

Paso 1. Identificación del problema

Definición del problema

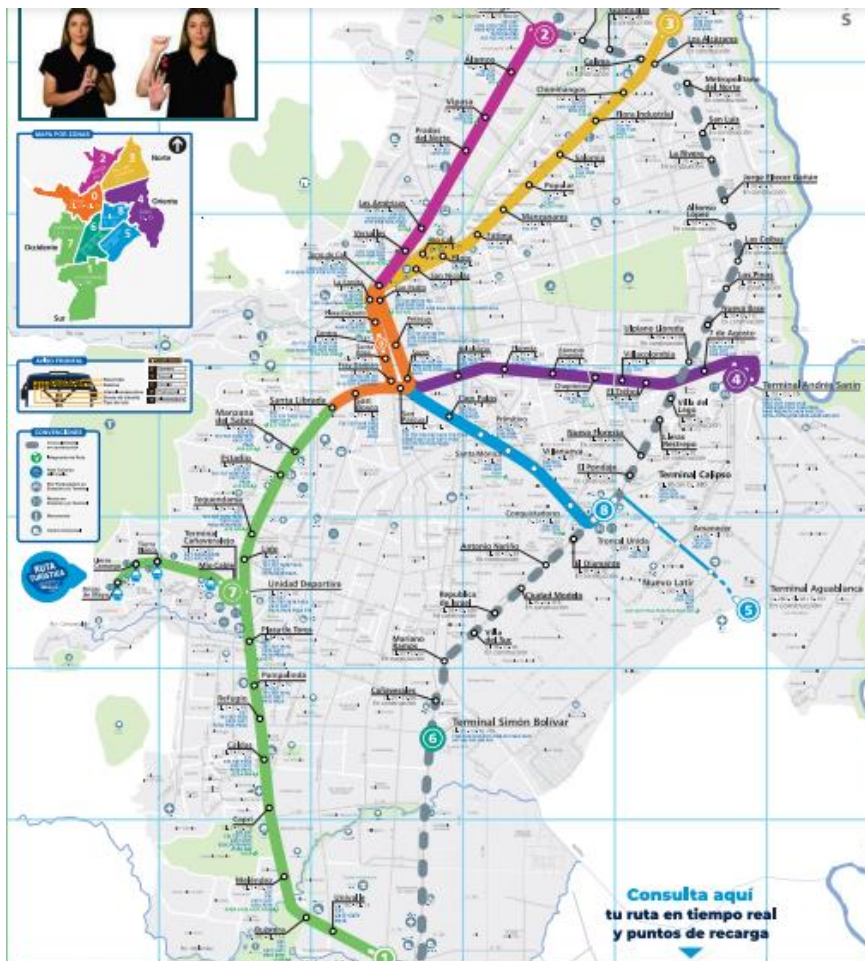
Se requiere un programa que permita encontrar las rutas más rápidas, entre cada una de las estaciones del transporte masivo MIO.

Identificación de necesidades y síntomas

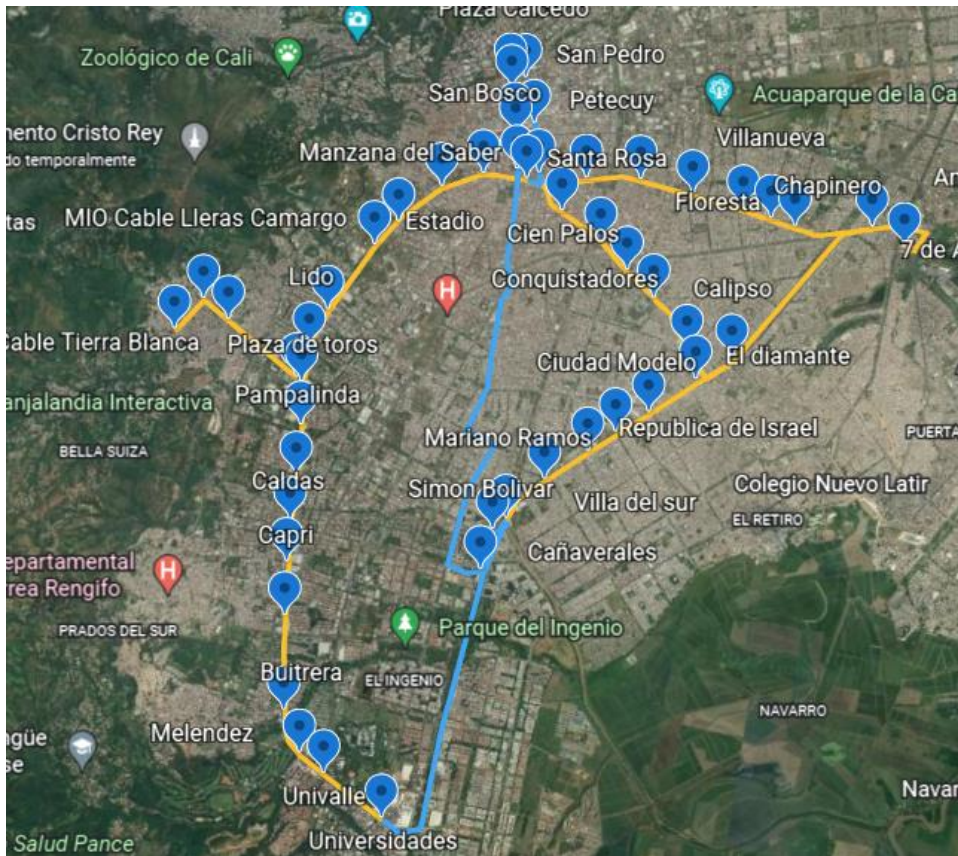
- Se requiere encontrar la ruta con menor recorrido de distancia entre diferentes estaciones del MIO.
- Se requiere encontrar la ruta con menor coste, con respecto a tiempo, entre diferentes estaciones del MIO.
- El programa, por el momento, va a ser usado para generar rutas entre las estaciones del MIO para los buses troncales.
- La solución debe ser eficiente en temas de ejecución.

