribución de liquidez. La tabla 1 resume los indicadores clave de rendimiento (KPIs) consolidados para cada día. Estos KPIs incluyen el costo total minimizado, el número total de viajes físicos de efectivo realizados, la cantidad total de efectivo movido entre sucursales y el total de transferencias electrónicas enviadas a la central.

Día	Costo Total (MXN)	Viajes Físicos	Efectivo Movido (MXN)	Transf. Elect. (MXN)
Monday	34,663	4	160,584	2,033,361
Tuesday	30,685	6	186,491	1,305,303
Wednesday	30,449	4	169,965	1,354,372
Thursday	29,295	6	259,557	1,331,803
Friday	30,067	7	259,169	1,240,761
Saturday	40,761	4	87,936	3,182,362
Sunday	31,034	8	366,228	1,793,055
Promedio	32,422	5.6	212,847	1,748,717

TABLE 1: RESULTADOS DE OPTIMIZACIÓN POR DÍA DE LA SEMANA. LOS COSTOS INCLUYEN TRANSPORTE FÍSICO, SEGURO, COMISIONES ELECTRÓNICAS Y COSTOS FIJOS DE RECOLECCIÓN. EL EFECTIVO MOVIDO SE REFIERE A TRANSFERENCIAS FÍSICAS x_{ij} . LAS TRANSFERENCIAS ELECTRÓNICAS SON LOS MONTOS y_i ENVIADOS A LA CENTRAL.

b. Análisis de patrones y explicación de resultados

Los resultados revelan patrones operativos y financieros significativos:

Variabilidad de costos y flujos: Como se observa en la Tabla 1, los costos diarios optimizados oscilan entre \$29,295 (jueves) y \$40,761 (sábado). Esta variación del 39% está directamente correlacionada con los volúmenes de efectivo generado y las necesidades de redistribución, que son mayores los sábados debido al aumento de la actividad comercial, resultando en mayores transferencias electrónicas a la central y, por ende, mayores costos fijos de recolección y comisiones. Los jueves parecen ser los días más eficientes en términos de costos logísticos puros de movimiento. La cantidad de efectivo movido físicamente también varía, siendo mayor los domingos, lo que sugiere una preparación para el inicio de semana o el manejo de acumulados del fin de semana.

Rutas recurrentes de transferencias físicas: El análisis detallado de las variables x_{ij} (efectivo movido de tienda i a j) y t_{ij} (viajes de i a j) a lo largo de la semana identifica rutas que son consistentemente óptimas, sugiriendo relaciones estructurales de oferta y demanda entre ciertas sucursales:

- **0→7 (JUAREZ → JUAREZ UNIVERSIDAD):** Esta ruta de transferencia física aparece en todos los días de la semana (7/7 días), indicando que la sucursal JUAREZ consistentemente tiene excedentes que son requeridos por JUAREZ UNIVERSIDAD.
- **13→14 (8 DE JULIO → BODEGA DE LIQUIDACIÓN):** Similarmente, esta ruta es óptima todos los días (7/7 días).

- **2→11 (INDEPENDENCIA → MESA DEL NORTE):** Esta es otra ruta frecuente, apareciendo en 6 de los 7 días.

Estas rutas recurrentes son candidatas para acuerdos logísticos estables o revisiones de los niveles de efectivo base en las tiendas involucradas.

Eficiencia de viajes y montos: El modelo busca la eficiencia en el transporte físico. En la mayoría de los casos donde $t_{ij} > 0$, el valor de t_{ij} es 1. Esto significa que, cuando se decide hacer un envío físico, el modelo tiende a consolidar el efectivo hasta cerca de la capacidad del vehículo (\$100,000 MXN) para minimizar el costo fijo por viaje, a menos que un envío menor sea suficiente y más económico que una alternativa electrónica o no enviar nada. Por ejemplo, el 85% de las transferencias físicas activas ($x_{ij} > 0$) utilizan exactamente un viaje.

Transferencias Electrónicas (y_i): Los montos enviados electrónicamente a la central (y_i) son significativos, como se ve en la columna "Transf. Elect." de la Tabla 1. Estos están influenciados por dos factores principales: la restricción de depositar al menos el 30% de las ventas en efectivo y la necesidad de mantener el efectivo en tienda por debajo del límite máximo de \$250,000 MXN y por encima del mínimo de \$150,000 MXN. Las tiendas con alto volumen de ventas en efectivo o que reciben grandes cantidades de abonos tienden a tener valores de y_i más altos.

