**TRABAJO FINAL ALGORITMICA Y PROGRAMACION II**

Alumno: Agustín Juárez

**Introducción**

Este trabajo se trata acerca de la simulación de un sistema de Líneas de subtes.

Dicho trabajo le brinda al usuario varios trayectos para llegar a su destino: el camino más rápido, el que tiene menos trasbordos entre estaciones de diferentes líneas y el que tiene menos congestión de gente.

El proyecto usa como ejemplo el sistema de subte de la ciudad de Buenos Aires, pero se construyó para que se pueda utilizar cualquier sistema de subte que exista en la actualidad. Las Líneas que la constituyen se denominarán A, B, C, D, E, H.

**Planteo del problema**

La problemática que se buscó solucionar con este programa era el de trazar la trayectoria que el usuario desee de una estación origen a una estación destino a través de una simulación de la red de Subtes de la ciudad de Buenos Aires, brindándole el mejor camino para llegar hasta ello.

**Análisis de las estructuras seleccionadas**

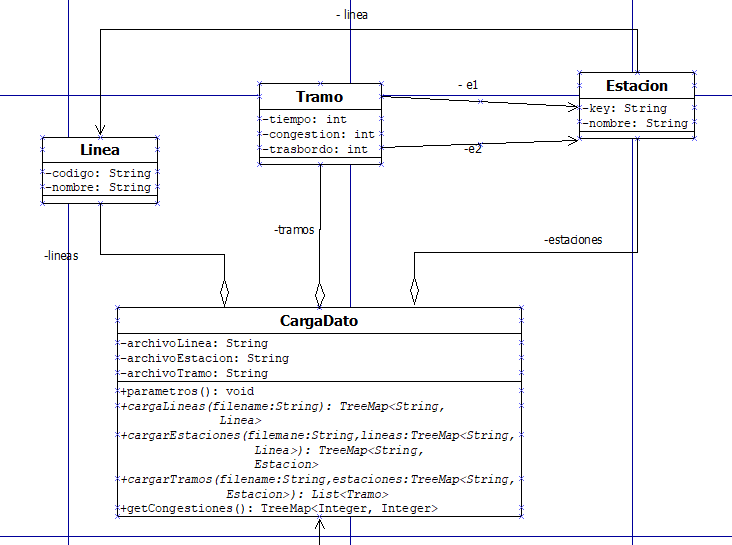
Para simular la red de líneas de los subtes se utiliza la estructura de un Grafo compuesto por sus vértices (estaciones), relacionados cada uno de ellos con arcos (tramo de una estación a otra), el cual contiene los valores con los cuales se trabaja en las soluciones de los problemas hablados anteriormente.

Como bien se dijo, se utiliza la estructura de un Grafo para la simulación. Los vértices de este están compuestos por las Estaciones y los pesos de los arcos son la variable que se requiera evaluar a la hora de elegir la opción que el usuario desea obtener.

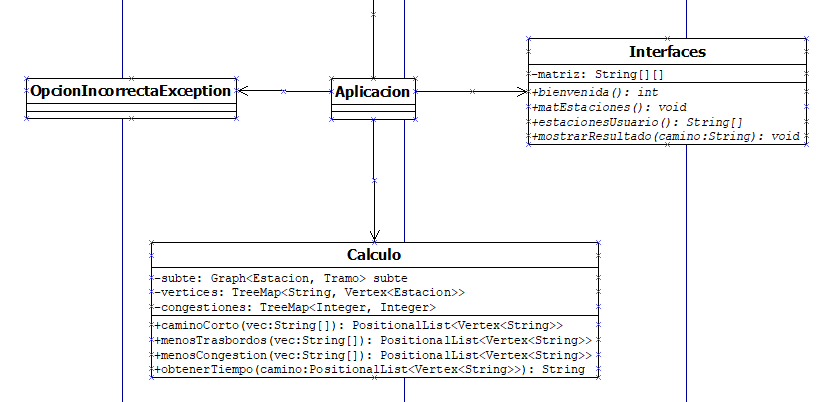
Por otro lado, se utiliza la estructura de los Mapas. El funcionamiento del Mapa es el de guardar datos, los cuales están asociadas a una clave. Se utiliza para guardar la información de las estaciones, las líneas y también para los valores de los coeficientes de la congestión.

También esta implementada la estructura de lista, ya que el método shortestPathList retorna como resultado una lista posicional de las estaciones que comprender el resultado de la opción elegida por el usuario.

**Diagrama de clases**



*Diagrama de clases UML Parte 1*



*Diagrama de clases UML Parte 2*

**Implementación de la solución**

Para la solución del problema se creó un programa que satisfaga la necesidad del usuario para elegir el camino que desee. Los datos que se utilizan comienzan guardados en archivos. El archivo de “estaciones.txt” contiene el nombre, la línea a la cual pertenece y el código correspondiente que lo identifica. El archivo líneas contiene el código de la línea y la letra asociada a cada una. El archivo tramo contiene los extremos (estaciones) de cada tramo que conforma el Grafo. El peso de cada tramo estará constituido dependiendo de la opción elegida por el usuario.

