Desafío 2 - Parcial 2

Informática II

Por:

Juan Felipe Higuita Perez

Semestre 2024-1

Universidad de Antioquia

2024

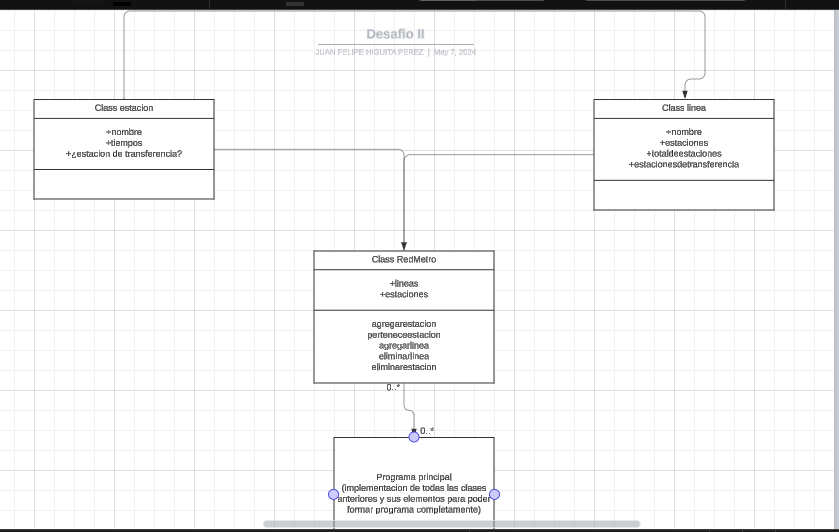
**Análisis del problema**

En este trabajo nos dan una pequeña introducción a un sistema de transporte muy conocido aqui en medellin que es el metro, diciéndonos que este desafío se trata de simular una red de este medio de transporte también explicando de qué estaba formada por líneas donde pueden ser clasificadas como tren o tranvía, además de que estas no presentan bifurcaciones ni bucles pero si son bidireccionales. Que dentro de esas líneas hay estaciones donde tienen un nombre, además de que hay que tener un tiempo de llegar a una estación a otra que se debe ser tomado en segundos y que algunas estaciones son especiales ya que estas son las intersecciones de dos líneas siendo nombrada estas estaciones por su nombre y luego el nombre de las líneas en la que están. Luego se explica que solo puede existir una línea en una red al igual que una estación no se puede repetir en una misma línea ni en las demas al menos que sea una estación de transferencia.

Ahora en la guia nos dan unos conceptos de ideas de unos subprogramas para implementar en nuestro código como la de crear y eliminar de una línea aunque la de eliminar no puede ser usada para las de transferencia, también de crear y eliminar líneas siempre y cuando no tengan una estación de transferencia, también averiguar cuantas estaciones hay en una red metro sin contar las estaciones de transferencias, saber cuántas estaciones hay en una línea y una en la que se puede calcular el tiempo que hay en llegar a una estaciona a otra, además de que en el código hay que tener en cuenta en usar obviamente lo visto en clase.

Entonces una posible solución sería organizar primero que nada los subprogramas en diferentes archivos para poder manejarlos en módulos, para que en el programa podamos invocarlos y poder usarlos para que formen el programa, un módulo podría ser para los subprogramas de las estaciones de como crearlas y eliminarlas ademas de la que se ve cuantas tiene cada linea y la red, otro que tenga que ver con las lineas y uno extra para el subprograma de tiempo entre estacion y estacion, con esto en el archivo principal hariamos un menu de usuario para que seleccione lo que desea hacer, tipo ponemos una opcion de observacion de estaciones, otra de administrar lineas y otra de administrar estaciones, en la de observacion de estaciones se agregaria otras 4 opciones en la cuales serian la de saber cuantas estaciones hay en una linea, otra para ver cuantas estaciones hay en la red, otra para saber el tiempo que hay entre estaciones y uno para devolverse, en las otras dos opciones del menu principal agregariamos 3 opciones que serian una agregar el elemento que se selecciono, otro para eliminarlo y uno para volver al menu principal, asi seria una primera idea que tendria de solucion para el programa.

***Diagrama de clases***



***Problemas durante el desarrollo del código***

Como en cualquier proyecto siempre se presentan algunas dificultades.

-Una de ellas fue la implementación de los tiempos a las estaciones, pensaba hacer que cuando fue la primera estación de una línea no se pusiera el tiempo y que si es la última no se pusiera el tiempo de la estación anterior, pero la verdad es que fue bastante difícil implementar esto al código y que se guardará correctamente, asi que preferí dejar eso a lo último para no complicar el código y no quedarme estancado en una misma parte.

-Otro problema que se me presentó fue el modo en el que se lee y guarda los datos, en un principio cuando recién se agregaba unas líneas y una estación si los detecta bien, pero cuando se salía del programa para volver a ingresar se desorganiza el archivo de texto, eso es un problema que se puede resolver a futuro.

-Ya algunos pequeños problemas que se presentaron fueron fallos en la lógica que no tardaron en resolverse.

***Algoritmos implementados***

-Funciones de menu: Estas son algunas funciones que hacen los menús del programa como es el menu de administrar líneas el cual es donde se selecciona las opciones de eliminar linea, agregarlas y ver cuales hay, tambien esta el menu de estaciones donde estan las opciones de agregar y eliminar estaciones, a la vez que tambien estara el menu principal donde estaran las opciones de administrar lineas, administrar estaciones, ver si una estacion pertenece a una linea y la simulación de tiempo.

-Clase estacion: en esta clase se recibiran dos atributos privados que son el nombre de la estacion junto si es de transferencia o no para seguidamente inicializarlos, luego como atributos publicos se intenta conseguir el nombre completo de la estación y verificar si es de transferencia o no.

-Clase linea: Aqui en esta clase se recibe como atributos en privado el nombre de la linea y un vector de punteros donde que apuntan a objetos de la clase estacion, seguidamente de forma publica se hace algunas funciones donde se permite agregar estaciones, si la linea tiene una estacion, que se eliminen de la linea, entre otras.

-Clase RedMetro: De forma privada se hace obtiene un vector de punteros que apuntan a objetos de la clase linea, luego de forma publica hay mas funciones como la de verificar si una linea ya existe, si una estacion existe en una linea, tambien la de agregar lineas, ver la cantidad de lineas, ver la cantidad de estaciones, tambien la funcion de eliminar lineas, la de agregar estaciones tanto de forma normal como entre dos estaciones ya existentes, de eliminar estaciones, verificar si una estacion pertenece, la funcion de la simulacion de tiempo de llegada, la funcion de guardar datos y cargarlos al archivo.

-Main: en el main tendriamos donde el usuario ingresa las variables para que el programa pueda leerlas y procesarlas para que este empiece a funcionar, como ingresar el nombre de una linea, el nombre de una estacion e ingresar el numero de las opciones del menu para poder ingresar.

***Evolucion de la solucion y consideraciones para la implementacion***

Primero se empezo con lo mas basico que fue el desarrollo de los menus, para que fuera de una forma organizada se implemento separar las opciones que tenian que ver con las lineas con las que tienen que ver con las estaciones como lo son las opciones de eliminar y agregar lineas, seguidamente de esto se implemento las funciones de cargar y guardar datos además de que cuando no se compruebe que el archivo seleccionado no existe entonces se crea uno automáticamente, esto fue lo siguiente ya que no podíamos empezar a desarrollar un programa sin tener algo de referencia antes como lo son los datos de la red metro.

Apartir de aquí se fue haciendo la clase estación para poder tener el nombre completo de estas junto a que si la estación era de transferencia según lo que ingresará el usuario en el menú gracias a que fue lo primero que hicimos, además de la creación de la clase línea para poder la clase anterior y hacer las funciones de agregar estación no sin antes verificar si ya existe en la linea que se desea agregar, ya con estos elementos fue posible crear la clase red metro siendo en esta que empezamos a agregar funciones como la de eliminar líneas como agregarlas al igual que la de las estaciones, después de esto empezo lo complicado ya que empece a hacer la funcion para que simule de calculo de tiempos que el usuario ingrese con cada estación para cuando el usuario quiera hacer la simulación, debo decir que me quede estancado durante unos pocos días intentando que el programa leyera correctamente los segundos que se contenían en el archivo o que tomara bien el tiempo, luego de varios intentos y errores decidí dejar esta función a medias para poder concentrarme en el resto.

Tambien pense en la posibilidad de que el usuario quisiera agregar una estación entre dos estaciones ya existentes en una linea, asi que hice una funcion para eso siendo que el usuario ingrese la estación anterior y la estación siguiente para que la nueva estación sea puesta entre estas dos, ya por ultimo hice las funciones de ver si una estacion pertenece a una linea ademas de ir modificando poco a poco el main para ajustarse a las nuevas clases y funciones que se fueron agregando