

RETO 4: Sistema de Comparendos en Bogotá

Objetivo

Poner en práctica los conceptos aprendidos en clase acerca de las estructuras de datos módulo No. 4 sobre del ADT Graph de tipo dirigido y no-dirigido.

Específicamente se desea:

1. **Utilizar** grafos como estructura de datos principal para resolver consultas y requerimientos.
2. **Integrar** el uso de las estructuras lineales (tablas de hash, Lista, pilas, colas y arboles) a los grafos dirigidos y no dirigidos.
3. **Utilizar** adecuadamente el patrón MVC (Modelo-Vista-Controlador).
4. **Aprender** a cargar y procesar en memoria datos.
5. **Utilizar** adecuadamente el administrador de versiones GIT y GitHub.
6. **Aprender** a trabajar en equipo.

Fecha Límite de Entrega

Máxima fecha de entrega para el **martes 5 de diciembre** antes de la media noche (11:59 p.m. hora BrightSpace).

Contexto

Dado el constante crecimiento de las ciudades en términos de población, infraestructura y servicios, las organizaciones encargadas de su administración se esfuerzan por mantener información actualizada sobre diversos aspectos que impactan la vida de los ciudadanos, como educación, salud, transporte, vivienda, infraestructura, entretenimiento, seguridad y economía. Esta información es crucial para mantener a los ciudadanos informados sobre sus actividades diarias y para que los administradores y autoridades locales tomen decisiones que mejoren la calidad de vida.

El transporte, en particular, es una problemática significativa en las ciudades, con especial atención a la regulación de este, incluyendo las infracciones de tránsito. Estas infracciones son herramientas utilizadas por las autoridades para controlar el comportamiento del transporte, tanto público como privado. El foco de nuestro proyecto se centra en el sistema de comparendos implementado en la ciudad de Bogotá D.C. Utilizaremos el portal oficial de datos abiertos de la ciudad, que recopila información sobre las infracciones al Código Nacional de Tránsito impuestas por la autoridad policial en el año 2019.

En este proyecto, nos proponemos desarrollar una aplicación que facilite la gestión de datos relacionados con la malla vial de Bogotá D.C., utilizando la información de los comparendos y las estaciones de policía disponibles en el repositorio. La malla vial se modelará como un Grafo No Dirigido, y su construcción se basará en la información suministrada en el mencionado repositorio.

Para el desarrollo de este reto tenemos **tres** diferentes fuentes de datos, las cuales expondremos a continuación.

Comparendos

Como se mencionó anteriormente utilizaremos como fuente de información el compilado de infracciones al Código Nacional de Tránsito, impuestas por la autoridad policial, generadas en el año 2019 en la ciudad de Bogotá D.C. Estos datos se deben cargar del archivo **Comparendos_2019_Bogota_D_C.geojson**.

La fuente original de datos contiene el registro de cada uno de los comparendos generados para el año 2019. En la Tabla 1 puede verse el resumen de los campos de datos disponibles de cada comparendo. En donde se incluyen propiedades cruciales como: el identificador del comparendo, el número del comparendo, la longitud y latitud del comparendo, la fecha y hora del comparendo, el mes, el medio de detección, y la clase de vehículo, entre otros.

Nombre	Descripción
X	Coordenada X (longitud) del lugar donde se registra la infracción.
Y	Coordenada Y (latitud) del lugar donde se registra la infracción.
OBJECTID	Identificador único del registro.
NUM_COMPARENDO	Número único asignado a cada registro de infracción.
FECHA_HORA	Fecha y hora en la que se registró la infracción.
ANO	Año en el que ocurrió la infracción.
HORA_OCURRENCIA	Hora en la que ocurrió la infracción.
MES	Mes en el que ocurrió la infracción.
MEDIO_DETECCION	Medio por el cual se detectó la infracción.
CLASE_VEHICULO	Clase o tipo de vehículo involucrado en la infracción.
TIPO_SERVICIO	Tipo de servicio relacionado con el vehículo involucrado en la infracción.
INFRACCION	Código de la infracción cometida.
DES_INFRACCION	Descripción detallada de la infracción.
LOCALIDAD	Localidad donde ocurrió la infracción.
MUNICIPIO	Municipio donde ocurrió la infracción.
LATITUD	Valor de latitud del lugar donde se registró la infracción.
LONGITUD	Valor de longitud del lugar donde se registró la infracción.
GlobalID	Identificador único global asignado a cada registro, utilizado para propósitos de identificación.

Tabla 1. Descripción de los campos del archivo GEOJSON de comparendos.

Estaciones de Policía

Una Estación de Policía es la unidad básica de la organización policial cuya jurisdicción corresponde a cada municipio, en el que se divide el territorio nacional, sin perjuicio que en un Municipio funcionen varias estaciones. En Bogotá las estaciones corresponden a las localidades que integran el Distrito Capital.

La información de las estaciones de policía se debe construir a partir de la carga del archivo **estacionpolicia.json** (formato JSON), información provista por la Secretaría Distrital de Seguridad, Convivencia y Justicia que se pueden descargar del portal Web de consulta de datos abiertos¹.

La fuente original de datos contiene el registro de cada una de las estaciones de policía actualizadas a Julio de 2023. En la Tabla 2 puede verse el resumen de los campos de datos disponibles de cada estación.

Nombre	Descripción
coordenadas [0]	Coordenada de longitud para la estación de policía
coordenadas [1]	Coordenada de latitud para la estación de policía
OBJECTID	Identificador único para la estación de policía
EPOCOD_PLAN	Código de planificación para la estación de policía
EPOCOD_ENT	Código de entidad para la estación de policía
EPOCOD_PROY	Código de proyecto para la estación de policía
EPOANIO_GEO	Año geográfico de la información de la estación de policía
EPOFECHA_INI	Fecha de inicio de operaciones de la estación de policía
EPOFECHA_FIN	Fecha de finalización de operaciones de la estación de policía
EPODESCRIP	Descripción o nombre de la estación de policía
EPOEST_PROY	Estado del proyecto de la estación de policía
EPOINTERV_ESP	Información de intervención especial en el espacio público por parte de la estación de policía
EPODIR_SITIO	Dirección o ubicación de la estación de policía
EPOCOD_SITIO	Código de sitio de la estación de policía
EPOLATITUD	Coordenada de latitud de la estación de policía
EPOLONGITU	Coordenada de longitud de la estación de policía
EPOSERVICIO	Descripción de los servicios proporcionados por la estación de policía
EPOHORARIO	Horario operativo de la estación de policía
EPOTELEFON	Número de teléfono de contacto para la estación de policía
EPOCELECTR	Contacto de correo electrónico para la estación de policía
EPOCONTACT	Información de contacto para la Policía Nacional
EPOPWEB	Enlace al sitio web de la estación de policía
EPOIUPLAN	Código de unidad de planificación para la estación de policía
EPOIUSCATA	Código de catálogo urbano para la estación de policía
EPOIULOCAL	Código local para la estación de policía
EPOEASOCIA	Información asociada para la estación de policía
EPOFUNCION	Función o propósito de la estación de policía
EPOTEQUIPA	Equipamiento disponible o utilizado por la estación de policía
EPONOMBRE	Nombre de la estación de policía
EPOIDENTIF	Código de identificación para la estación de policía
EPOFECHA_C	Fecha de creación de los datos de la estación de policía

Tabla 2. Descripción de los campos del archivo JSON para las estaciones de policía.

¹ <https://datosabiertos.bogota.gov.co/dataset/estacion-de-policia-para-bogota-d-c>


Malla Vial de Bogotá

La malla vial de Bogotá, representada como un grafo donde los vértices son las intersecciones de las calles y los arcos las calles, puede ser utilizada en una gran cantidad de aplicaciones para navegación terrestre. A continuación, se mostrarán unos ejemplos con el contenido de los archivos de *bogota_vertices.txt* y *bogota_arcos.txt*.