Paso 2. Recopilación de la información

Mapa actualizado de los puntos embarque del transporte masivo MIO para los buses troncales, petroncal, alimentador y MIOCABLE.

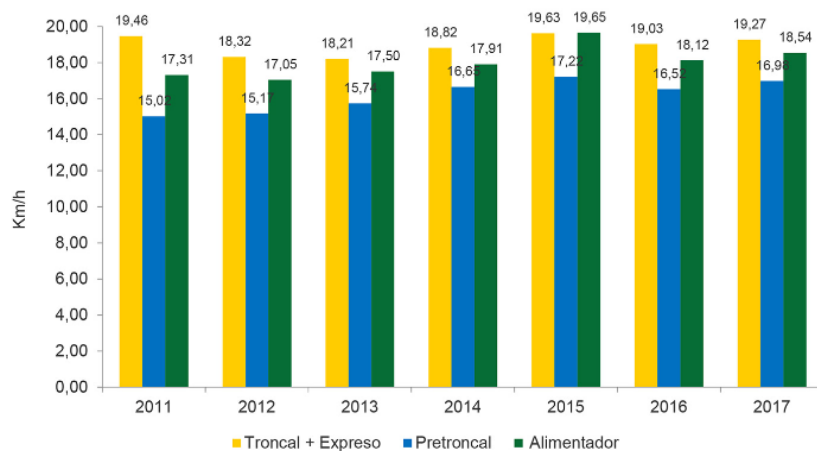


Mapa con direccionamiento y rutas entre diferentes estaciones del MIO que se van a utilizar para la fase de prueba.



Velocidad promedio de los buses troncales.

Los buses del MIO tipo troncales tienen una velocidad promedio de 19,27 km/h.



Paso 3. Búsqueda de soluciones creativas

Alternativa 1:

Grafos

Un grafo es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).

Opciones de uso de la alternativa 1:

- **Alternativas de Recorridos sobre grafos:**
 - **Algoritmo BFS:** Una búsqueda en anchura (BFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo, comenzando en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo), para luego explorar todos los vecinos de este nodo.
 - **Algoritmo DFS:** Una búsqueda en profundidad (DFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo).
- **Alternativas de caminos de peso mínimo:**
 - **Algoritmo de Dijkstra:** El algoritmo de Dijkstra es utilizado para encontrar el camino más corto en un grafo dado $G(V,E)$, donde V es el número de vértices y E es el número de aristas. Además, cada vértice tiene un costo asignado. El objetivo

de este algoritmo es calcular el camino para recorrer el grafo desde un punto “a” a un punto “b” con el menor costo.

- **Algoritmo de Floyd-Warshall:** Es un algoritmo para encontrar las rutas más cortas en un grafo ponderado con pesos de borde positivos o negativos (pero sin ciclos negativos). Lo hace comparando todos los caminos posibles a través del grafo entre cada par de vértices.

- **Alternativas para creación de árboles de caminos mínimos:**

- **Algoritmo de Prim:** El algoritmo de Prim, dado un grafo conexo, no dirigido y ponderado, encuentra un árbol de expansión mínima. Es decir, es capaz de encontrar un subconjunto de las aristas que formen un árbol que incluya todos los vértices del grafo inicial, donde el peso total de las aristas del árbol es el mínimo posible.
- **Algoritmo de Kruskal:** El algoritmo de Kruskal, dado un grafo conexo, no dirigido y ponderado, encuentra un árbol de expansión mínima. Es decir, es capaz de encontrar un subconjunto de las aristas que formen un árbol que incluya todos los vértices del grafo inicial, donde el peso total de las aristas del árbol es el mínimo posible.

Paso 4. Transición de las Ideas a los Diseños Preliminares

Dentro de las opciones que tenemos en el momento solo encontramos una alternativa que asocia con el concepto de encontrar las rutas más rápidas entre puntos, el cual es, la teoría de grafos

Paso 5. Evaluación y Selección de la Mejor Solución

Ahora bien, para la selección de los algoritmos que conforman el grafo hemos utilizado estos criterios para definir las mejores opciones entre cada una de las secciones:

Complejidad temporal: Se describe la cantidad de tiempo que lleva ejecutar un algoritmo. La complejidad temporal se estima comúnmente contando el número de operaciones elementales realizadas por el algoritmo, suponiendo que cada operación elemental requiere una cantidad fija de tiempo

Algoritmo de Dijkstra: $O(n^2)$

Algoritmo de Floyd-Warshall: $O(|V|^3)$

Algoritmo de Prim: $O(n^2)$

Algoritmo de Kruskal: $O(n \log n)$

Algoritmo BFS:

Algoritmo DFS: