

Informe Técnico: Simulador de Ruta de Transportes

Realizado por: Juan Esteban Rijo Pineda y Jean Carlos Cruz. **Materia:** Algoritmos Clasicos

1. Estructura de Datos

Para modelar la red de transporte de manera eficiente, el núcleo del proyecto se basa en la estructura de **Grafos Dirigidos**.

- **Nodos (Vértices):** Representan las **Paradas**. Cada parada es una entidad única dentro del sistema.
- **Aristas (Enlaces):** Representan las **Rutas** que conectan dos paradas. A diferencia de un grafo simple, nuestras aristas contienen múltiples pesos o atributos para simular la realidad:
 - Distancia (km).
 - Costo (\$).
 - Tiempo (minutos).
- **Almacenamiento:** Se optó por una estructura que permite registrar y listar estas conexiones dinámicamente, facilitando la adición o eliminación de nodos sin romper la integridad de la red.

2. Algoritmos Implementados

Para cumplir con el objetivo de encontrar la "ruta más eficiente", no nos limitamos a un solo método. Implementamos tres algoritmos clásicos de la teoría de grafos para cubrir distintos escenarios:

- **Algoritmo de Dijkstra:** Lo utilizamos como el motor principal para grafos con pesos positivos. Es el más eficiente para calcular el camino más corto entre una parada de origen y una de destino basándonos en tiempo o distancia.
- **Algoritmo de Floyd-Warshall:** Implementado para ofrecer un análisis completo de la red. Este algoritmo nos permite calcular los caminos más cortos entre **todos los pares de nodos** simultáneamente, ideal para visualizar la conectividad global del sistema.
- **Algoritmo de Bellman-Ford:** Se incluyó para dar robustez al sistema. Nos permite manejar casos donde existan pesos negativos (si aplicara a algún tipo de bonificación en la ruta) y es capaz de detectar ciclos negativos que podrían romper la lógica del simulador.

3. Decisiones de Diseño

El diseño del software se centró en la **usabilidad** y la **flexibilidad** para el usuario final. Las principales decisiones tomadas fueron:

- **Interfaz Gráfica Modular:** Decidimos separar las funcionalidades en paneles dedicados. Existe un panel exclusivo para el registro (entrada de datos) y otro para la visualización (salida de datos), evitando saturar al usuario con información.
- **Visualización de Grafos:** Se determinó que una lista de texto no era suficiente. Por ello, integramos una visualización gráfica que muestra las rutas, direcciones y pesos, permitiendo entender la estructura de la red de un vistazo.
- **Parametrización Completa:** En lugar de calcular rutas solo por distancia, diseñamos el sistema para que el usuario pueda ingresar tiempo y costo. Esto permite que el simulador sea útil en escenarios más realistas donde la ruta más corta no siempre es la más rápida o la más barata.
- **Edición Dinámica:** Se implementó la capacidad de editar o eliminar rutas y paradas ya creadas, dando control total al administrador de la red para corregir errores sin tener que reiniciar el programa.

4. Pruebas Realizadas

Para garantizar el correcto funcionamiento del simulador, se realizaron las siguientes pruebas:

- **Pruebas de Registro:** Se verificó la creación exitosa de paradas y rutas, asegurando que los datos (distancia, costo, tiempo) se guarden correctamente en la estructura del grafo.
- **Pruebas de Integridad:** Se realizaron tests eliminando paradas intermedias para confirmar que las rutas conectadas se actualizan o eliminan correctamente, evitando "rutas huérfanas".
- **Validación de Algoritmos:**
 - Se compararon resultados manuales vs. los obtenidos por **Dijkstra** en rutas simples para confirmar la precisión.
 - Se probó **Bellman-Ford** introduciendo valores atípicos para verificar que el sistema no colapsara y detectara ciclos correctamente.
- **Pruebas de Visualización:** Se confirmó que el grafo dibujado en pantalla corresponda fielmente a los datos ingresados en el listado de texto.