**Diseño de proyecto**

**Carga de datos**

* Implementación: Utilizar lectura de archivos de texto para cargar los vértices y los arcos de la maya vial en un grafo dirigido y se utilizan las propiedades del grafo para conocer el número de vértices, de arcos, y el arco y vértice con mayor OBJECTID. Posteriormente se utiliza lectura de archivos JSON para cargar los comparendos y las estaciones policía y guardarlas, cada una, en un alista encadenada, y se va llevando el registro del comparendo y la estación de mayor OBJECTID
* Parámetros de entrada: ruta del archivo Json a leer
* Retorno: Comparendo y estación con mayor OBJECT\_ID
* Orden del algoritmo: N

**Parte Inicial**

**1.**

* Implementación: Se recorre toda el arreglo de vértices hasta encontrar el vértice más cercano a la estación de policía
* Parámetros: Coordenadas de la estación
* Retorno: Vértice más cercano
* Orden: N

**2 y 3.**

* Implementación: Se para cada comparendo se recorre toda el arreglo de vértices buscando el de menor distancia y a ese vértice se le añade le comparendo, y a sus arcos correspondientes se les suma un comparendo a su segundo costo.
* Parámetros: Lista de comparendos Y Grafo
* Retorno: Ninguno
* Orden: N\*M

**4.**

* Implementación: Se utiliza el método del literal 1 sobre cada estación y se asigna a los vértices de retorno una estación
* Parámetros: Lista de estaciones
* Retorno: Ninguno
* Orden: N\*M

**Parte A**

1A.

* Implementación: Se utiliza le algoritmo de camino más corto sobre le grafo dirigido
* Parámetros de entrada: Ubicaciones geográficas
* Retorno: Lista con las coordenadas del camino
* Orden del algoritmo: logN

2A.

* Implementación: Se copian los vértices una cola de prioridad donde la prioridad este dada por la gravedad de sus comparendos y se hacen M eliminaciones que se agregan a una lista encadenada para retornar.
* Parámetros de entrada: Entero del número de vértices a buscar
* Retorno: Lista encadenada con los vértices
* Orden del algoritmo: N\*M

**Parte C**

1C.

* Implementación: Se copian los vértices a una cola de prioridad donde la prioridad es la gravedad de sus comparendos y se eliminan los M mayores. Y para cada comparendo se busca la estación más cercana, luego se busca el camino más corto de la estación al vértice.
* Parámetros: Entero del número de comparendos a atender
* Retorno: Lista de vértices recorridos
* Orden del algoritmo: N\*MlogN

2C.

* Implementación: Se revisan todos los comparendos y se obtienen sus vértices, y para cada uno se busca la estación más cercana y se dibuja y se agrega la información a una lista para retornar.
* Parámetros de entrada: Ninguno
* Retorno: Arreglo de listas encadenadas con la infomacion para dibujar
* Orden del algoritmo: N\*M\*KlogN