

## RETO 4: Construcción y Análisis de tiempos de servicios de domicilios

---

### Objetivo

En este reto se pretende poner en práctica los conceptos aprendidos en clase acerca de las estructuras de datos que implementan el TAD Grafo y realizar búsquedas eficientes de información. Específicamente se desea:

Específicamente se pretende:

1. **Practicar** los conceptos sobre estructuras de grafos.
2. **Integrar** el uso de las estructuras lineales (tablas de hash, Lista, Mapas, pilas y colas) a los grafos.
3. **Practicar** los algoritmos de ordenamientos y búsquedas lineales.
4. **Aprender** a cargar y procesar en memoria datos en formato CSV.
5. **Utilizar** adecuadamente el administrador de versiones GIT y GitHub.
6. **Aprender** a trabajar en equipo.

### Fecha Límite de Entrega

El reto podrá entregarse hasta el 27 de mayo de 2025, 11:59 p.m. Ver las condiciones de entrega en la sección Entrega.

### Contexto

Se tiene un conjunto de datos que describe las entregas de pedidos a domicilio. Utiliza la edad del repartidor, su calificación previa el punto del restaurante y el punto de entrega. Cuenta con el tiempo de cada entrega

### Carga de Datos

Los datos del reto fueron elaborados por el equipo de Estructuras de Datos y Algoritmos y están preparados y disponibles para los estudiantes en el aula **unificada** en Bloque Neón (BrightSpace).

Se cuenta con un archivo donde se presenta toda la información de los domicilios. En el archivo deliverytime.csv se cuenta con todas las variables necesarias en el análisis. En la Tabla 1 puede verse la descripción de los campos del archivo deliverytime.csv.

Nombre de la columna	Descripción
ID	Número único que identifica el pedido
Delivery_person_ID	Número único que identifica al domiciliario
Delivery_person_Age	Edad del domiciliario
Delivery_person_Ratings	Calificación promedio del domiciliario
Restaurant_latitude	Latitud geográfica del restaurante
Restaurant_longitude	Longitud geográfica del restaurante
Delivery_location_latitude	Latitud geográfica del punto destino del pedido
Delivery_location_longitude	Longitud geográfica del punto destino del pedido
Type_of_order	Tipo de pedido
Type_of_vehicle	Tipo de vehículo
Time_taken	Tiempo en minutos del pedido

**Tabla 1.** Descripción de los campos del archivo CSV para los domicilios.

## Trabajo Propuesto

### Parte 1: Configuración Repositorio

Complete los siguientes pasos para configurar su repositorio de trabajo:

- p.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/ISIS1225-Reto-Template>
- p.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.
- p.3.** Edite el **README** del repositorio e incluya los nombres completos, correo Uniandes y códigos de los miembros del equipo de trabajo.
- p.4.** Realice el procedimiento según lo aprendido en clase para clonar el repositorio en su máquina local y sincronizarlo con su repositorio en GitHub.
- p.5.** Descargue los datos desde la sección unificada del curso y cópielos en la carpeta **data** del repositorio local.

