08/05/2024

Universidad de Antioquia

Informática 2 – Informe de “desafío 2”

William Andres Ortiz – 1004190356

Para comenzar con el desarrollo del desafío, tuve en cuenta los requisitos puntuales del documento, por lo que mi solución al principio se basó en el uso de la STL para generar arreglos dinámicos sin la necesidad de preocuparme por el control de la memoria. Después, escuché bien, y entendí que su manejo estaba prohibido, por lo que proseguí a implementar mi solución usando punteros.

La solución tenía que realizarse con POO, analizando las relaciones de clases, me di cuenta que sería útil, por ejemplo, para representar la contención de las líneas dentro de la red, o la contención de las estaciones dentro de una línea, un objeto de clase Metro o un objeto de clase línea tuvieran cada uno un arreglo con sus líneas y estaciones respectivas. Partiendo de este punto, si utilizaba una herencia directa, tomando a metro como clase padre, línea como clase hija de metro y estación como clase hija de línea, no podría generar los mencionados arreglos, ya que la herencia en C++ no es bidireccional.

Creé cada clase por separada, y la incluí dentro de la otra, así, metro puede usar métodos de línea y línea puede usar métodos de estación.

Un aspecto para analizar dentro de la concepción del desafío en cuanto a sus condiciones, es el hecho de eliminar una línea. Solo se puede eliminar una línea si esta no tiene estaciones de transferencia, por lo tanto, solo puede eliminarse si la línea no está conectada a ninguna otra, pero ya existe la condición de que una línea debe estar conectada a la red. Esto, personalmente lo entendí como una contradicción, por lo que decidí omitir esta función del programa.

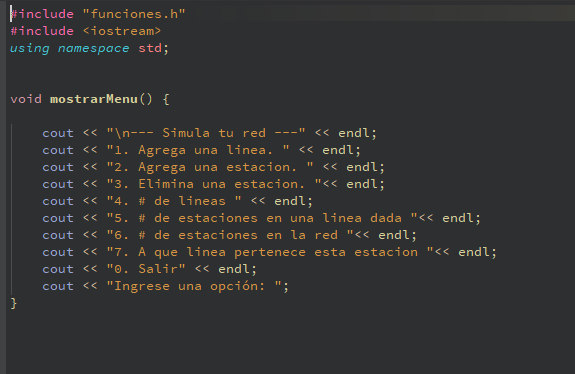
Dentro de las funcionalidades, la que mas se me dificultó en implementar fue el hecho de agregar una línea, para esto debía conectarla a la red y en ese momento mi programa comenzó a presentar errores por fragmentación de segmento. Ahí me di cuenta que no estaba validando si las direcciones de memoria creadas apuntaban a un objeto de clase línea o estación, o a cualquier otra cosa, por lo que decidí primero que todo inicializar los arreglos con punteros nulos, para después validar si se apuntaba a un “nullptr” o a una variable de tipo línea o estación, de las cuales si puedo extraer atributos o utilizar sus métodos.

Texto

Descripción generada automáticamente

Particularmente me quedé sin tiempo para implementar la funcionalidad “Hora de llegada”, pero soy consciente de que podría hacerse nada mas con el uso de los atributos de tiempo que tienen las estaciones y algo de algebra sencilla, me gustaría poder mostrarla al momento de la sustentación.

Por último, para hacer la interacción con el usuario mas amigable, simplemente cree una función que lanza un menú con las opciones a elegir dentro de la simulación de nuestra red, que se muestra hasta que el usuario desee salir.

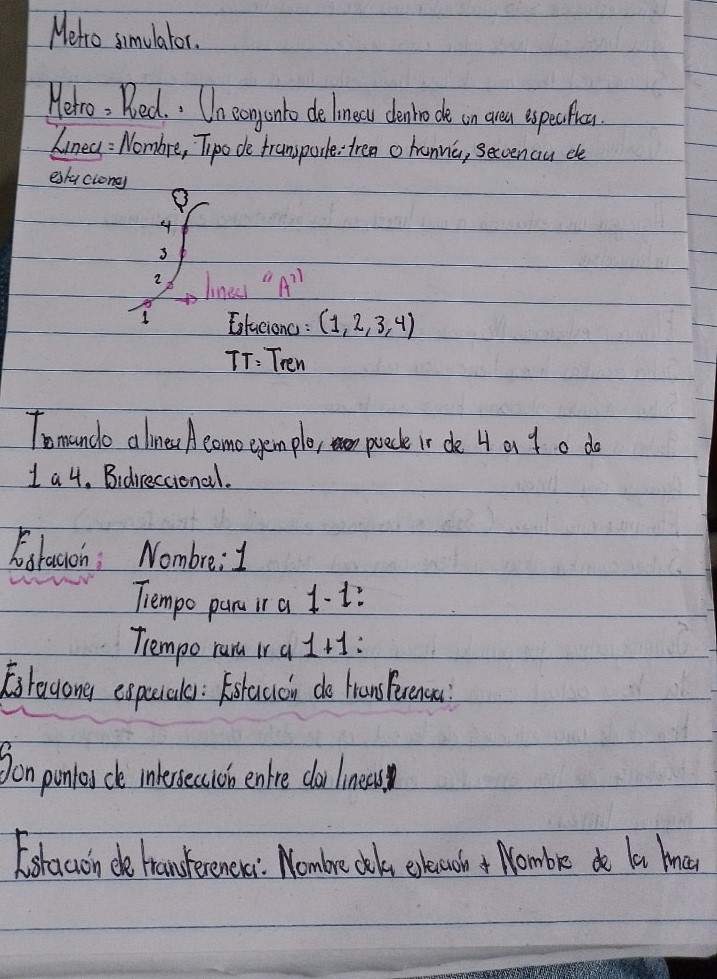


Texto

Descripción generada automáticamente

Algunas muestras del avance y el análisis hecho tras los días:

Texto, Carta

Descripción generada automáticamente Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza mediaImagen que contiene texto, documento

Descripción generada automáticamente

|  |
| --- |
| Metro |
| -Nl : int  -Cont : int  -Nombre : string  +Líneas\_ : Linea\*\* |
| +getNuml: int.  +addl(índice:int,nombre:string,  tipoT:string,numE:int):void.  +delLi(nombre:string):void.  +empty():bool.  +connL(n1:string,n2:string):void.  +validar(Nom:string):bool.  +addest(noml:string,nome:string,  Pos:int):void.  +delest(noml:string,nome:string):void  +numes(noml:string):int  +numEsRed():void  +belonges(nomes:string)void |

|  |
| --- |
| Linea |
| - Nombre : string  - Tipot : string  + NumE : int  + cont : int  +Estaciones : Estacion\*\* |
| -getNombre() : string  -getTipoT() : string  -getNumE() : int  -addEs(pos:int,nombre:string,  Trans:bool) : void  -delEs(nombre:string) : void |

0\* 0\*

0\*

|  |
| --- |
| Estacion |
| -nombre : string  -trans : bool  -tiempo1 : int  -tiempo2 : int |
| +getNombre():string  +setNombre(nombre:string):void  +setTrans(t:bool):void  +getTrans():bool |

0\*

DIAGRAMA DE CLASES.