**Proyecto 3**

**Entrega del Diseño**

**Requerimientos funcionales**

**Parte Inicial:**

|  |
| --- |
| **Identificador y Título**: Carga de Información |
| **Descripción**:  Realizar la carga de informacion de los archivos: Comparendos\_DEI\_2018\_Bogotá\_D.C.geojson, estacionpolicia.geojsonRevisa y el archivo json con el grafo de la malla vial de Bogotá creado por nosotros. |
| **Datos de Entrada**:  1. Comparendos\_DEI\_2018\_Bogotá\_D.C.geojson  2.estacionpolicia.geojson  3. Archivo Json con el grafo de la malla vial de Bogotá |
| **Datos de Salida**:  1. String con Total de comparendos, información del comparendo con el mayor OBJECTID encontrado, Total de Estaciones de Policía en el archivo, información de la estación de policía con el mayor OBJECTID encontrado, Total de vértices en el grafo de la malla vial de Bogotá, información del vértice con el mayor ID encontrado, Total de arcos en el grafo de la malla vial de Bogotá y la información de los arcos con el mayor ID encontrado. |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).  Se guardará la malla vial como vértices y arcos del grafo y se le agregara a cada vértice un arreglo de comparendos y de ser correspondiente, una estacion de policia.  1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(V+N).** Siendo V la cantidad de vértices y N la cantidad de comparendos.  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

|  |
| --- |
| **Identificador y Título**: 1 Incial- Encontrar el Id del vértice de la malla vial más cercano |
| **Descripción**:  Dada una localización geográfica con latitud y longitud, encontrar el Id del vértice de la  malla vial más cercano por distancia haversiana. |
| **Datos de Entrada**:  1. String latitud.  2. String longitud. |
| **Datos de Salida**:  1. String id del vértice de la malla vial más cercano por distancia haversiana |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).  1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(V).** Siendo V la cantidad de vértices.  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

|  |
| --- |
| **Identificador y Título**:2 Inicial- Adicionar la información de cada uno de los comparendos al grafo de la malla vial |
| **Descripción**:  Se adiciona la información de cada uno de los comparendos del año 2018 al grafo de la malla vial, agregandolo a un arreglo de comparendos que se encuentra en el vértice de la malla vial más cercano a la ubicación geográfica de cada comparendo |
| **Datos de Entrada**:  1. El usuario no ingresa ningún dato. |
| **Datos de Salida**:  1. El usuario no recibe ningún dato. |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).   1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(VN).** Siendo V la cantidad de vértices y N la cantidad de comparendos.  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

|  |
| --- |
| **Identificador y Título**:3 Inicial- Agregar al grafo información de costo. |
| **Descripción**:  Agregar a cada arco del grafo la informacion de su costo, El grafo tendrá 2 costos en sus arcos:  a. El primer costo asociado a un arco es la distancia haversiana (en kilómetros) entre las localizaciones geográficas de los vértices que conecta. Esta distancia fue calculada en el taller 7.  b. El segundo costo asociado a un arco es el total de comparendos entre los vértices que conecta. |
| **Datos de Entrada**:  1. El usuario no ingresa datos de entrada |
| **Datos de Salida**:  1. El usuario no recibe datos de salida |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).  1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(A)** Siendo A la cantidad de arcos.  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

|  |
| --- |
| **Identificador y Título**:4 Inicial-Adicionar la información de cada una de las estaciones de policía al grafo. |
| **Descripción**:  Agregar a cada vértice del grafo la información de cada una de las estaciones de policía al grafo.. Para este fin, ubicamos el vértice de la malla vial más cercano a la ubicación geográfica de cada estación de policía y sobre este vértice almacenamos la información de la estación que considere pertinente. |
| **Datos de Entrada**:  1. El usuario no ingresa datos de entrada |
| **Datos de Salida**:  1. El usuario no recibe datos de salida |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).  1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(EV)** Siendo E el número de estaciones y V el número de vértices.  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