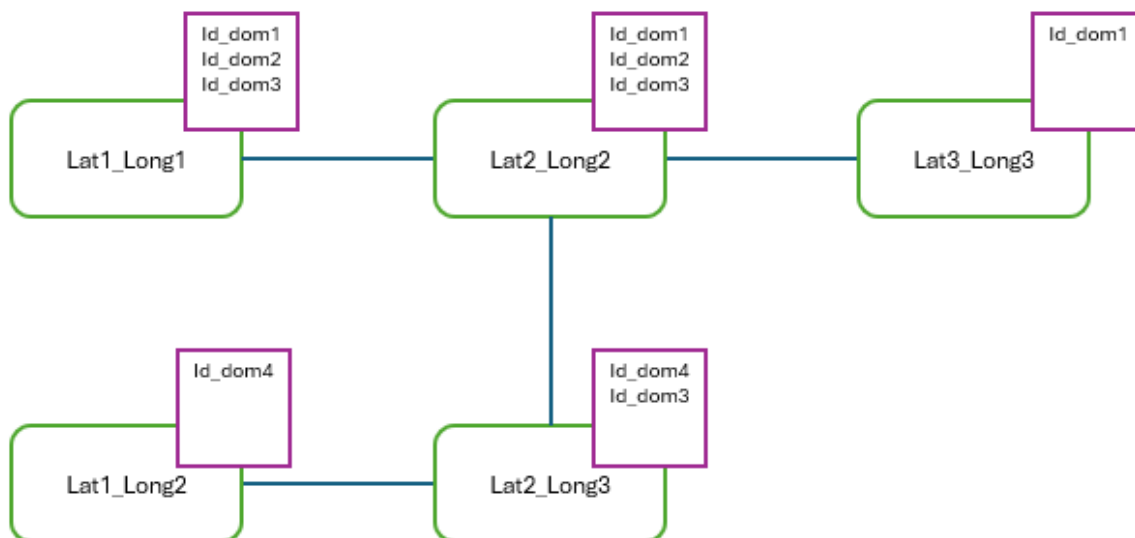
### Parte 2: Carga de Datos

En la sección de **Reto 4**, de la sección **unificada** en Bloque Neón, encontrarán los datos oficiales del reto. El ZIP contiene varios archivos con los sufijos **\_10**, **\_20**, **\_30**, etc. Estos son archivos con diferente número de registros. (ej.: el archivo **deliverytime\_10.csv** contiene 15.200 registros aproximadamente y el archivo **deliverytime\_large.csv** contiene la totalidad de los datos con 45.500 registros aproximadamente). Esto facilita la implementación y pruebas en computadores con memoria RAM y procesadores reducidos.

A continuación, se detalla el cómo se representará el grafo de estudio para así poder realizar la carga y desarrollar los requerimientos propuestos en este reto:

1. El Identificador de un nodo se construye a partir de su ubicación geográfica: Un String de la forma "<Latitud>\_<Longitud>" donde la latitud y la longitud tienen 4 dígitos decimales. Completar con 0s para tener cuatro decimales. Por ejemplo "22.7446\_ 75.8943" es el identificador de un nodo del grafo.
2. Cada domicilio solicitado tiene dos ubicaciones geográficas, el origen y el destino, quiere decir que cada solicitud de domicilio define dos posibles nodos en el grafo. Tenga en cuenta que No puede haber identificadores repetidos de nodos:
  - 2.1 Si la ubicación geográfica del origen de dos o más domicilios es la misma se consideran el mismo nodo.
  - 2.2 Si la ubicación geográfica del destino de dos o más domicilios es la misma se consideran el mismo nodo.
3. Entre los nodos de las ubicaciones de origen y destino de un domicilio se define un arco No dirigido. El peso del arco es el tiempo del domicilio respectivo. Tenga en cuenta que No se pueden crear arcos paralelos entre nodos (es decir, No puede haber más de un arco que conecte dos nodos). Si llegara a haber más de un domicilio entre un mismo par de ubicaciones origen y destino, el tiempo del único arco que conecta sus nodos es el tiempo promedio de los respectivos domicilios.
 

Adicionalmente por cada domicilio se debe crear un arco entre el nodo destino del domicilio actual y el destino del domicilio anterior atendido por el mismo domiciliario. El tiempo de este arco será el promedio del tiempo entre el domicilio actual y el último domicilio atendido por el mismo domiciliario.
4. Cada nodo identificado por su ubicación geográfica tendrá un listado de IDs de los domiciliarios que aparecen en domicilios que tienen como origen o destino esa ubicación geográfica.
5. La representación del grafo se ilustra así:



*Ilustración 1: Representación del Grafo No Dirigido a partir de las ubicaciones geográficas de los domicilios.*

Para responder los requerimientos deberán cargar la información de los archivos entregados; recuerde que solo se permite leer una vez la información de cada archivo y que las pruebas finales sobre sus algoritmos serán sobre los archivos \*\_large.csv.