Se representarán las intersecciones de la malla vial como los vértices del grafo. Estas intersecciones están representadas en el archivo *bogota_vertices.txt* con id, longitud, y latitud:

```
0, -74.08921298299998,4.582989396000016
1, -74.08952746199998,4.582560966000017
2, -74.093892202,4.576679366000008
3, -74.09408026199998,4.576433936999991
...
```

Las vías de la malla vial que conectan las intersecciones están almacenadas en el archivo *bogota_arcos.txt*. Estas vías se representan como un conjunto de adyacencias donde el primer campo especifica el id del vértice origen, y los demás campos son los ID de sus vértices adyacentes nuevos:

	<pre>0 1 733 # el vértice 0 tiene como adyacentes a los vértices 1 y 733 1 49 # el vértice 1 tiene como adyacentes "nuevos" al vértice 49. El vértice 1 No tiene al vértice 0 porque el vértice 0 ya había declarado al vértice 1 como adyacente.</pre>
--	--

Trabajo Propuesto

Parte 1: Configuración Repositorio

Complete los siguientes pasos para configurar su repositorio de trabajo:

1. Cree en GitHub un repositorio basado en la plantilla propuesta para el reto, el cual se encuentra en el URL: <https://github.com/ISIS1225DEVs/Reto4-Template>
2. Renombre el repositorio de su reto con el formato **Reto4-G<<Número del grupo>>** ej.: **Reto4-G01** para el grupo 1 de la sección 2.
3. Edite el **README** del repositorio e incluya los nombres completos, correo Uniandes y códigos de los miembros del equipo de trabajo.
4. Realice el procedimiento según lo aprendido en clase para clonar el repositorio en su máquina local y sincronizarlo con su repositorio en GitHub.
5. Descargue los datos desde la sección unificada del curso y cópielos en la carpeta **data** del repositorio local.

Parte 2: Carga de Datos

En la sección unificada, en el apartado de Reto 4 encontrarán los datos oficiales del proyecto. Para responder a los requerimientos presentados deberán cargar la información de los archivos entregados; recuerde que solo se permite leer una vez los datos de cada archivo.

Para la creación del grafo tenga en cuenta las siguientes indicaciones:

- Construya el grafo inicial tomando como base la información provista de la malla vial de Bogotá
- Adicionar la información de cada una de las estaciones de policía al grafo.
 - Para este fin, ubique el vértice de la malla vial **más cercano**² a la ubicación geográfica de cada estación de policía y sobre este vértice almacene la información de la estación que considere pertinente.
- Adicionar la información de cada uno de los comparendos del año 2019 al grafo de la malla vial.
 - Para este fin, ubique el vértice de la malla vial **más cercano** a la ubicación geográfica de cada comparendo y sobre este vértice almacene la información del comparendo que considere relevante.

² Para determinar el vértice más cercano a una ubicación geográfica con latitud y longitud, utilice el vértice que presente la menor distancia Haversine al punto dado.

- Agregar información de costo a los arcos del grafo.
 - El grafo tendrá 2 costos en sus arcos:
 - El primer costo asociado a un arco es la **distancia Haversine**^{3,4,5,6} (en kilómetros) entre los vértices que conecta.
 - El segundo costo asociado a un arco es el **total de comparendos** entre los vértices que conecta.

Al final de la carga hay que reportar los siguientes datos:

- El total de comparendos cargados en la aplicación.
- Mostrar los primeros cinco y últimos cinco comparendos con las siguientes características:
 - El ID del comparendo.
 - La latitud.
 - La longitud.
 - La fecha del comparendo.
 - El medio de Detección.
 - La clase de Vehículo.
 - El tipo de Servicio.
 - La infracción.
 - La descripción de la Infracción.
- El total de estaciones de policía cargadas en la aplicación.
- Mostrar las primeras cinco y últimas cinco estaciones de policía cargadas con las siguientes características:
 - El ID de la estación.
 - El nombre de la estación.
 - La latitud.
 - La longitud.
 - La descripción.
 - La dirección.
 - El servicio.
 - El horario.
 - El teléfono.
 - El correo Electrónico.
- El total de vértices de la malla vial cargadas en la aplicación.
- Mostrar los primeros cinco y últimos cinco vértices cargados con las siguientes características:
 - El ID del vértice.
 - La latitud.
 - La longitud.

