A) Análisis del problema y consideraciones para la alternativa de solución propuesta:

En este desafío se nos solicitó diseñar un sistema que simula una red de metro. La cual tiene líneas, y las líneas tienen estaciones. La misión principal de este sistema es permitir editar la red en la mayoría de calores y calcular tiempos de transporte en la red.

Al analizar el problema tuvimos que tener en cuenta una nueva herramienta que se puso a nuestra disposición, las clases. Viendo como se nos ha enseñado decidimos crear 3 clases, estación, línea y red.

En la clase red solo añadimos como variables el nombre de esta, un puntero que nos permitirá indexar las líneas por el hecho de que debemos usar arreglos dinámicos y los métodos, por cómo funciona este sistema la creación y eliminación de líneas está en los métodos de red. También añadimos a red todo lo que tiene que ver con calcular cuantas estaciones hay en toda la red y llevar el registro de cuantas líneas hay, lo cual nos sirve también para controlar el puntero.

Luego la clase línea, la cual programamos pensando en el hecho de que en una red hay muchas líneas, pero todas las líneas están en una red. En las líneas añadimos el nombre, el tipo de transporte y la cantidad de estaciones de la línea. Los métodos que añadimos acá siguen el mismo orden de red, el crear estación y eliminarla esta en esta clase, añadimos el método para calcular el viaje, ya que se nos dijo que solo estaría viajando en la misma línea, también, aunque no se añadió al principio se añadirá un método que revisa si hay alguna estación de transferencia.

Por ultimo la estación. Al principio se había pensado en una clase la cual fuese de estación de transferencia, porque queríamos implementar métodos y variables que decían a cuál estación estaban conectada para poder calcular el tiempo de viaje, pero después de que nos comentaran que los viajes solo serían en la misma línea decidimos estableces esto solo como un bool para evitar que una línea se borre si tiene una estación de transferencia.

B) Diagrama de clases de la solución planteada.

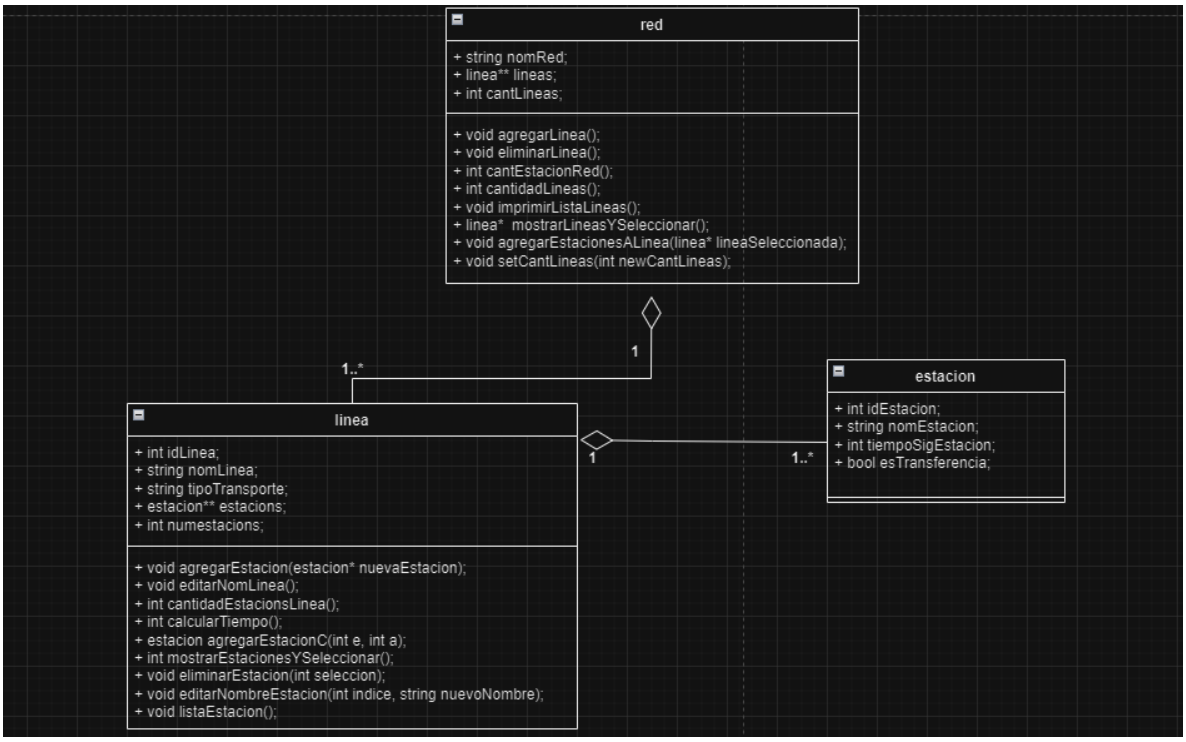
Clase red:
cantidadEstacionesRed(): Para establecer este método teníamos que tener en cuenta que no contara dos veces una estación de transferencia, san Antonio a y san Antonio b solo son una estación. Al contar las estaciones reviasaremos el bool de es transferencia y revisaremos en las otras redes donde esta la transferencia para revisar que solo se cuente como una estación.