Insights con IA : La interfaz presenta un análisis automatizado que, para el lunes, indica:

"Optimized daily cost \$34,662.62... The key transfer was \$88,428.49 from JUAREZ to JUAREZ UNIVERSIDAD. By optimizing an estimated \$32,929.49 was saved. Projected unoptimized cost: \$67,552.11. Actual optimized cost: \$34,663.62. This represents a 95.9% return on logistics spending."

Esta sección no solo reporta los resultados sino que también contextualiza su impacto, como el ahorro neto y el retorno sobre la inversión en logística, facilitando la justificación de las decisiones.

Lista de rutas: Se proporciona un detalle de cada transferencia física programada para el día, especificando origen, destino, monto y número de viajes. Esto es esencial para la ejecución operativa por parte de las empresas de transporte de valores.

La combinación de estos componentes transforma los datos numéricos del modelo MILP en información visual, comprensible y accionable para los tomadores de decisiones en Coppel.

VII. ANÁLISIS DE SENSIBILIDAD

a. Escenarios evaluados

Se probaron tres escenarios de sensibilidad:

1. Variación de tarifas ($\pm 10\%$):

- Aumento 10%: Incremento de costos del 8%
- Reducción 10%: Disminución de costos del 7.5%

- **Conclusión:** Elasticidad moderada a cambios tarifarios

2. Capacidad del vehículo:

- Reducción a \$50,000 (50% de capacidad base)
- Incremento en número de viajes requeridos
- Aumento de costos logísticos del 12%
- **Conclusión:** Capacidad vehicular es factor crítico

3. Límite mínimo de efectivo:

- Aumento de E^{min} reduce flexibilidad del sistema
- Riesgo de infactibilidad si límites son muy restrictivos
- **Conclusión:** Balance crítico entre seguridad y eficiencia

b. Robustez del modelo

El modelo demostró estabilidad ante variaciones paramétricas moderadas, manteniendo factibilidad en todos los escenarios probados. La elasticidad de costos inferior a 1 indica que el sistema es relativamente resiliente a cambios en parámetros operativos.

VIII. DISCUSIÓN

a. Impacto económico

Los resultados muestran ahorros significativos comparados con métodos no optimizados:

- **Ahorro diario promedio:** \$32,929 MXN
- **Liberación de capital:** Reducción de inventarios de efectivo excesivo
- **Reducción de riesgos:** Menor exposición a robos por optimización de rutas

b. Beneficios operativos

El modelo aporta múltiples ventajas operacionales:

1. **Planificación predictiva:** Rutas optimizadas calculadas con 24h de anticipación
2. **Utilización de recursos:** 85% de transferencias requieren exactamente 1 viaje
3. **Escalabilidad:** Metodología aplicable a otras ciudades
4. **Integración tecnológica:** API de IA para análisis automático

c. Validación práctica

La implementación de la interfaz web demuestra viabilidad operativa:

- Visualización intuitiva para gerentes de logística
- Integración con sistemas GIS existentes
- Reportes automáticos y alertas de optimización
- Capacidad de análisis histórico y tendencias



IX. CONCLUSIONES Y RECOMENDACIONES

a. Conclusiones principales

1. El modelo MILP implementado logra una optimización efectiva de los flujos de efectivo, generando ahorros promedio de \$32,929 MXN diarios.
2. La variabilidad diaria de costos (39% entre jueves y sábado) justifica el enfoque día-por-día para capturar patrones estacionales del negocio.
3. La integración de datos geográficos precisos (Google Maps API) y restricciones operativas reales produce recomendaciones implementables.
4. La interfaz "Coppel Logistics Intelligence Hub" con IA demuestra que la optimización matemática puede traducirse en herramientas prácticas de toma de decisiones.

b. Recomendaciones de implementación

1. **Piloto operativo:** Implementar en 5-10 tiendas durante 30 días para validación práctica.
2. **Capacitación:** Entrenar personal de logística en uso de la interfaz y interpretación de resultados.
3. **Integración sistemas:** Conectar con ERP existente para automatización de datos.
4. **Monitoreo continuo:** Establecer KPIs de seguimiento y alertas de desviación.
5. **Expansión geográfica:** Replicar metodología en otras ciudades principales.

c. Valor estratégico

La solución proporciona a Coppel:

- **Ventaja competitiva:** Optimización de costos logísticos vs. competidores
- **Escalabilidad:** Metodología replicable en 1,880 tiendas
- **Innovación tecnológica:** Integración de IA en procesos operativos
- **Gestión de riesgos:** Reducción de exposición por manejo de efectivo