Al completar la carga de datos de las relaciones y los usuarios se debe reportar los siguiente:

- Número total de domicilios procesados
- Número total de domiciliarios identificados
- Número total de nodos en el grafo creado
- Número de arcos en el grafo creado. Por ser un grafo No dirigido, cada arco aunque es bidireccional solo se debe contar una única vez.
- Número de restaurantes identificados por su ubicación geográfica (si la ubicación geográfica del origen de dos domicilios es la misma se consideran el mismo restaurante). Se tienen en cuenta solo los orígenes de los domicilios.
- Número de ubicaciones donde han llegado los domiciliarios (si la ubicación geográfica del destino de dos domicilios es la misma se consideran la misma ubicación atendida). Se tienen en cuenta solo los destinos de los domicilios.
- Promedio de tiempo de entrega de todos los domicilios procesados

#### Recomendaciones:

- Se recomienda utilizar librerías por extensión de Python como tabulate<sup>1</sup> para imprimir adecuadamente los resultados como se muestran en los ejemplos.
- En caso de que no exista algún valor dentro de los campos solicitados, completarlos con el valor “Desconocido” o “Unknown”.

### Parte 3: Desarrollo de los Requerimientos

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

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

---

<sup>1</sup> librería Python tabulate, PyPi URL: <https://pypi.org/project/tabulate/>

## Básico

REQ 1: Identificar el camino entre dos ubicaciones geográficas (G)

REQ 2: Identificar el camino con menos puntos intermedios entre dos ubicaciones para un domiciliario (G)

## Intermedio

REQ 3: Identificar el domiciliario con mayor cantidad de pedidos para un punto geográfico en particular (I)

REQ 4: Identificar los domiciliarios en común entre los puntos geográficos A y B (I)

REQ 5: Identificar el domiciliario que recorre mayor cantidad de distancia en N cambios de ubicación geográficas a partir de un punto inicial (I)

## Avanzado

REQ 6: Encontrar los caminos de costo mínimo en tiempo a partir de un punto geográfico específico (G)

REQ 7: Establecer una subred (árbol de recubrimiento de costo mínimo) de un domiciliario particular desde un punto inicial (G)

REQ 8: Graficar en un mapa el recorrido de un domiciliario en particular (G)

**NOTA:** Si el equipo este conformado solamente por **dos (2)** integrantes deberán resolver solamente **cinco (5) requerimientos obligatorios**: **uno (1)** de los requerimientos **básicos**, **dos (2)** de los requerimientos **intermedios**, y los **dos (2)** primeros requerimientos **avanzados**. Opcionalmente, el último requerimiento **avanzado (Bono)**.

## Requerimiento No. 1 (Grupal): Identificar un camino simple entre dos ubicaciones geográficas

Dados dos ids de ubicaciones A y B se requiere saber si existe un camino simple para conectarlos.

Los parámetros de entrada de este requisito son:

- Punto geográfico A (id del punto) origen desde donde se iniciará búsqueda.
- Punto geográfico B (id del punto) destino de la búsqueda.

La información que se espera como resultado debe incluir lo siguiente:

- Tiempo de ejecución del requerimiento.
- Cantidad de puntos geográficos en el camino que conecta a las dos ubicaciones de parámetro.
- Detalles del camino, en particular:
  - Id de los domiciliarios que compone el camino (sin repetir)
  - Secuencia de las ubicaciones que definen el camino simple desde la ubicación de origen hasta la ubicación destino.
  - Listado de restaurantes encontrados

**NOTA:** es importante aclarar que las soluciones para el requerimiento 1 y 2 deben implementarse con algoritmos distintos para que sea válida la solución.