Cuando ya se cargan todos los datos de los archivos en los TreeMap, se pasan estos mapas a la clase que contiene la lógica del programa, Calculo. Esta clase contiene los métodos que construyen los grafos con sus correspondientes pesos en los arcos dependiendo de la opción que elija el usuario. Los métodos que contiene son caminoCorto, que calcula la trayectoria con menos tiempo de la estación origen a la estación destino. El método menosTrasbordos construye la trayectoria que contenga menos combinaciones de líneas entre la estación origen y la estación destino. Y, por último, la menosCongestion, donde calcula el camino con menos congestión de gente entre las estaciones nombradas anteriormente.

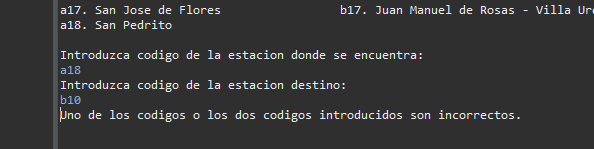
Todos los métodos mencionados anteriormente utilizan el algoritmo de Dijkstra, que, dado un grafo, calcula el camino entre vértices con la trayectoria entre los tramos que contengan el menor peso. Dicho grafo entregado al algoritmo de Dijkstra son construidos dentro de cada uno de los métodos anteriores.

**Manual de funcionamiento**

Al iniciar el programa se muestra una lista de las opciones que podrá elegir el usuario. Cada opción esta enumerada. Luego de elegir el número correspondiente a una opción, se pasa a mostrar una matriz con el código y nombre de todas las estaciones ubicada con la línea correspondiente. Se le pedirá al usuario que ingrese el código de la estación origen y luego el código de la estación destino. Seguido a esto se muestra la respuesta en una concatenación de los nombres de las estaciones con los respectivos tiempos de demora que se ven involucrados en la solución de la trayectoria que se debería hacer.

En caso de que el número de la opción o uno de los códigos de las estaciones sea invalido o no se encuentre, se lanzara un error al usuario en donde se le comunica lo dicho anteriormente.

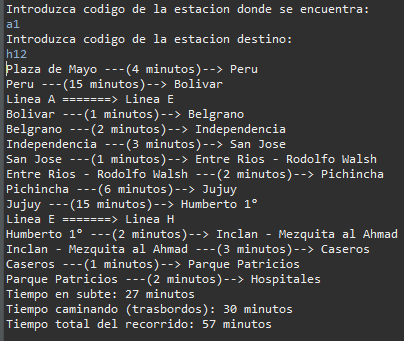
**Errores detectados**

1) Uno de los primeros errores que se nota a la hora de hacer el código, es cuando el usuario pone un código de estación y a la hora de calcular el trayecto, este no se encuentre entre los códigos de las estaciones (código invalido). En este caso se lanzaría un error diciéndole que el código ingresado es invalido. 

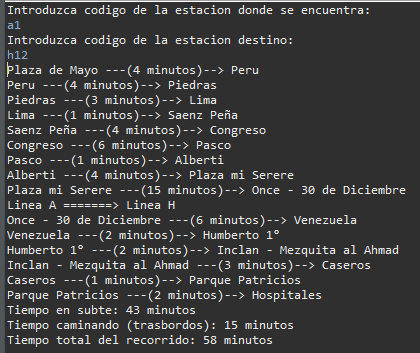
2) Si un archivo que se tiene que leer tiene un formato invalido para la aplicación, saltara un error.

3) Cuando se tiene que elegir la opción de trayectoria, si se elige un número que no corresponde a ninguna de las opciones que aparecen en pantalla, se mostrara un mensaje de error.

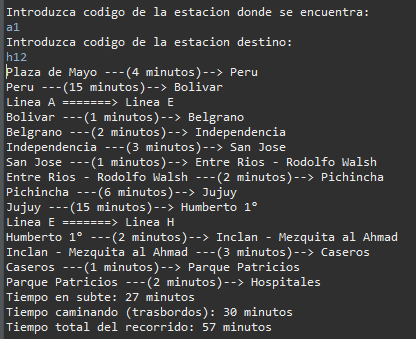
**Lotes de prueba**

****

Opción 1: camino más corto.



Opción 2: camino con menos trasbordos.



Opción 3: camino menos congestionado.

**Posibles mejoras y extensiones**

Para el futuro, me gustaría emplear la unidad grafica para que el usuario obtenga un mejor manejo de la aplicación a la hora de elegir sus estaciones y también, que tenga mejor experiencia en el uso de la misma, pudiendo ver en pantalla el mapa donde se encontrarían la ubicación de las estaciones y las líneas a los que corresponden.

**Conclusión**

Al principio no sabía por qué parte empezar, ya que se nos pedía varias cosas a implementar, pero a medida que iba avanzando se armaba el trabajo con sus correspondientes implementaciones.

Lo que más costo fue hacer uso de archivos, ya que era la primera vez que lo usaba. Otras complicaciones aparecieron en los métodos para construir los caminos con el uso de grafos y mapas.

A medida que avancé con el proyecto, hice entregas al profesor Gustavo Samec. Con las devoluciones que me entregaba, pude ir corrigiendo los errores que encontraba con indicaciones que me ofrecía.

Todavía en desarrollo hasta tener buenas devoluciones del trabajo y así contar las experiencias obtenidas.