³ Haversine formula, URL: https://en.wikipedia.org/wiki/Haversine_formula

⁴ Calculate distance, bearing and more between Latitude/Longitude points, URL: <https://www.movable-type.co.uk/scripts/latlong.html>

⁵ Haversine Formula in Python (Bearing and Distance between two GPS points), URL: <https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points>

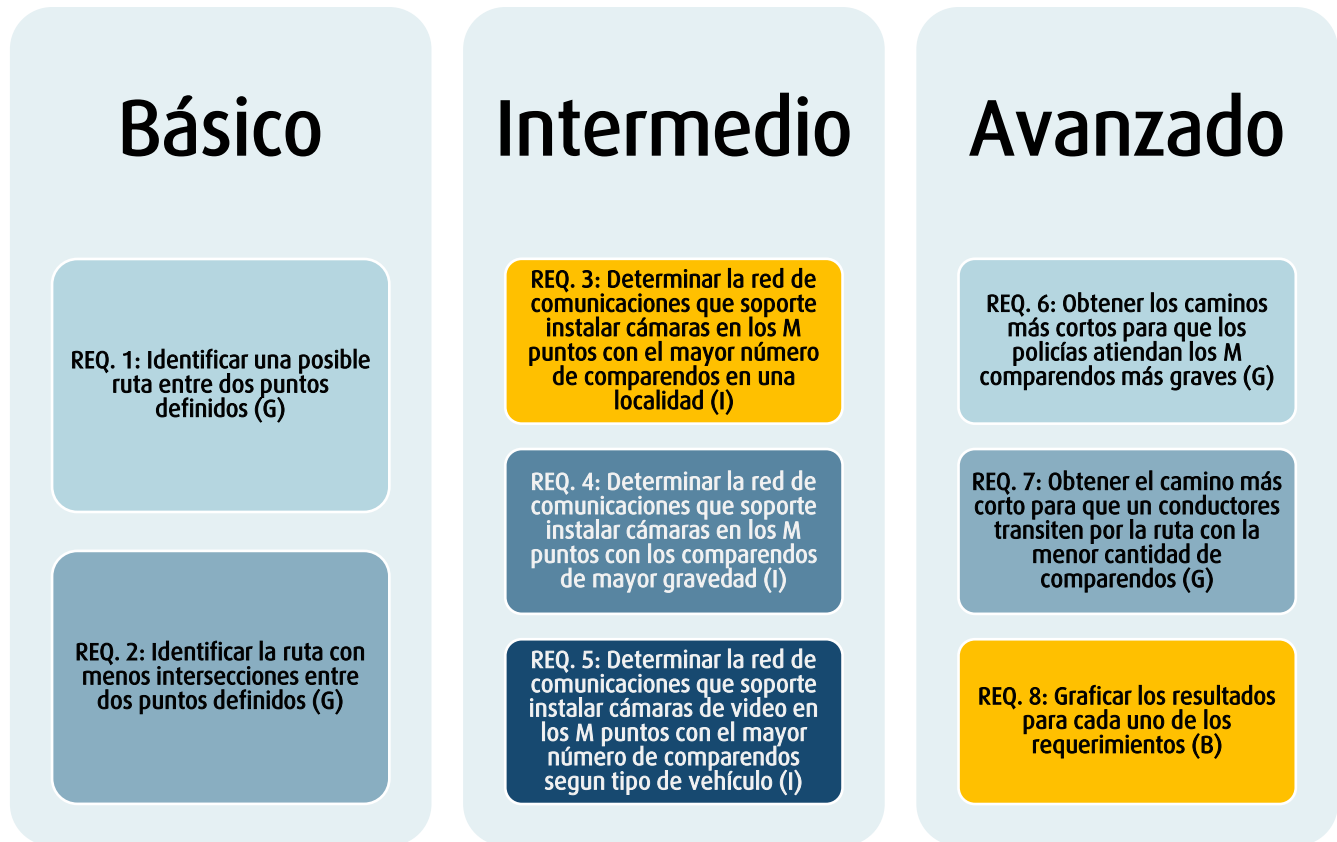
⁶ Haversine 2.5.1, URL: <https://pypi.org/project/haversine/>

- La zona geográfica que define “los límites” de la ciudad por: la longitud mínima, la longitud máxima, la latitud mínima y la latitud máxima en la malla vial.
- El total de arcos de la malla vial cargadas en la aplicación.
- Mostrar los primeros cinco y últimos cinco arcos de la lista de adyacencia dentro del grafo con las siguientes características:
 - Id del vértice inicial.
 - Lista de vértices adyacentes al vértice inicial.

Parte 3: Desarrollo de los Requerimientos

El resumen de los requerimientos se muestra en la siguiente tabla y se explican detalladamente en la siguiente sección.