## Requerimiento No. 2 (Grupal): Identificar el camino simple con menos puntos intermedios entre 2 ubicaciones para un domiciliario

Dadas dos ubicaciones A y B y un domiciliario se requiere obtener el camino de menor número de ubicaciones para conectarlos.

Los parámetros de entrada de este requisito son:

- Punto geográfico A (id del punto) origen desde donde se iniciará búsqueda.
- Punto geográfico B (id del punto) destino de la búsqueda.
- Domiciliario X (id del domiciliario) de la búsqueda

La información que se espera como resultado debe incluir lo siguiente:

- Tiempo de ejecución del requerimiento.
- Cantidad de puntos geográficos en el camino que conecta a las dos ubicaciones de parámetro.
- Detalles del camino, en particular:
  - Id de los domiciliarios que compone el camino (sin repetir)
  - Secuencia de las ubicaciones que definen el camino simple desde la ubicación de origen hasta la ubicación destino.
  - Listado de restaurantes encontrados

**NOTA:** es importante aclarar que las soluciones para el requerimiento 1 y 2 deben implementarse con algoritmos distintos para que sea válida la solución.

### Requerimiento No. 3 (Individual): Identificar el domiciliario con mayor cantidad de pedidos para un punto geográfico en particular.

Se quiere saber el domiciliario que más domicilios ha atendido en una ubicación geográfica específica.

Los parámetros de entrada de este requerimiento son:

- Punto geográfico A (id del punto) para realizar la consulta.

La respuesta esperada debe contener la siguiente información:

- El tiempo en que se demora el algoritmo en encontrar la solución.
- El domiciliario más popular según su cantidad pedidos en el punto geográfico de consulta.
- La cantidad de pedidos atendidos por el domiciliario con el punto geográfico de consulta.
- Tipo del vehículo que más repitió el domiciliario



## Requerimiento No. 4 (Individual): Identificar los domiciliarios en común entre los puntos geográficos A y B

Dado dos ubicaciones geográficas A y B, se requiere el listado de domiciliarios en común que están en el camino simple con menos puntos intermedios entre los puntos geográficos A y B.

Los parámetros de entrada de este Requisito son:

- Punto geográfico A (id del punto) origen desde donde se iniciará búsqueda.
- Punto geográfico B (id del punto) destino de la búsqueda.

La respuesta esperada debe contener la siguiente información:

- El tiempo en que se demora el algoritmo en encontrar la solución.
- Secuencia de las ubicaciones del camino simple con menos puntos intermedios entre A y B
- El listado de domiciliarios que se encuentra en común en este camino.

**Requerimiento No. 5 (Individual):** Identificar el domiciliario que recorre mayor cantidad de distancia en N cambios de ubicación geográficas a partir de un punto inicial.

Dado un punto geográfico específico determinar el domiciliario que recorre la mayor distancia geográfica en un número N de cambios de puntos geográficos.

Los parámetros de entrada de este requisito son:

- Punto geográfico A (id del punto) origen desde donde se iniciará búsqueda.
- El número N de cambios de ubicación a consultar.

La respuesta esperada debe contener la siguiente información:

- El tiempo que se demora el algoritmo en encontrar la solución.
- El domiciliario (detallando Id y distancia en Km) que recorrió la mayor distancia desde el punto de inicio dado.
- Secuencia de ubicaciones que definen el camino simple de mayor distancia que recorre el domiciliario desde la ubicación inicial hasta la N-esima ubicación.

**NOTA:** es importante utilizar el cálculo de la distancia Haversine con la Fórmula del semiverseno (Haversine formula) Ver sección del Bono

## Requerimiento No. 6 (Grupal): Identificar los caminos de costo mínimo en tiempo desde una ubicación geográfica específica

Se desea obtener los caminos de costo mínimo en tiempo desde una ubicación geográfica dada.

Los parámetros de entrada de este requerimiento son:

- Punto geográfico A (id del punto) origen desde donde se iniciará búsqueda.

