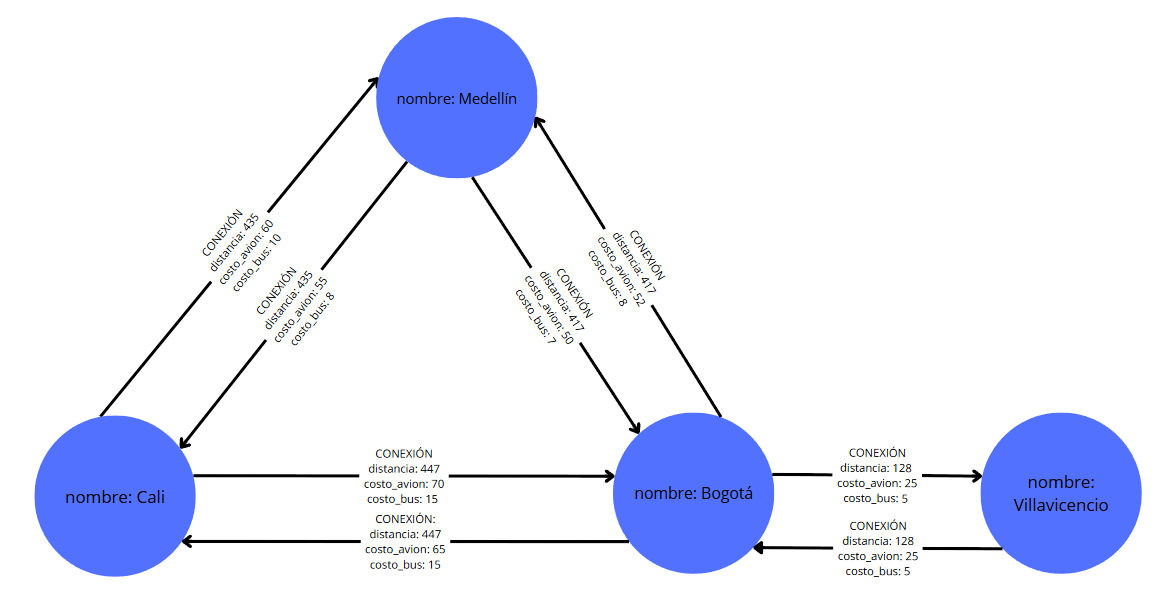
**Trabajo 3 - Bases de Datos 2 (10%)**

En una base de datos de grafos (en Neo4j), se tienen nodos con etiqueta LUGAR, estos nodos solo poseen un atributo: el nombre del lugar. Entre estos nodos se establecen relaciones etiquetadas como CONEXIÓN, y estas relaciones tienen los siguientes atributos: distancia (la distancia en kilómetros entre los dos lugares), costo\_avion (en dólares) y costo\_bus (en dólares). Un ejemplo de esta estructura es la siguiente:



Note que los lugares están conectados en las dos direcciones, la distancia es la misma en ambas direcciones, pero los costos de avión y bus pueden variar; Neo4j permite hacer estas relaciones bidireccionales en instrucciones separadas, mostrando los grafos de manera similar al ejemplo.

En otra base de datos de documentos (en MongoDB), se tienen dos colecciones:

La primera colección es de usuarios, esta guarda documentos JSON con la siguiente estructura: nombre (el nombre del usuario), cod (la identificación del usuario) y dinero\_disponible (dinero con el que cuenta el usuario, en dólares).

Ejemplo:

[

{

nombre: 'Lauren Mayberry',

cod: 10,

dinero\_disponible: 500

},

{

nombre: 'Hayley Williams',

cod: 5,

dinero\_disponible: 600

},

{

nombre: 'Dua Lipa',

cod: 20,

dinero\_disponible: 10

},

{

nombre: 'Carmen Electra[[1]](#footnote-1)',

cod: 15,

dinero\_disponible: 50

}

]

Y otra colección llamada viajes\_deseados, esta colección tiene la siguiente estructura: usuario (que es el código del usuario asociado, o sea el usuario que desea este viaje), nom\_lugar\_inicio (el nombre del lugar de partida del viaje del usuario) nom\_lugar\_destino (el nombre del lugar de destino del viaje del usuario). Por simplicidad, se entiende que un usuario no tiene deseos repetidos.

Ejemplo:

[

{

usu: 10,

nom\_lugar\_inicio: 'Cali',

nom\_lugar\_destino: 'Villavicencio'

},

{

usu: 10,

nom\_lugar\_inicio: 'Cali',

nom\_lugar\_destino: 'Bogotá'

},

{

usu: 5,

nom\_lugar\_inicio: 'Cali',

nom\_lugar\_destino: 'Bogotá'

},

{

usu: 5,

nom\_lugar\_inicio: 'Medellín',

nom\_lugar\_destino: 'Bogotá'

},

{

usu: 20,

nom\_lugar\_inicio: 'Villavicencio',

nom\_lugar\_destino: 'Medellín'

},

{

usu: 15,

nom\_lugar\_inicio: 'Villavicencio',

nom\_lugar\_destino: 'Medellín'

},

]

1. Dado el código de un usuario y un medio de transporte (bus o avión), para cada uno de sus deseos de viaje obtener la ruta más barata en cuanto a costo.

