

## Listado de Requerimientos

<b>Nombre</b>	<b>1A - Obtener el camino de costo mínimo entre dos ubicaciones geográficas por distancia</b>
<b>Resumen</b>	El usuario debe especificar el punto de origen y destino de un trayecto, usando las coordenadas de latitud y longitud. Dichos puntos deben aproximarse a los vértices más cercanos de la malla vial. Para hallar el costo mínimo de cada camino se toma como referencia la distancia haversiana de cada arco.
<b>Entradas</b>	
La longitud y latitud del punto de origen y el punto de destino del trayecto especificado.	
<b>Resultados</b>	
Información del camino como: el total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (distancia haversiana) y la distancia estimada(km).	
Graficar el camino resultante.	
<b>Complejidad Estimada</b>	E logV (grafo, cola de prioridad y algoritmo de dijkstra)

<b>Nombre</b>	<b>2A - Determinar la red de comunicaciones que soporte la instalación de cámaras de video en los <math>M</math> puntos donde se presentan los comparendos de mayor gravedad.</b>
<b>Resumen</b>	Se deben buscar las ubicaciones donde hayan ocurrido los comparendos de mayor gravedad; público siendo el más grave, seguido de oficial y luego particular. El usuario debe especificar una cantidad $M$ de ubicaciones para colocar cámaras de video y el programa debe hacer uso eficiente de los recursos para colocar estas cámaras en los puntos donde se encuentren los comparendos más graves.
<b>Entradas</b>	
Un número $M$ que determina la cantidad de ubicaciones donde se van a colocar las cámaras.	
<b>Resultados</b>	
Debe mostrar el tiempo que toma el algoritmo en encontrar la solución, e información de la red de cámaras como: los identificadores de los vértices y sus arcos respectivos, y el costo monetario total.	
Graficar la red de cámaras resultante, resaltando las $M$ ubicaciones.	

<b>Complejidad Estimada</b>	E logV (grafo, cola de prioridad, Eager Prim)
-----------------------------	---

<b>Nombre</b>	<b>1B - Obtener el camino de costo mínimo entre dos ubicaciones geográficas por número de comparendos</b>
<b>Resumen</b>	Para determinar el costo de un arco se debe utilizar la cantidad de comparendos. El usuario debe ingresar un punto de origen y un punto de destino mediante las coordenadas de longitud y latitud. Después, el programa debe encontrar el camino de costo mínimo entre dichos puntos.
<b>Entradas</b>	
Las coordenadas (longitud y latitud) del punto de origen y el punto de destino.	
<b>Resultados</b>	
Información del camino resultante como: el total de vértices, sus vértices (Id, latitud, longitud), el costo mínimo (cantidad de comparendos) y la distancia estimada (Km).	
Graficar el camino resultante.	
<b>Complejidad Estimada</b>	E logV (grafo, cola de prioridad y algoritmo de dijkstra)

<b>Nombre</b>	<b>2B - Determinar la red de comunicaciones que soporte la instalación de cámaras de video en los <math>M</math> puntos donde se presenta el mayor número de comparendos en la ciudad.</b>
<b>Resumen</b>	Se deben buscar las ubicaciones donde se hayan registrado la mayor cantidad de comparendos. El usuario debe especificar una cantidad $M$ de ubicaciones para colocar cámaras de video y el programa debe hacer uso eficiente de los recursos para colocar estas cámaras en los puntos donde se hayan registrado la mayor cantidad de comparendos.
<b>Entradas</b>	
Un número $M$ que determina la cantidad de ubicaciones donde se van a colocar las cámaras.	
<b>Resultados</b>	
Mostrar el tiempo que toma el algoritmo en encontrar la solución (milisegundos) e información de la red propuesta como: el total de vértices, los identificadores de los vértices, los arcos incluidos (Id vértice inicial e Id vértice final) y el costo monetario total.	
Graficar la red de cámaras resultante, resaltando las $M$ ubicaciones.	

<b>Complejidad Estimada</b>	E logV (grafo, cola de prioridad, Eager Prim)
-----------------------------	---

<b>Nombre</b>	<b>1C - Obtener los caminos más cortos para que los policías puedan atender los M comparendos más graves.</b>
<b>Resumen</b>	El usuario debe ingresar una cantidad M de comparendos para procesar. Estos comparendos deben ser atendidos por las estaciones de policía más cercanas, de forma que el programa debe indicar cuales son las estaciones que van a atender la cantidad de comparendos especificados.
<b>Entradas</b>	
Un número M que determina la cantidad de comparendos de mayor gravedad a procesar.	
<b>Resultados</b>	
Mostrar el tiempo que toma el algoritmo en encontrar la solución (milisegundos) e información de los caminos resultantes como: la secuencia de vértices y arcos, y el costo total (dado en kilómetros).	
Graficar los caminos resultantes. Cada trayecto desde un comparendo hasta su estación de policía respectiva debe ser graficado con un color diferente.	
<b>Complejidad Estimada</b>	V + E (grafo, cola de prioridad y algoritmo de dijkstra)

<b>Nombre</b>	<b>2C - Identificar las zonas de impacto de las estaciones de policía.</b>
<b>Resumen</b>	Conociendo cual es la estación de policía más cercana a cada uno de los comparendos debe ser posible identificar la cantidad de comparendos atendidos por cada una de estas estaciones. Para ello se debe crear un grafo tomando los vértices y arcos de los caminos creados entre los comparendos y las estaciones, de forma que se puedan identificar los componentes conexos respectivos a cada estación de policía.
<b>Entradas</b>	
<b>Resultados</b>	
Mostrar el tiempo que toma el algoritmo en encontrar la solución (milisegundos), el número de vértices y arcos del grafo no dirigido, la cantidad de comparendos que atiende	

cada una de las estaciones de policía. Por cada componente conectado se debe mostrar el color, el número de vértices y el ObjectId de las estaciones de policía dentro de este.	
Un círculo en la posición de cada estación de policía, cuya área es proporcional al porcentaje de comparendos atendidos. El color del círculo y de sus arcos es el mismo que el del componente conexo a la que pertenecen.	
<b>Complejidad Estimada</b>	V+E (grafo, algoritmo dfs)