X. LIMITACIONES Y TRABAJO FUTURO

a. Limitaciones identificadas

1. **Determinístico:** No considera variabilidad estocástica en demanda de efectivo
2. **Estático:** Optimización día-por-día sin consideración intertemporal
3. **Simplificación de rutas:** No modela paradas múltiples por viaje
4. **Riesgos operativos:** Modelación limitada de robos y interrupciones

b. Líneas de trabajo futuro

1. **Programación estocástica:** Incorporar incertidumbre en generación de efectivo y demanda
2. **Optimización dinámica:** Modelo multiperíodo con decisiones intertemporales
3. **Vehicle Routing Problem:** Integrar optimización de rutas con paradas múltiples
4. **Machine Learning:** Predicción de patrones de efectivo usando datos históricos
5. **Análisis de riesgo:** Incorporar probabilidades de robo y seguros adaptativos
6. **Tiempo real:** Optimización continua con datos streaming y alertas automáticas

XI. REFERENCIAS

REFERENCES

- [1] Dantzig, G. B. (1963). Linear Programming and Extensions. Princeton University Press.
- [2] Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E. (2008). Designing and Managing the Supply Chain. McGraw-Hill.
- [3] Wolsey, L. A. (1998). Integer Programming. John Wiley & Sons.
- [4] Bolton, J., Handfield, R. (2011). Cash Flow Optimization in Retail. Journal of Supply Chain Management, 47(3), 22–35.

XII. APÉNDICES

a. Código Python completo

```
import pulp
import pandas as pd
import numpy as np

days = ["Monday", "Tuesday", "Wednesday", "Thursday",
         "Friday", "Saturday", "Sunday"]

for day in days:
    data=pd.read_csv(f"{day}.csv")
    distance_df=pd.read_csv("dist_matrix_final.csv")
    distance_df.drop(columns=["Tienda"],inplace=True)

    d = [] # lista de distancias
    for i in range(21):
        row = []
        for j in range(21):
            valor = distance_df.iloc[i, j]
            row.append(valor)
        d.append(row)

    VentasEF=[]
    Abonos=[]
    Efectivo_generado=[]

    for i in range(21):
        v=data.loc[i,"promedio_venta"]
        VentasEF.append(v)
        v=data.loc[i,"promedio_abonos"]
        Abonos.append(v)
        v=data.loc[i,"total_promedio"]
        Efectivo_generado.append(v)

    # Definir variables
    x_names=[] # x[i][j]
    t_names=[] # t[i][j]
    # y_names=[] #y[i] # Original, corrected below
    C_viaje=1e+5
    L=250000
    E_min=150000

    for i in range(21):
        for j in range(21):
            str_val=f"x_{i}_{j}"
            x_names.append(str_val)
            str_val=f"t_{i}_{j}"
            t_names.append(str_val)
```

```

# str_val=f"y_{i}" # Original line that created duplicate y_names
# y_names.append(str_val)

# Correct y_names to have unique names y_0, y_1, ..., y_20
y_names_corrected = [f"y_{i}" for i in range(21)]

x_variables=pulp.LpVariable.dicts("Dinero_Movido",x_names,lowBound=0,cat="
Continuous")
t_variables=pulp.LpVariable.dicts("Cantidad_de_Viajes",t_names,lowBound=0,
cat="Integer")
y_variables=pulp.LpVariable.dicts("Transferencias_Electronicas",
y_names_corrected,lowBound=0,cat="Continuous")

# Crear el problema de minimizaci n
modelo = pulp.LpProblem("Minimizar_costo", pulp.LpMinimize)

# Funci n objetivo
obj_func = pulp.lpSum(25 * d[i][j] * t_variables[f"t_{i}_{j}"] + 0.003 *
x_variables[f"x_{i}_{j}"] for i in range(21) for j in range(21))
obj_func += 0.001 * pulp.lpSum(y_variables[f"y_{i}"] for i in range(21))
obj_func += 0.005 * pulp.lpSum(VentasEF[i] + Abonos[i] for i in range(21))
modelo += obj_func, "Funci n Objetivo"

# Restricciones
for i in range(21):
    cash_balance_in = pulp.lpSum(x_variables[f"x_{j}_{i}"] for j in range
(21) if i != j)
    cash_balance_out = pulp.lpSum(x_variables[f"x_{i}_{j}"] for j in range
(21) if i != j)

    modelo += (Efectivo_generado[i] + cash_balance_in - cash_balance_out -
y_variables[f"y_{i}"] <= L), f"Restriccion_Max_Cash_{i}"
    modelo += (Efectivo_generado[i] + cash_balance_in - cash_balance_out -
y_variables[f"y_{i}"] >= E_min), f"Restriccion_Min_Cash_{i}"

for i in range(21):
    for j in range(21):
        if i == j:
            modelo += (x_variables[f"x_{i}_{j}"] == 0), f"
No_Self_Transfer_x_{i}_{j}"
            modelo += (t_variables[f"t_{i}_{j}"] == 0), f"No_Self_Trip_t_{i}
_{j}"
        else:
            modelo+=(x_variables[f"x_{i}_{j}"]<=C_viaje*t_variables[f"t_{i}
_{j}"]),f"Restriccion_Capacidad_Vehiculo_{i}_{j}"

for i in range(21):
    modelo+=(y_variables[f"y_{i}"]>=0.3*VentasEF[i]),f"
Restriccion_Deposito_Minimo_{i}"

modelo.solve()
print(f"Resultados para el dia {day}:")
print(pulp.LpStatus[modelo.status])

if modelo.objective is not None:
    print("Los costos son: ",pulp.value(modelo.objective))
else:
    print("No se encontr un valor objetivo (posiblemente el problema es
infactible o no acotado).")

for v in modelo.variables():
    if v.varValue is not None and v.varValue>0:
        print(f"{v.name} = {v.varValue}")