**Ejemplo:** Supongamos que el usuario Lauren Mayberry con cod=10 quiere conocer las rutas más baratas de sus viajes deseados viajando en bus, según la colección viajes\_deseados, ella desea viajar de **Cali a Villavicencio**, y de **Cali a Bogotá**, entonces al ingresar su código y bus, se obtiene que la ruta más barata de **Cali a Villavicencio** en bus es

Cali -> Bogotá -> Villavicencio, es 15+5=20. Otra posible ruta sería

Cali -> Medellín -> Bogotá -> Villavicencio, pero esta no se muestra, debido a que no es la más barata (bus: 10+7+5=22).

También se debe mostrar el resultado correspondiente a su otro deseo (Cali a Bogotá).

1. Dado el código de un usuario, para cada uno de sus deseos de viaje obtener la ruta más corta en cuanto a distancia.

**Ejemplo:** Similar al punto anterior. Supongamos que Hayley Williams con cod=5 quiere conocer las rutas más cortas de sus viajes deseados, según la colección viajes\_deseados, ella desea viajar de **Cali a Bogotá** y de **Medellín a Bogotá**, entonces al ingresar su código, se obtiene que la ruta más corta de Cali a Bogotá es la directa: Cali -> Bogotá pues la distancia es 447 km, otra ruta posible sería Cali -> Medellín -> Bogotá, pero esta no se muestra, pues la distancia sería 435+417=852. Lo mismo aplica para su otro deseo de viaje que es de Medellín a Bogotá, entonces se obtiene que la ruta más corta es Medellín -> Bogotá, pues la distancia es 417, otra ruta posible sería Medellín -> Cali -> Bogotá, pero esta no se muestra, pues la distancia es 435+447=882.

**Nota**: Para los dos puntos anteriores, en caso de empate, usted decide cómo proceder.

1. Dado el código de dos usuarios y un medio de transporte (bus o avión), si estos dos usuarios quisieran hacer un viaje juntos (usando siempre un mismo medio de transporte, es decir, hacen todo el viaje en bus o en avión) y cada usuario tiene su lista de viajes deseados, verifique si hay deseos en común entre los dos usuarios y para cuáles deseos les alcanza el dinero (en este caso, la suma del dinero disponible de cada uno) y teniendo en cuenta que deben comprar dos pasajes según el medio de transporte indicado.

**Ejemplo:** Queremos ver si Hayley Williams y Lauren Mayberry[[2]](#footnote-2) tienen viajes en común, entonces ingresamos sus respectivos códigos, y obtenemos que ambas quieren ir de Cali a Bogotá en avión, ellas tienen 500 y 600 dólares respectivamente, sumando un total de 1100 dólares; el viaje en avión cuesta 70 dólares un boleto, por lo que tendrían que pagar 140 dólares en total si quieren ir las dos y; por lo tanto, sí les alcanza.

Ahora veamos a Dua Lipa y Carmen Electra, ellas tienen un deseo en común, las dos quieren viajar de Villavicencio a Medellín por avión, como el costo de este viaje en avión es de 25+52=77 (pues tendrían que ir de Villavicencio a Bogotá y luego de Bogotá a Medellín), tendrían que pagar 154 dólares en total, pero la suma de sus dineros disponibles es de apenas 60 dólares; por lo tanto, no les alcanza para hacer el viaje en avión.

Resuelva todo lo anterior conectándose desde Python a las correspondientes bases de datos de Neo4j y de MongoDB (puede usar Visual Studio o conectarse mediante Google Colab…) No se califica la “belleza” de la impresión, lo importante es que los resultados se entiendan y sean correctos. Usted decide si se conecta a las bases de datos localmente o ubicadas “en la nube”. No hay restricciones en este aspecto.

**Notas adicionales:**

• Puede usar todas funcionalidades de Neo4j y de MongoDB que desee.

• Para entregar por email a [*fjmoreno@unal.edu.co*](mailto:fjmoreno@unal.edu.co)con copia al monitor **Alejandro Orozco Ochoa** <aorozcooc@unal.edu.co>, el lunes 7 de julio hasta las 5 pm.

• No se reciben trabajos en hora posterior. No se reciben versiones “mejoradas”.

• Grupos máximo de tres personas, mínimo de dos.

• Los trabajos deben ser independientes entre los grupos. Trabajos copiados así sea en un SOLO punto se califican con 0 (cero) en su totalidad para todos los integrantes. El trabajo debe ser desarrollado por los integrantes del grupo no por personas ajenas a él.

• Cualquier duda consultarla personalmente o por email con el profesor.

• El monitor les puede ayudar con aspectos técnicos pero su función no es hacerles la práctica **ni está autorizado para cambiar las condiciones del trabajo**.

• Si hay errores en el enunciado por favor informarme lo más pronto posible para corregirlos.

Atentamente,

Francisco Moreno, **junio 20, 2025**

**Nota:** Enunciado elaborado por el monitor Alejandro Orozco con base en una idea discutida conmigo.

1. ¡Sello de alta calidad del enunciado! [↑](#footnote-ref-1)
2. Estas dos cantantes, Hayley Williams de Paramore y Lauren Mayberry de Chvrches colaboraron juntas no precisamente en un viaje de Cali a Bogotá sino en la canción [Bury It](https://www.youtube.com/watch?v=B9BLMNn0PrQ), como parte de la edición extendida del álbum “Every Open Eye” de Chvrches. 🡪 Nota elaborada por el monitor. [↑](#footnote-ref-2)