eliminarLinea(): Es esta tuvimos que implementar algo extra, ya que no se puede borrar una linea con transferencia hicimos que revisara estación por estación y si tiene uno de los bool de es tranferencia positivo no se podrá eliminar

Clase estación:
calcularTiempo(): Para poder calcular el tiempo de viaje usamos los índices de las

estaciones, dependiendo de la orientación del movimiento usamos la variable de estaciónSig o estacionAnt.

eliminarEstacion(): En eliminar estación lo que tuvimos en cuenta de mantener fue el tiempo de viaje, al eliminar una estación intermedia, se heredara el tiempo de viaje de la estación eliminada a la estación de la derecha. Y viceversa en añadirEstacion() si es en medio de dos estaciones



D) Problemas de desarrollo que afrontó

Estructura de las clases: Al momento de estructurar las clases al principio teníamos una idea porque dábamos por hecho algunas cosas dentro del problema, pero por cómo iba desenvolviéndose el problema tuvimos que editar muchas cosas de esta estructura, sus métodos cuantos tenía y más cosas por el estilo. Así mismo, por el hecho de que no podíamos usar contenedores como se podía en la practica 4, toco repensar el cómo íbamos a acceder a la información almacenada en cada objeto.

Uso de punteros en las clases: Mientras que en el punto de vista gráfico y textual teníamos claro el funcionamiento de la red se nos dificulto al momento de codificar, y creemos que gran parte de esto es por los punteros, aunque después de entender la primera estructura, ya solo era ponerla de nuevo para que funcionara en otra clase.

Darle control al usuario sobre los objetos: Al intentar darle el acceso al usuario para que pudiese manejar los objetos de la red a voluntad nos encontramos con el problema de acceder a una variable exacta. Si lo hacíamos directamente desde el código sabíamos hacerlo, pero como podíamos hacer para que el usuario pudiera ingresar a esto. Al principio

nos estábamos complicando de más pero después nos dimos cuenta que solo era implementar un menú

E) Evaluación de la solución y consideraciones

La solución que dimos al problema nos parece adecuada, aunque no sabemos si es la más eficaz, al momento de usar el código recomendamos tener en consideración lo siguiente:

- Después de crear una estación de transferencia no se puede borrar una linea, tener muy en cuenta cuales son las estaciones de transferencia para poder eliminar la linea de manera correcta

- Poner atención a los datos ingresados, hay algunos datos que no se pueden revertir ni borrar, lo cual haría que deba reiniciar el proceso con la red

- Los viajes solo son en la misma línea, a pesar de que hay estaciones de transbordo los viajes solo se miden en la misma linea