Para este reto se han identificado ocho (8) requerimientos: siete (7) obligatorios y un (1) bono. Divididos de acuerdo con su dificultad en nivel básico, intermedio y avanzado. Además, es importante resaltar que tres (3) de estos requerimientos se deben desarrollar de forma individual y los cinco (5) restantes son grupales.



NOTA: En caso de que el equipo este conformado solamente por dos integrantes deberán resolver solamente cinco (5) requerimientos obligatorios; el equipo podrá escoger cuál de los requerimientos básicos va a realizar y la pareja podrá elegir dos (2) de los tres (3) requerimientos individuales a implementar.

Requerimiento No. 1 (Grupal): Identificar una posible ruta entre dos puntos definidos

Como conductor **deseo** encontrar un posible camino entre dos puntos de geográficos localizados en los límites de la ciudad de Bogotá.

El punto de origen y destino son ingresados por el usuario como latitudes y longitudes (debe validarse que dichos puntos se encuentren dentro de los límites encontrados de la ciudad). Estas ubicaciones deben aproximarse a los vértices más cercanos en la malla vial.

Los **parámetros de entrada** de este requerimiento son:

- Punto de origen (una localización geográfica con latitud y longitud).
- Punto de destino (una localización geográfica con latitud y longitud).

La **respuesta esperada** debe contener la siguiente información:

- La distancia total que tomará el camino entre el punto de origen y el de destino.
- El total de vértices que contiene el camino encontrado.
- La secuencia de vértices (sus identificadores) que componen el camino encontrado

EJEMPLO Deseo buscar una ruta entre los puntos

Origen:

- Latitud: 4.60293518548777,
- Longitud: -74.06511801444837

• **Destino:**

- **Latitud:** 4.693518613347496,
- **Longitud:** -74.13489678235523

Requerimiento No. 2 (Grupal): Identificar la ruta con menos intersecciones entre dos puntos definidos

Como conductor **deseo** encontrar *un posible camino "más corto", según el número de intersecciones a cruzar*, entre dos puntos de geográficos localizados en los límites de la ciudad de Bogotá.

El punto de origen y destino son ingresados por el usuario como latitudes y longitudes (debe validarse que dichos puntos se encuentren dentro de los límites encontrados de la ciudad). Estas ubicaciones deben aproximarse a los vértices más cercanos en la malla vial.

Los **parámetros de entrada** de este requerimiento son:

- Punto de origen (una localización geográfica con latitud y longitud).
- Punto de destino (una localización geográfica con latitud y longitud).

La **respuesta esperada** debe contener la siguiente información:

- La distancia total que tomará el camino entre el punto de encuentro de origen y el de destino.
- El total de vértices que contiene el camino encontrado.
- La secuencia de vértices (sus identificadores) que componen el camino encontrado

EJEMPLO Deseo buscar una ruta entre los puntos

- | | |
|--------------------------------|--------------------------------|
| • Origen: | • Destino: |
| ○ Latitud: 4.60293518548777, | ○ Latitud: 4.693518613347496, |
| ○ Longitud: -74.06511801444837 | ○ Longitud: -74.13489678235523 |

Requerimiento No. 3 (Individual): Determinar la red de comunicaciones que soporte instalar cámaras en los M puntos con el mayor número de comparendos en una localidad

Como empleados del distrito **deseo** instalar una red de comunicaciones por fibra óptica de cámaras de video en M sitios. Sin embargo, se requiere que esta red tenga el menor costo de instalación posible. Recuerde que el costo de instalación de la fibra óptica es de \$ 1.000.000 COP/km.

Para que la red sea eficiente se seleccionan como puntos de supervisión los M vértices con el **mayor número de comparendos para localidad** definida por el usuario.

Los **parámetros de entrada** de este requerimiento son:

- La cantidad de cámaras de video que se desean instalar (M).
- La localidad donde se desean instalar.

La **respuesta esperada** debe contener la siguiente información:

- El tiempo que se demora algoritmo en encontrar la solución (en milisegundos).
- La siguiente información de la red de comunicaciones propuesta:
 - El total de vértices de la red.
 - Los vértices incluidos (identificadores).
 - Los arcos incluidos (Id vértice inicial e Id vértice final).
 - La cantidad de kilómetros de fibra óptica extendida.
 - El costo (monetario) total.

EJEMPLO: Deseo instalar 20 cámaras en la localidad de Chapinero.