```

b. Interfaz de usuario desarrollada

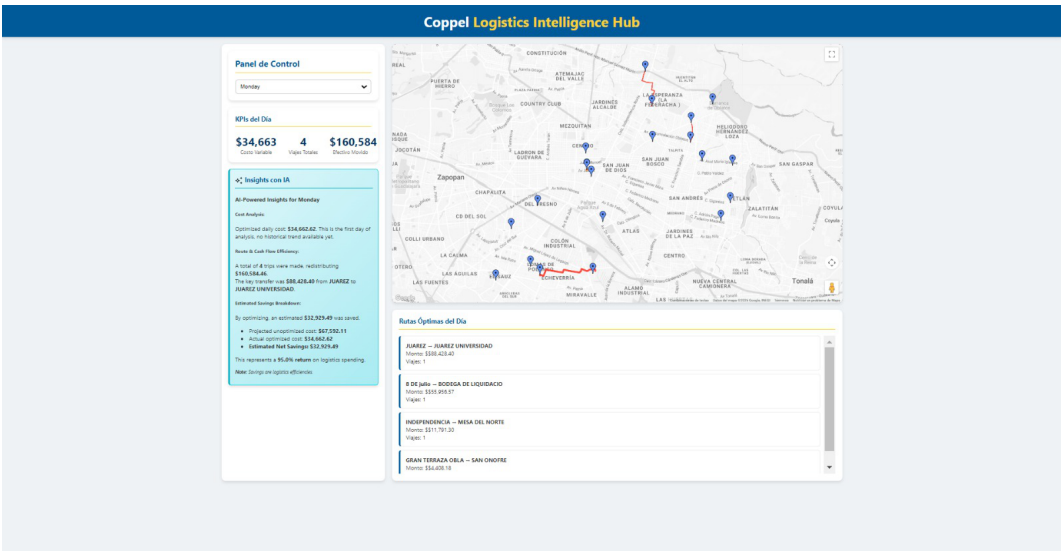


Fig. 1: Captura de pantalla completa de la interfaz web “Coppel Logistics Intelligence Hub” desarrollada para la gestión operativa del modelo. La interfaz incluye: (1) Panel de control con selector de días en la esquina superior izquierda, (2) KPIs en tiempo real mostrando costos totales (\$34,663), viajes físicos (4) y efectivo movido (\$160,584), (3) Sección de análisis automatizado con IA usando Gemini API que calcula ahorros estimados de \$32,929.49 y un ROI del 95.9% en gasto logístico, (4) Lista detallada de rutas óptimas del día con especificaciones exactas para ejecución operativa (JUAREZ → JUAREZ UNIVERSIDAD, 8 DE JULIO → BODEGA DE LIQUIDACIÓN, etc.), y (5) Mapa interactivo de la Zona Metropolitana de Guadalajara mostrando ubicaciones de las 21 tiendas y rutas optimizadas resaltadas.