

Sistema de planificación de viajes

Descripción General

El programa es un sistema de planificación de viajes que utiliza grafos para modelar ciudades, conexiones y medios de transporte.

Su objetivo es ayudar al usuario a organizar un itinerario óptimo entre varias ciudades, teniendo en cuenta criterios como costo, tiempo, filtros de transporte y límites personalizados.

Las ciudades y conexiones están almacenadas en archivos JSON separados por continente, lo que permite construir un grafo de forma dinámica en función de la región que elija el usuario.

1. Representación mediante grafos

- Nodos → ciudades
- Aristas → conexiones entre ciudades
- Atributos en las aristas → lista de medios de transporte disponibles, cada uno con:
 - tipo (avión, tren, bus...)
 - costo
 - tiempo de viaje
 - disponibilidad

Cada arista puede tener varios medios de transporte asociados, representando distintas maneras de viajar entre dos ciudades.

2. Funcionalidad principal del programa

El sistema debe permitir al usuario:

2.1 Seleccionar un continente o región

El usuario elige uno de los archivos JSON (Europa, Asia, etc.).

El programa carga:

- lista de ciudades (nodos)
- lista de conexiones con sus medios de transporte (aristas)

Con esto se construye el grafo principal del viaje.

2.2 Elegir las ciudades que desea visitar

El usuario tiene dos opciones:

- **Recorrido fijo**

Indica el orden exacto en el que quiere visitar las ciudades, se selecciona las ciudades y se van agregando a una estructura en ese orden (ej: Madrid → París → Roma). El costo y tiempo dependen del medio de transporte

- **Recorrido libre**

Indica sólo las de ciudades seleccionandolas, ya no solo dependen del medio de transporte si no del recorrido elegido

2.3 Definir criterios de optimización

El usuario selecciona cómo desea calcular la ruta:

- Ruta más barata (minimizar costo total)
- Ruta más rápida (minimizar tiempo total)
- Ruta equilibrada (ponderación entre tiempo y costo)

El programa aplicará algoritmos de rutas en grafos como Dijkstra o variantes modificadas.

2.4 Establecer restricciones opcionales

El usuario puede configurar:

Filtros de transporte

- no aviones
- no trenes
- no buses

Límite de presupuesto

Cualquier ruta que supere el monto permitido se descarta.

Límite de tiempo total

Incluye:

- tiempo de transporte
- tiempo de estancia en cada ciudad (ingresado por el usuario)

3. Generación del itinerario

Con los datos, criterios y restricciones seleccionados:

- El sistema busca la ruta óptima entre cada par de ciudades.
- En modo libre, calcula el mejor orden para visitarlas (heurística TSP).
- El resultado final incluye:
 - ruta completa
 - medios de transporte seleccionados
 - costo total
 - tiempo total
 - tiempos de estancia
 - desglose tramo por tramo

5. Algoritmos utilizados

- Dijkstra → ruta más barata / rápida
- Dijkstra modificado → rutas con restricciones
- Búsqueda exhaustiva o heurística → recorrido libre (TSP pequeño)
- Modificación dinámica de pesos → simulación de eventos
- Validación de restricciones → dinero, tiempo, filtros

6. Interfaz de usuario

El programa mostrará:

- menú para seleccionar continente
- menú para elegir ciudades
- filtros y restricciones
- criterio de optimización
- itinerario completo

7. Resumen

El programa debe:

1. Cargar datos de ciudades desde JSON por continente
2. Construir un grafo dinámico con rutas y múltiples transportes
3. Permitir al usuario definir ciudades, filtros y criterios
4. Calcular la mejor ruta (fija o libre)
5. Producir un itinerario completo (costo + tiempo)