Requerimiento No. 4 (Individual): Determinar la red de comunicaciones que soporte instalar cámaras en los M puntos con los comparendos de mayor gravedad

Como empleados del distrito **deseo** instalar una red de comunicaciones por fibra óptica de cámaras de video en M sitios. Sin embargo, se requiere que esta red tenga el menor costo de instalación posible. Recuerde que el costo de instalación de la fibra óptica es de \$ 1.000.000 COP/km.

Para que la red sea eficiente se seleccionan como puntos de supervisión los M vértices con los comparendos de **mayor gravedad**. Para saber si un comparendo es más grave que otro primero se debe tener en cuenta los siguientes parámetros:

- **El tipo de servicio:** Diplomático es más grave que Oficial, Oficial es más grave que Público, y Público es más grave que Particular.
- Si dos comparendos tienen el mismo tipo de servicio se compara el **código de la infracción** usando el orden lexicográfico (ej. A12 es más grave que A11 y B10 es más grave que A10).

La **respuesta esperada** debe contener la siguiente información:

- El tiempo que se demora algoritmo en encontrar la solución (en milisegundos).
- La siguiente información de la red de comunicaciones propuesta:
 - El total de vértices de la red.
 - Los vértices incluidos (identificadores).
 - Los arcos incluidos (Id vértice inicial e Id vértice final).
 - La cantidad de kilómetros de fibra óptica extendida.
 - El costo (monetario) total.

EJEMPLO: Deseo instalar 15 cámaras para detectar los comparendos de mayor gravedad.

Requerimiento No. 5 (Individual): Determinar la red de comunicaciones que soporte instalar cámaras de video en los M puntos con el mayor número de comparendos según tipo de vehículo

Como empleados del distrito **deseo** instalar una red de comunicaciones por fibra óptica de cámaras de video en M sitios. Sin embargo, se requiere que esta red tenga el menor costo de instalación posible. Recuerde que el costo de instalación de la fibra óptica es de \$ 1.000.000 COP/km.

Para que la red sea eficiente se seleccionan como puntos de supervisión los M vértices con el **mayor número de comparendos** para una **clase de vehículo** definido por el usuario.

Los **parámetros de entrada** de este requerimiento son:

- La cantidad de cámaras de video que se desean instalar (M), y
- Clase de vehículo.

La **respuesta esperada** debe contener la siguiente información:

- El tiempo que se demora algoritmo en encontrar la solución (en milisegundos).
- La siguiente información de la red de comunicaciones propuesta:
 - El total de vértices de la red.
 - Los vértices incluidos (identificadores).
 - Los arcos incluidos (Id vértice inicial e Id vértice final).
 - La cantidad de kilómetros de fibra óptica extendida.
 - El costo (monetario) total.

EJEMPLO: Deseo instalar 10 cámaras para detectar los comparendos de motocicletas.

Requerimiento No. 6 (Grupal): Obtener los caminos más cortos para que los policías atiendan los M comparendos más graves

Como empleados del distrito **deseo** mejorar los tiempos de respuesta de la policía ante los comparendos. Se espera que cada ubicación de un **comparendo grave** debe ser atendida por la estación de policía más cercana (mínimo desplazamiento en kilómetros usando la malla vial).

Recuerde que para saber si un comparendo es más grave que otro primero se debe tener en cuenta los siguientes parámetros:

- **El tipo de servicio:** Público es más grave que Oficial y Oficial es más grave que Particular;
- Si dos comparendos tienen el mismo tipo de servicio se compara el **código de la infracción** usando el orden lexicográfico (ej. A12 es más grave que A11 y B10 es más grave que A10).

Los **parámetros de entrada** de este requerimiento son:

- La cantidad de comparendos que se desea responder (M)
- La estación de policía más cercana al comparendo más grave.

La respuesta **esperada** debe contener la siguiente información:

- El tiempo que se demora algoritmo en encontrar la solución (en milisegundos).
- La siguiente información de cada uno de los caminos seleccionados:
 - El total de vértices del camino.
 - Los vértices incluidos (identificadores).
 - Los arcos incluidos (Id vértice inicial e Id vértice final).
 - La cantidad de kilómetros del camino.

Recomendaciones:

- Asuma que cualquier policía en una estación puede atender un comparendo y que inicialmente todos los policías están en sus estaciones.