**Requerimientos parte A**

|  |
| --- |
| **Identificador y Título:** 1A. Obtener el camino de costo mínimo entre dos ubicaciones geográficas por distancia |
| **Descripción**:  Se busca el camino de costo mínimo conforme la distancia haversiana de cada arco entre un punto de origen y uno de destino ingresados por el usuario como latitudes y longitudes, las cuales serán aproximadas a los vértices más cercanos en la malla vial. |
| **Datos de Entrada**:  1. String latitud y longitud punto de origen.  2. String latitud y longitud punto de destino. |
| **Datos de Salida**:  1. String con el total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (menor distancia haversiana) y la distancia estimada (sumatoria de distancias harvesianas en Km).  **Visualización mapa**: Camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).  1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(n)**  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

|  |
| --- |
| **Identificador y Título**:2A-Determinar la red de comunicaciones que soporte la instalación de cámaras de video en los M puntos donde se presentan los comparendos de mayor gravedad. |
| **Descripción**:  Para este requerimiento implementaremos **INSERTE TIPO DE BUSQUEDA DE CAMINOS,** lo que nos permitirá establecer una red de comunicaciones eficiente entre los M vértices donde se presentan los comparendos de mayor gravedad. |
| **Datos de Entrada**:  1. Int M sitios requeridos con cámaras. |
| **Datos de Salida**:  1. String el tiempo que toma el algoritmo en encontrar la solución, y la siguiente información de la red propuesta: los vértices (identificadores) y los arcos incluidos, y el costo (monetario) total.  Visualización mapa: Mostramos en un mapa en Google Maps la red de comunicaciones propuesta. Con las M ubicaciones de las cámaras resaltadas y los arcos de la red que las unen. |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no Dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).   1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(Log(n))**  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

**Requerimientos parte B**

|  |
| --- |
| **Identificador y Título:** 1B.Obtener el camino de costo mínimo entre dos ubicaciones geográficas por número de comparendos |
| **Descripción**:  Se busca el camino de costo mínimo conforme el número de comparendos de cada arco entre un punto de origen y uno de destino ingresados por el usuario como latitudes y longitudes, las cuales serán aproximadas a los vértices más cercanos en la malla vial. |
| **Datos de Entrada**:  1. String latitud y longitud punto de origen.  2. String latitud y longitud punto de destino. |
| **Datos de Salida**:  1. String con camino a seguir, el total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (menor distancia haversiana) y la distancia estimada (sumatoria de distancias harvesianas en Km).  **Visualización mapa**: Camino resultante en Google Maps (incluyendo la ubicación de inicio y la ubicación de destino). |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).  1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(n)**  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

|  |
| --- |
| **Identificador y Título**:2B-Determinar la red de comunicaciones que soporte la instalación de cámaras de video en los M puntos donde se presenta el mayor número de comparendos en la ciudad |
| **Descripción**:  Para este requerimiento implementaremos **INSERTE TIPO DE BUSQUEDA DE CAMINOS,** lo que nos permitirá establecer una red de comunicaciones eficiente entre los M vértices donde se presentan el mayor número de comparendos en la ciudad. |
| **Datos de Entrada**:  1. Int M sitios requeridos con cámaras. |
| **Datos de Salida**:  1. String el tiempo que toma el algoritmo en encontrar la solución, y la siguiente información de la red propuesta:el total de vértices en el componente, los vértices (identificadores), los arcos incluidos (Id vértice inicial e Id vértice final) y el costo (monetario) total.  Visualización mapa: Mostramos en un mapa en Google Maps la red de comunicaciones propuesta. Con las M ubicaciones de las cámaras resaltadas y los arcos de la red que las unen. |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no Dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).   1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(Log(n))**  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