La respuesta esperada debe contener la siguiente información:

- El tiempo que se demora el algoritmo en encontrar la solución.
- La cantidad de ubicaciones que definen los caminos de costo mínimo en tiempo (incluyendo la ubicación inicial).
- Identificadores de las ubicaciones “alcanzables” desde la ubicación dada que definen los caminos de costo mínimo (ordenados alfabéticamente)
- El camino de costo mínimo desde la ubicación inicial a la ubicación “alcanzable” que implica mayor tiempo. Para este camino mostrar la secuencia de ubicaciones y el tiempo total que se requiere para recorrerlo.

## Requerimiento No. 7 (Grupal): Establecer una subred (árbol de recubrimiento mínimo) de un domiciliario particular desde una ubicación inicial.

Se desea obtener una *sub-red* (Árbol de recubrimiento de Costo Mínimo en Tiempo) con una ubicación geográfica inicial y que conecta a las ubicaciones por donde pasa un domiciliario particular con costo mínimo.

Los parámetros de entrada de este requisito son:

- Punto geográfico A (id del punto) origen desde donde se iniciará búsqueda.
- Id del domiciliario a buscar.

La respuesta esperada debe contener la siguiente información:

- El tiempo que se demora el algoritmo en encontrar la solución.
- La cantidad de ubicaciones que definen la sub-red (incluyendo la ubicación inicial).
- Identificadores de las ubicaciones que definen la sub-red (ordenados alfabéticamente)
- Total en tiempo del Arbol de Recubrimiento de costo mínimo.

## Requerimiento No. 8 (Bono Grupal): Graficar en un mapa el recorrido de un domiciliario en particular

Dada una ubicación geográfica (expresada en latitud y longitud) y un radio (expresado en kilómetros), graficar en un mapa las ubicaciones geográficas junto con sus arcos existentes en el grafo que se encuentren en esa región circular que haya recorrido un domiciliario dado.

Los parámetros de entrada de este requisito son:

- Punto geográfico A (id del punto) centro de la región circular.
- Radio en kilómetros de la zona alrededor del punto geográfico A
- ID del domiciliario a consultar sus recorridos

Para calcular la distancia entre las coordenadas y determinar si están dentro del radio especificado, recomendamos utilizar la fórmula de **Haversine**<sup>2,3,4,5</sup>. Esta fórmula permite calcular la distancia entre dos puntos en la superficie de una esfera, considerando la curvatura de la Tierra, y es ideal para trabajar con coordenadas geográficas.

Para completar este requisito, recomendamos utilizar la librería de Python llamada Folium para visualizar los datos en un mapa interactivo. Folium se puede instalar en su ambiente con el siguiente 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

---

<sup>2</sup> Haversine formula, URL: [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula)

<sup>3</sup> Calculate distance, bearing and more between Latitude/Longitude points, URL: <https://www.movable-type.co.uk/scripts/latlong.html>

<sup>4</sup> 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>

<sup>5</sup> Haversine 2.5.1, URL: <https://pypi.org/project/haversine/>

Dentro del proyecto debe incluir, en la carpeta **Docs**, un documento en formato **PDF** donde se evidencie el análisis de complejidad y las pruebas de tiempos de ejecución 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.
- Análisis de complejidad de los requerimientos en Notación O.
- Pruebas de tiempos de ejecución para cada uno de los requerimientos. En estas pruebas se deben incluir:
  - Las tablas de tiempos de ejecución registrados.
  - Las gráficas comparativas de los experimentos.
  - Un análisis de resultados comparándolo los resultados obtenidos con el análisis de complejidad realizado.

#### Recomendaciones:

- Ejecute las pruebas de los requerimientos siempre con los mismos parámetros de entrada y con los archivos más grandes que pueda procesar su computador.

## 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 reto **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) de su sección.
- 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/Hora Límite de Entrega indicada al inicio de este enunciado.