

Gestor de Rutas municipal (Bucaramanga):

En esta tercera entrega, se reemplazó la estructura de árbol por un enfoque basado en grafos ponderados para modelar y gestionar las rutas entre distintos barrios de Bucaramanga. Este cambio nos permite representar de forma más flexible las conexiones reales entre ubicaciones, asignando distancias personalizadas entre cada par de barrios conectados.

Ventajas y desventajas del uso de grafos comparado con árboles o listas:

- Ventajas de usar árboles

- Modelado natural de conexiones reales:
Cada barrio se conecta directamente con uno o varios vecinos, permitiendo representar rutas reales y no jerárquicas.
- Soporte para pesos personalizados:
Cada conexión puede tener una distancia diferente, lo cual es fundamental para encontrar rutas óptimas.
- Búsqueda eficiente de rutas:
Gracias a algoritmos como Dijkstra y Floyd-Warshall, se puede hallar la distancia mínima entre cualquier par de barrios de forma precisa.
- Flexibilidad y edición dinámica:
Pensar en un árbol nos ayuda a entender mejor las conexiones entre lugares, como si fuera un mapa mental de rutas.

- Desventajas de usar árboles

- Complejidad de implementación:
Comparado con listas o árboles, programar y mantener un grafo requiere más esfuerzo y conocimiento de algoritmos avanzados.
- Mayor consumo de memoria
Guardar pesos, nodos y conexiones requiere más recursos computacionales.



JOHAN FRANCISCO LOPEZ JAIMES - 2241548
EDUARDO ZAMBRANO HERNANDEZ - 2241636
JUAN FELIPE HERNANDEZ QUINTERO - 2241796
JOSEPH EMMANUEL SANCHEZ SIERRA – 2240959

Materia: Estructuras de datos y Análisis de Algoritmos
Grupo: E1

Librerías Clave en el Proyecto:

Para el desarrollo de este proyecto, trabajamos con tres librerías fundamentales que nos permitieron implementar la visualización y lógica de rutas de manera eficiente:

- NetworkX:

Esta librería fue el núcleo del proyecto, ya que permite crear y manipular estructuras de grafos de forma sencilla y poderosa.

- Fue utilizada para crear un grafo no dirigido, donde cada nodo representa un barrio y cada arista una conexión entre ellos.
- Asigna un peso (distancia) a cada conexión, permitiendo representar rutas de diferentes longitudes.
- Facilitó la implementación de algoritmos como:
 - **Dijkstra**, para encontrar el camino más corto entre dos puntos.
 - **Floyd-Warshall**, para generar la matriz de distancias mínimas entre todos los barrios.

Gracias a NetworkX, pudimos abstraer la lógica compleja de grafos y centrarnos en el diseño funcional del sistema.

- Matplotlib (Pyplot):

Fue utilizada para la representación gráfica del grafo y las rutas calculadas.

- A través de funciones como `plt.figure()` y `nx.draw()`, se creó una visualización clara de los barrios y sus conexiones.
- Se usó `nx.draw_networkx_edge_labels()` para mostrar los pesos (distancias) de cada conexión directamente sobre el grafo.
- Al calcular rutas óptimas, se utilizaron colores distintivos (por ejemplo, rojo) para resaltar el camino más corto entre dos barrios.

Esto hizo que el sistema no solo fuera funcional, sino también visualmente intuitivo para el usuario.

- **Pandas:**

Tuvo un rol específico pero muy útil: mostrar la matriz de distancias mínimas calculada por el algoritmo de Floyd-Warshall.

- Se utilizó para crear una tabla con las distancias entre todos los barrios, facilitando su lectura y análisis.
- La matriz generada permite entender rápidamente qué tan lejos está cada barrio del resto, sin necesidad de graficar todos los caminos posibles.

Evolución del proyecto

La tercera entrega marca una transición clave:

- Se pasó de estructuras jerárquicas (árboles) a grafos con conexiones múltiples.
- Se reemplazó BFS por algoritmos de búsqueda óptima (Dijkstra, Floyd-Warshall).
- Se implementó un menú interactivo que permite agregar y eliminar barrios, consultar rutas y modificar conexiones.
- La visualización se actualizó para mostrar los caminos resaltados sobre el grafo completo.

Esta evolución permite que el sistema funcione como un GPS básico urbano: el usuario elige origen y destino, y el programa encuentra la ruta más corta, mostrándola gráficamente y calculando la distancia total. La estructura actual es escalable y se adapta fácilmente a nuevas expansiones del mapa.