

# Caso de Estudio: Optimización de Red de Transporte (CDMX)

## 1. Descripción del Problema

Se modeló una sección de la red de transporte (similar a la Línea 1 del Metro CDMX) con transbordos ficticios para analizar tiempos de viaje, centralidad y robustez ante fallos.

## 2. Implementación

Se desarrolló un sistema en Python (`transport_system.py`) con las siguientes capacidades:

- **Dijkstra:** Para consultas rápidas de rutas desde una estación origen (ej. Observatorio).
- **Floyd-Warshall:** Para pre-computar todas las distancias y calcular métricas de centralidad.
- **Análisis de Bottlenecks:** Identificación de estaciones críticas basada en el promedio de tiempo de viaje a todas las demás.
- **Simulación de Fallos:** Evaluación del impacto al eliminar un nodo de la red.

## 3. Resultados del Análisis

### Tiempos desde Observatorio

Gracias a los transbordos añadidos (ej. Bus directo a Pino Suárez), algunos tiempos son competitivos frente a la ruta lineal.

- A Tacubaya: 4 min
- A Insurgentes: 15 min
- A Pino Suárez: 25 min (Directo) vs ~23 min (por línea rosa)

### Estaciones Críticas (Bottlenecks)

Las estaciones con menor tiempo promedio de viaje hacia el resto de la red son:

1. **Insurgentes (7.20 min):** Ubicación central estratégica.
2. **Cuauhtémoc (7.40 min)**
3. **Sevilla (7.50 min)**

## Simulación de Cierre: Estación Balderas

Al cerrar "Balderas", una estación clave de transbordo (en nuestro modelo ficticio conecta con Tacubaya), la red sufre una degradación significativa.

- **Nueva estación crítica:** Juanacatlán.
- **Impacto:** El tiempo promedio de viaje desde la nueva estación central subió drásticamente a **15.33 min** (vs 7.20 min original). Esto indica que la red depende fuertemente de los enlaces rápidos que proveía Balderas.

## 4. Reflexión Crítica

- **Solución IA:** La combinación de Dijkstra para usuario final y FW para análisis estructural es óptima para redes de este tamaño (< 100 nodos según la IA y yo), quizá para redes más grandes deberíamos considerar el apoyo de más algoritmos y optimizaciones.
- **Que mejoraría yo como estudiante?:**
  - Incluir penalización por tiempo de transbordo (no solo tiempo de viaje).
  - Modelar frecuencias de trenes como pesos dinámicos.
  - Usar librerías como NetworkX para grafos masivos (esta optimizada con los mejores algoritmos y las mejores prácticas, así no reinventas la rueda), aunque la implementación manual es excelente para fines educativos y control total.