



-Programación II-

Trabajo practico obligatorio

Facultad de Ingeniería y Ciencias Exactas (FAIN)

Ingeniería en Informática

Materia: Programación II

Día-Turno: jueves- Mañana

Fecha del experimento: 26 de Junio de 2025

Profesor: Perez Nicolas Ignacio

Alumnos: Llousas Nicolas - Marolda Guido - Iriarte Facundo - Kim Pablo - Peralta Santiago

Año 2025

Índice

1.Enunciado	3
2. Problema real.....	3
3. Modelado de grafo	3
Imagen de red nueva.....	4
Pesos	5
4. Posibles problemáticas	5
1. Estaciones muy llenas:.....	5
2. Interrupciones en el servicio:	6
5.Conclusion	6

1.Enunciado

- 1) Buscar un problema real que se resuelva con alguno de los 3 algoritmos de Grafos Pesados que vimos en clase.
- 2) Crear o conseguir un programa que muestre este problema resuelto con alguno de estos algoritmos, bajo TDA.
- 3) En el repositorio del Proyecto crear un PDF breve (3/4 páginas) donde se explique el problema, la solución y las diferencias si las hay, con el algoritmo mostrado en clase. Además, si en la investigación apareció algún otro algoritmo comentarlo y buscar un testeo bajo TDA.

2. Problema real

En este trabajo práctico decidimos resolver el problema de las **conexiones** de las líneas de **subte de CABA**, es decir, que recorrido deberíamos hacer para llegar a nuestro destino con los **menores costos** (tiempo). Además de las líneas que ya existen, agregaremos 5 extra, basado en el plan PETERS ¹, así las líneas de subte para que se parecerán más a un grafo, lo que nos permitiría implementar el algoritmo de Dijkstra de una manera más clara.

3. Modelado de grafo

Cada nodo representa una estación de subte, es decir, tiene como dato un objeto "Estacion". Esta contiene el nombre de la estación y la línea a la que pertenece.

Las aristas representan la conexión entre las estaciones, las cuales se dividen en dos:

- Conexiones de la **misma línea**.
- Conexiones de **diferentes líneas**.

Las estaciones de la misma línea las conectamos con un método llamado "conectarLinea", el cual utiliza un for para conectar las estaciones. Pero para conectar estaciones de diferentes

¹ Originalmente creado en 1930, planeaba una red de subtes con 13 líneas.

líneas tuvimos que buscar los 2 nodos que queremos conectar, y luego de encontrarlos podemos conectarlos entre sí.

A continuación, le dejamos una imagen de la actual red de subtes, junto con las nuevas líneas con sus conexiones y recorridos.

Imagen de red nueva



- Línea A tiene el color celeste.
- Línea B tiene el color rojo.
- Línea C tiene el color azul.
- Línea D tiene el color verde oscuro.
- Línea E tiene el color violeta.
- Línea F tiene el color naranja.
- Línea G tiene el color marrón.
- Línea H tiene el color amarillo.
- Línea I tiene el color verde lima.
- Línea J tiene el color azul oscuro.
- Línea K tiene el color rosa.

Pesos

Para determinar el peso y el tiempo total en hacer el recorrido consideramos las siguientes cosas:

1. El tiempo que tarda en ir el subte **entre estaciones**
2. El tiempo que tarda **entre conexiones**. Estos tiempos de espera serán generados de forma aleatoria entre 1 a 5 minutos. Para dejarlo mas claro, haremos un ejemplo de cómo se calcularía el tiempo del recorrido.

Digamos que queremos ir de una línea de subte X en la estación 1, a una línea Y a la estación 2. Una vez que se recorrió todo el trayecto (utilizando dijkstra), se calcula el tiempo que habría consumido al realizar ese trayecto.

1. Tiempo de **espera inicial** en Línea X: 1-5 minutos (aleatorio)
2. Tiempo **de X1 a X2**: 120 segundos (pesoLineaX)
3. Tiempo de **conexión de X2 a Y1**: Tiempo adicional definido en la arista (tiempo que tarda una persona de ir de un andén a otro)
4. Nuevo **tiempo de espera** para la **Línea Y1**: 1-5 minutos (aleatorio)
5. Tiempo de **Y1 a Y2**: 135 segundos (pesoLineaY)

4. Posibles problemáticas

Mientras trabajábamos en este proyecto, nos dimos cuenta de que en la vida real siempre hay situaciones que afectan a los pasajeros del subte y que no consideramos en nuestro modelo actual. Algunas cosas importantes que nos gustaría considerar en una futura versión son:

1. Estaciones muy llenas:

Muchas veces hay estaciones que se llenan de gente, especialmente en horarios pico. Cuando eso pasa, la espera y el cambio de una línea a otra puede llevar mucho más tiempo del que estimamos. Aunque pusimos tiempos aleatorios de espera entre 1 y 5 minutos, estaría bueno poder incluir tiempos más realistas que tengan en cuenta cuánta gente hay en determinados horarios o días especiales. En este caso podríamos haber utilizado el **método A***, ya que,

utilizando las promesas, podríamos recomendar al usuario no utilizar/pasar por cierta conexión en horario pico.

2. Interrupciones en el servicio:

Sabemos que a veces el subte se interrumpe por mantenimiento, fallas técnicas o problemas inesperados. Actualmente, nuestro modelo no contempla esto, pero sería interesante incluir algún aviso o manera de recalcular la ruta cuando alguna estación o línea no esté funcionando.

5.Conclusion

Este trabajo nos ayudó a entender cómo usar lo que aprendimos en clase en un caso real. Elegimos el subte porque es algo que todos conocemos y usamos, y nos pareció una buena forma de aplicar el algoritmo de Dijkstra para buscar el camino más rápido entre dos estaciones.

Si bien hay cosas que podríamos mejorar (como tener en cuenta estaciones cerradas o con mucha gente), logramos simular una red bastante completa, sumando nuevas líneas y creando tiempos de espera realistas.