EJEMPLO: deseo identificar las rutas más cortas para los 20 comparendos más graves de la ciudad.

Requerimiento No. 7 (Grupal): Obtener el camino más corto para que los conductores transiten por la ruta con la menor cantidad de comparendos

Como conductor **deseo** encontrar el *camino “más corto” en términos número de menor cantidad de comparendos* entre dos puntos de geográficos localizados en los límites de la ciudad de Bogotá.

El punto de origen y destino son ingresados por el usuario como latitudes y longitudes (debe validarse que dichos puntos se encuentren dentro de los límites encontrados de la ciudad). Estas ubicaciones deben aproximarse a los vértices más cercanos en la malla vial.

Los **parámetros de entrada** de este requerimiento son:

- Punto de origen (una localización geográfica con latitud y longitud).
- Punto de destino (una localización geográfica con latitud y longitud).

La **respuesta esperada** debe contener la siguiente información:

- La siguiente información del camino seleccionado:
 - El total de vértices del camino.
 - Los vértices incluidos (identificadores).
 - Los arcos incluidos (Id vértice inicial e Id vértice final).
 - La cantidad de comparendos del camino.
 - La cantidad de kilómetros del camino.

EJEMPLO Deseo buscar una ruta entre los puntos

- | | |
|--------------------------------|--------------------------------|
| • Origen: | • Destino: |
| ○ Latitud: 4.60293518548777, | ○ Latitud: 4.693518613347496, |
| ○ Longitud: -74.06511801444837 | ○ Longitud: -74.13489678235523 |

Requerimiento No. 8 (Bono Grupal): Graficar los resultados para cada uno de los requerimientos

Con la finalidad de facilitar el análisis e interpretación de resultados, se otorgará bono a los equipos de trabajo que grafiquen los resultados de los requerimientos obligatorios enunciados (del primer al séptimo) con recursos multimedia e interfaz gráfica.

Para completar este requerimiento recomendamos utilizar la librería por extensión de Python llamada **"folium"** que se puede instalar en su ambiente por medio del comando **"pip install folium"**.

Para más información sobre esta librería dirigirse a los siguientes enlaces:

- Enlace oficial de PYPI, URL: <https://pypi.org/project/folium/>
- Enlace oficial de la librería, URL: <https://github.com/python-visualization/folium>
- Enlace al tutorial de la Librería. URL: <https://python-visualization.github.io/folium/quickstart.html>

Parte 4: Análisis de resultados

Dentro del proyecto debe incluir un documento en la carpeta **Docs** en formato **PDF** donde se evidencie el análisis de complejidad, las pruebas de tiempos de ejecución y de memoria utiliza para cada requerimiento. Se sugiere que el documento tenga la siguiente distribución del contenido:

- Nombres, código y correo Uniandes de los integrantes del grupo.
- Para los requerimientos individuales se debe indicar que estudiante del equipo lo realizó.
- Un diagrama de las estructuras de datos que usó para resolver los diferentes requerimientos y que se crean al realizar la carga de datos.
- Descripción de las funciones y código implementado en la carga de datos y en cada uno de los requerimientos (incluye el bono si aplica).
- Análisis de complejidad temporal de cada uno de los requerimientos en **Notación O** (incluye el bono si aplica).

Recomendaciones:

- Tomen como guía las herramientas, metodología y análisis realizados en los laboratorios; en especial los laboratorios 10 y 11.

Recomendaciones para la Entrega

Para realizar la entrega del reto deben:

1. Agregar los usuarios de los monitores y profesores del curso a su organización de GitHub para hacer la entrega adecuada de la actividad,
2. Dar permisos adecuados repositorio a los monitores y al profesor, de lo contrario el taller **NO** podrá ser calificado,
3. Asegurarse que la visibilidad del repositorio entregado sea **privada** y que solo pueda accederse con los permisos configurados para los integrantes del grupo,
4. Enviar el enlace de GitHub en la actividad correspondiente dentro de Bloque Neón Uniandes (BrightSpace).
5. Incluir en el repositorio GIT todo el material, código y documentos solicitados durante la actividad.

IMPORTANTE: Recuerde que solo se calificará el material hasta el último **COMMIT** realizado previo a la fecha/horas definidas en la sección **Fecha Límite de Entrega**.