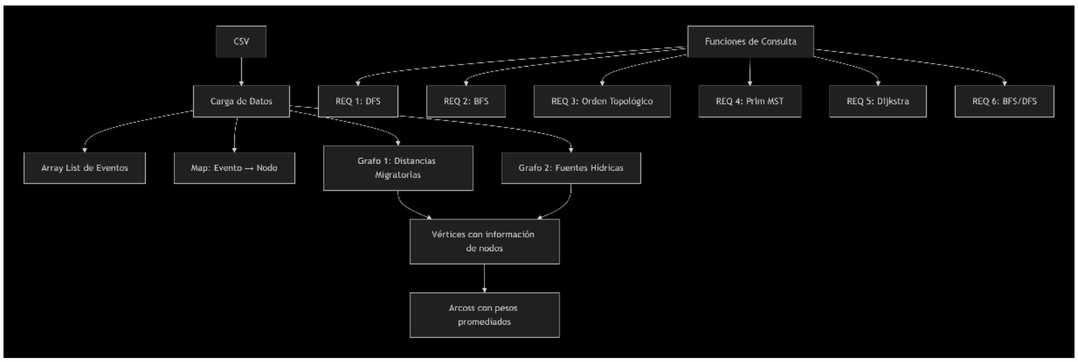
Reto 4

1. Angie Dahana Aponte Ariza, a.apontea@uniandes.edu.co, 202416313.

2. Mateo Ávila Benavides, m.avilab2@uniandes.edu.co, 202410101.

3. Santiago Sarmiento Pinzón, s.sarmientop234@uniandes.edu.co, 202510617.



Requerimiento 1 (Mateo):

O(V + E)

Se deben encontrar los nodos más cercanos a las coordenadas de origen y destino proporcionadas por el usuario lo que tiene una complejidad de O(V). Después, se realiza un DFS desde el nodo de origen hacia el destino con una complejidad de O(V + E) en el peor caso.

Requerimiento 2 (Dahana):

O(V + E)

En el peor caso, el BFS visita todos los vértices y arcos, es decir, O(V + E), además de buscar los nodos mas cercanos a las coordenadas dadas con una complejidad de O(V).

Requerimiento 3 (Santiago):

O(V + E)

Al calcular por medio de DFO el orden topológico del grafo se obtiene una complejidad de O(V + E), siendo esta la complejidad para la detección de ciclos, el orden topológico y al calculo del camino más largo.

Requerimiento 4:
O((V + E) log V)

La complejidad se debe al uso del algoritmo Prim para la construcción de un MST, con una cola de prioridad la complejidad es de O((V + E) log V), después se realiza un recorrido donde se halla el camino de nodos de menor costo con una complejidad de O(V).

Requerimiento 5:

O((V + E) log V)

La complejidad temporal y su explicación es similar a la del requerimiento 4, solo que en este se implementa el algoritmo de Dijkstra, sin embargo, esta comparte, dentro de este caso, la misma complejidad que el algoritmo de Prim.

Requerimiento 6:

O(V log V + E)

Debido al recorrido del grafo se tiene una complejidad de O(V + E), al recorrer cada vértice y arco del grafo. Al necesitar un ordenamiento se tiene una complejidad de O(V log V), al sumar las dos anteriores complejidades se obtiene que la total del requerimiento es igual a O(V log V + E).