

## Taller 8: Grafos no dirigidos

---

### Objetivos

- Aprender a construir un grafo en dos de sus posibles representaciones.
- Familiarizarse con el concepto de grafo.
- Aprender a navegar un grafo según su representación.

### Descripción General

Al inicio del siglo pasado se inauguró la primera sección del metro de New York, y a la fecha cuenta con 493 estaciones alrededor de esta importante ciudad y 100 rutas diferentes que operan según la zona, el horario y los tipos de emergencia que se puedan presentar. El gobernador de esta ciudad ha decidido realizar una aplicación para visualizar las rutas y los paraderos del sistema y lo ha contratado a usted para realizar el aplicativo, de esta manera mediante un extenuante trabajo se ha logrado recopilar la información de las estaciones (paraderos) y las rutas como se muestra a continuación:

- Estaciones: Nombre, latitud y longitud.
- Rutas: Nombres y estaciones por los que pasa
- Distancia entre estaciones

Esta información está dividida en dos archivos diferentes; el primero `stations.txt` contiene las estaciones (Nombre, latitud y longitud.) figura 1, y el segundo archivo `routes.txt` contiene las rutas y la distancia entre estaciones como se muestra en la figura 2.

#### **stations.txt**

<Número de estaciones>  
<Nombre de la estación>;<Latitud>;<Longitud>

Figura 1 Archivo `station.txt`

#### **routes.txt**

<Número de rutas>  
-----  
<Nombre de la ruta>  
<Cantidad de estaciones>  
<Nombre de la estación> <distancia>

Figura 2 Archivo `routes.txt`

Se dice que una estación (A) está conectada con otra (B) directamente si existe por lo menos una ruta que pase por las dos (A-B) estaciones. Dos estaciones pueden estar conectadas por varias rutas.

## Lo que usted debe hacer

La gobernación desea modelar el sistema como un grafo en donde los nodos son las estaciones y los arcos están representados por una estación (paradero) inicial (A), estación final (B), la distancia entre A y B, y la ruta que las conecta.

### Parte 1 – Preparación (en casa)

1. Descargue el proyecto Taller8.zip e impórtelo en Eclipse.
2. Lea la documentación de Grafo, Nodo y Arco que se encuentran en el paquete de estructuras.
3. Complete la clase GrafoNoDirigido.
  - 3.1. Declare la estructura que va a contener los nodos.
  - 3.2. ¿Para qué sirve el HashMap adj?, Cree y responda en el archivo README.txt en la carpeta data.
  - 3.3. Reemplace las estructuras de Java.util utilizadas para implementar la lista de adyacencia (adj) por las estructuras que creó anteriormente en el curso.

**Nota:** Si desea, puede implementar la lista adyacencia como lo prefiera, o usar la representación del grafo que quiera.

- 3.4. Implemente los métodos requeridos por la interface Grafo.

**Nota:** Si desea, cree las clases que considere necesarias para satisfacer la implementación.

### Parte 2 – Trabajo en clase

1. Complete la clase Estación (atributos, constructor y los métodos getters). Esta clase debe implementar la interface Nodo.

**Nota:** Recuerde que los atributos de la clase son: Nombre, Latitud y Longitud.

### En la clase **Administrador**:

2. Declare la estructura que va a contener la información del sistema (Grafo).
3. En el constructor inicialice la estructura como un GrafoNoDirigido.
4. Implemente el método cargarInformacion (utilice las constantes para cargar los archivos desde esas rutas).

*Recuerde revisar la sección donde se describen los datos.*

### La administración ha establecido que para la primera fase de la aplicación se requiere lo siguiente:

- R1: Conocer TODAS las rutas que pasan por un paradero dado.
  - R2: Conocer la distancia mínima y máxima entre dos paraderos adyacentes. Se debe mostrar los dos paraderos y la distancia.
5. Implemente:
    - 5.1. El método darRutasParaderos.
    - 5.2. El método distanciaMinimaParaderos y distanciaMaximaParaderos.
  6. Pruebe el funcionamiento de la aplicación y jugando con ella.

## Entrega

1. Verifique que su proyecto cumple con los requisitos de la entrega de talleres.
2. Cree y responda las preguntas en el archivo README.txt en la carpeta data.
3. Entregue su taller por medio de **BitBucket**. Recuerde, si su repositorio no tiene el taller o está vacío, su taller no será calificado por más de que lo haya desarrollado.