

Gestor de Rutas municipal (Bucaramanga):

En nuestro proyecto, usamos estructuras de árbol para organizar y buscar rutas entre distintos lugares de Bucaramanga. Si bien podríamos haber optado por listas simples para guardar las ubicaciones, los árboles nos ofrecen ventajas clave cuando trabajamos con conexiones jerárquicas, caminos y relaciones entre sitios.

Ventajas y desventajas de los árboles comparado con las listas:

- Ventajas de usar árboles

- Organización jerárquica:
Cada lugar se conecta directamente a su "padre", imitando la estructura real de calles, barrios y puntos de interés. No solo almacenamos el lugar, sino también cómo llegar a él.
- Búsqueda eficiente de rutas
Con algoritmos como BFS (Búsqueda en Anchura) o DFS (Búsqueda en Profundidad), podemos encontrar rutas de manera rápida y ordenada. En una lista, tendríamos que revisar manualmente cada elemento, lo que sería mucho más lento.
- Escalabilidad
Si el mapa crece y añadimos nuevos lugares, el árbol se adapta sin problemas. Simplemente agregamos nodos en la rama correcta, sin tener que reorganizar toda la estructura.
- Visualización intuitiva
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

- Mayor complejidad de implementación
Construir un árbol y programar algoritmos de búsqueda (como BFS o DFS) requiere más esfuerzo que simplemente recorrer una lista.
- Mas consumo de memoria
Cada nodo del árbol guarda referencias a su padre e hijos, lo que ocupa más espacio que una simple lista de nombres.

- Dependencia de librerías externas
Requiere usar librerías como bigtree, mientras que las listas son parte básica del lenguaje.

Si solo necesitáramos almacenar una lista de lugares sin relaciones entre ellos, una lista simple sería suficiente. Pero como nuestro objetivo es modelar rutas reales, conexiones entre sitios y encontrar caminos de manera rápida y organizada, los árboles son la mejor opción.

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:

- **Matplotlib (Pyplot):**
 - Esta librería fue nuestra herramienta principal para la visualización gráfica.
 - La utilizamos específicamente para:
 - Crear la interfaz gráfica del mapa mediante plt.subplots()
 - Trazar las rutas entre ubicaciones usando ax.plot(), donde representamos los caminos con líneas rojas que conectan los puntos clave
 - Nos permitió integrar perfectamente el mapa base con los elementos dinámicos de nuestra solución
- **Matplotlib.image (mpimg)**
 - Este módulo fue esencial para el manejo de imágenes dentro de nuestro sistema
 - Su función principal fue cargar el archivo Mapa_Bucaramanga.jpg que sirvió como base
 - Nos permitió superponer las rutas calculadas sobre la representación real de la ciudad, dando mayor contexto geográfico a los resultados
- **Bigtree**
 - Con ella implementamos un árbol jerárquico donde:
 - Cada nodo representa un lugar significativo (ej: "UIS", "SanToto")
 - Las relaciones padre-hijo modelan las conexiones reales entre ubicaciones



- Simplificó enormemente operaciones complejas como:
 - Búsqueda de rutas óptimas
 - Implementación de algoritmos como BFS (Búsqueda en Anchura)
 - Manejo eficiente de las relaciones geográficas

Evolución del proyecto

El cambio clave radica en que pasamos de usar simples listas a implementar:

- Árboles de conexiones: Donde cada lugar sabe cómo se relaciona con los demás
- Búsqueda inteligente (BFS): Que encuentra automáticamente el camino más corto entre dos puntos
- Funciona como un mini-GPS: Elige origen y destino, y el sistema traza la ruta óptima
- Mapa interactivo: Las rutas se dibujan sobre el mapa real, haciendo todo más intuitivo
- Fácil de mejorar: La estructura interna ahora permite agregar nuevos lugares sin romper lo que ya funciona.