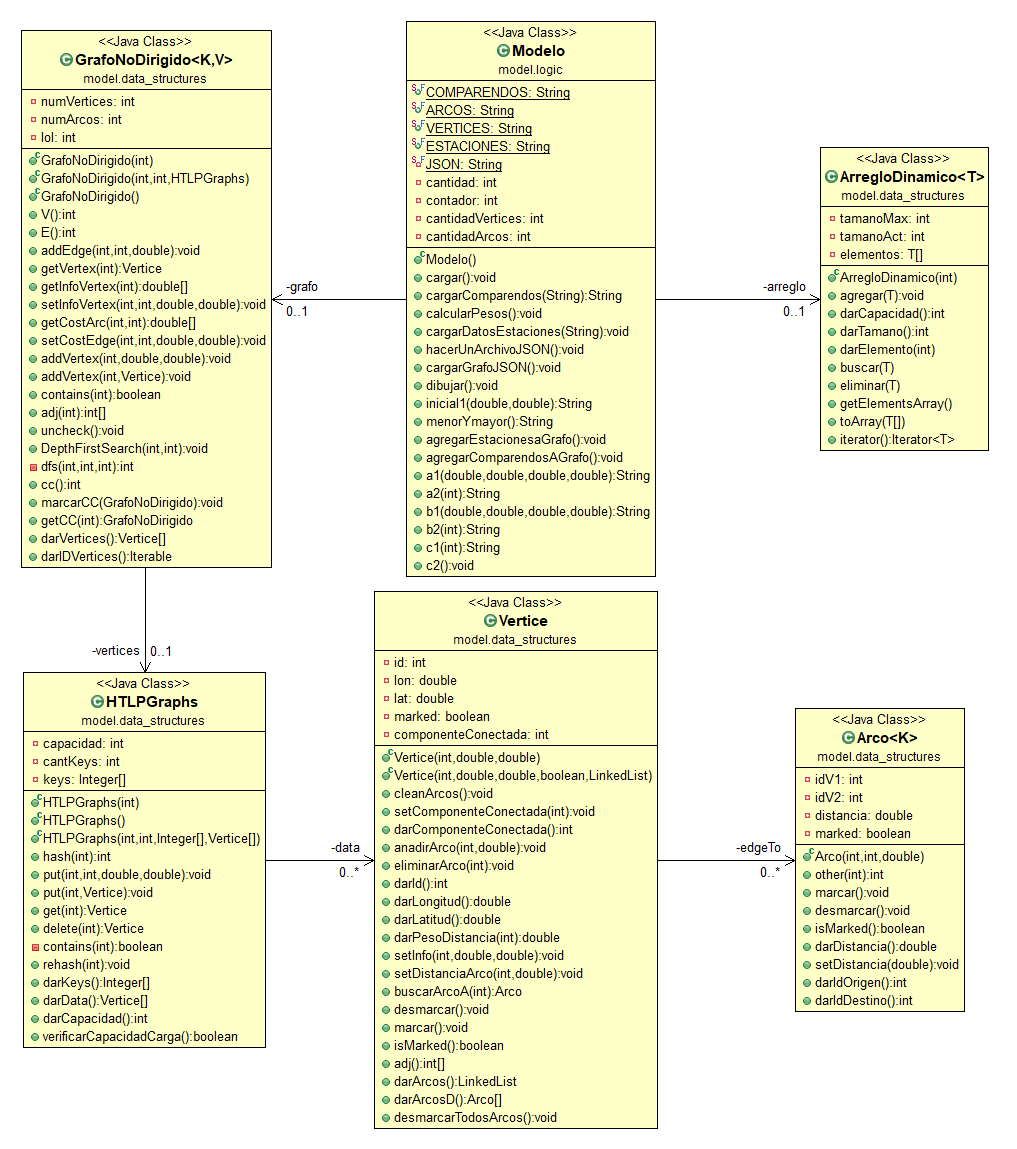
**PARTE C**

|  |
| --- |
| **Identificador y Título**:1C-Obtener los caminos más cortos para que los policías puedan atender los M comparendos más graves. |
| **Descripción**:  Se organizan los M numero de comparendos más graves que se desea atender para generar caminos entre los comparendos y la estación de policía a cada uno para que dicha estación lo atienda. |
| **Datos de Entrada**:  1. Int M numero de comparendos que se quiere atender. |
| **Datos de Salida**:  1. String el tiempo que toma el algoritmo en encontrar la solución, tiempo que toma el algoritmo en encontrar la solución (en milisegundos) y los caminos resultantes para cada uno de los M comparendos: su secuencia de vértices y arcos, así como su costo total (sumatoria de distancia de los arcos en kilómetros).  **Visualización mapa**: Mostramos en un mapa en Google Maps los caminos resultantes diferenciando la ubicación del comparendo (origen) y la ubicación de las estaciones de policía (destinos). Le asignamos un color diferente a cada una de las rutas más eficientes de cada uno de los M comparendos. |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no Dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).   1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(Log(n))**  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

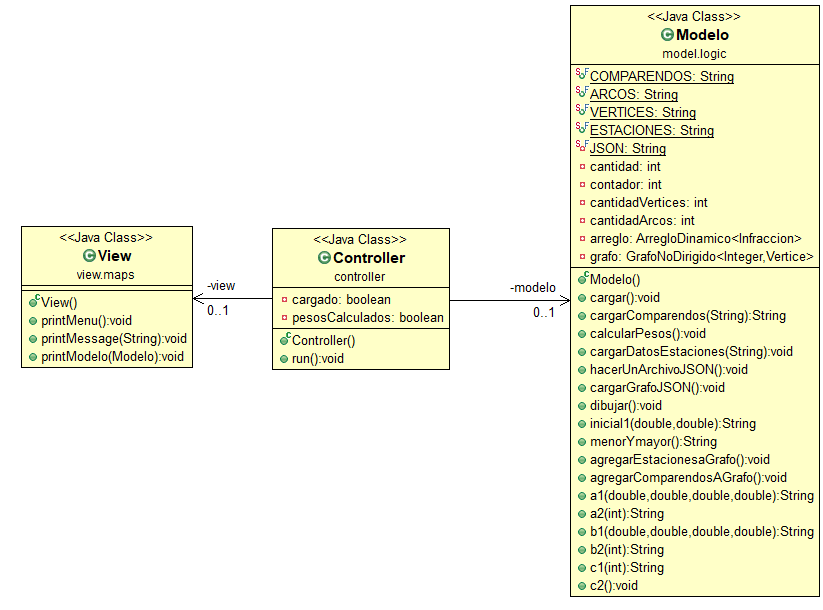
|  |
| --- |
| **Identificador y Título**:2C-Identificar las zonas de impacto de las estaciones de policía. |
| **Descripción**:  Se asigna a cada estación los comparendos que pueda atender en el menor tiempo posible, es decir, a cada estación con la ruta más corta a un comparendo se le asigna este. Se genera un grafo tomando únicamente los vértices y arcos involucrados en dichos caminos. Definimos un esquema JSON para persistir el grafo. Seguido a esto se hace cc a cada uno de los componentes identificados y asignamos el color correspondiente a todos los vértices de cada componente. |
| **Datos de Entrada**:  1. El usuario no ingresa nada. |
| **Datos de Salida**:  1. String el tiempo que toma el algoritmo en encontrar la solución, El número de vértices y arcos que tiene el grafo no dirigido, La cantidad de comparendos que atiende cada una de las estaciones de policía y por cada componente conexa: el color, el ObjectId de las estaciones de policía dentro del componente, y el número de vértices incluidos.  **Visualización mapa**: Mostramos en un mapa en Google Maps un círculo en la posición de cada estación de policía con su área proporcional al porcentaje de comparendos atendidos por dicha estación y el color de su componente conexo. Adicional a esto los arcos del color del componente conexo al cual pertenece el vértice del grafo donde se producen los M primeros comparendos de cada estación. |
| **Estructura(s) Genérica(s) de Datos a utilizar**:  1. Grafo no Dirigido. |
| **Parametrización de Estructura(s) Genérica(s) de Datos a utilizar** (corresponde a decir las clases que se van a utilizar en cada estructura de datos genérica a utilizar).   1. Grafo no dirigido en el cual se guarda la malla vial como vértices y arcos del grafo. |
| **Complejidad Temporal y justificación**:  La complejidad temporal es **O(Log(n))**  Porque esto es lo que se necesita en el peor caso para la implementación de esta estructura. |

**UML**

**UML de las Estructuras:**

****

**UML arquitectura MVC:**

****