INFORME DESAFIO 2

Autores

John Axel Ortega Rivera

Victor David Muñoz Ramirez

Profesores

Augusto salazar

Anibal

Universidad de Antioquia

Carrera de ingeniería electrónica

Medellín, Antioquia

24 marzo de 2024

Índice:

Problemática:

- **Breve descripción del problema que se va a resolver.**

Se requiere desarrollar un simulador de una red de Metro que permita modelar algunas características de su funcionamiento. Una red de Metro está compuesta por líneas, y cada línea tiene una secuencia de estaciones. Algunas estaciones pueden ser estaciones de transferencia, perteneciendo a múltiples líneas. El simulador debe permitir agregar/eliminar estaciones y líneas, así como realizar operaciones como calcular el tiempo de llegada de un tren entre dos estaciones de la misma línea. El diseño debe seguir los principios de la Programación Orientada a Objetos (POO) y utilizar estructuras de datos eficientes, como arreglos dinámicos, sin el uso de contenedores de la biblioteca estándar de C++.

- **Identificación de requisitos funcionales y no funcionales.**

Requisitos funcionales:

Funciones y métodos → Apartado en el diseño de la solución.

Entradas y salidas → Estaciones y líneas. Red de metro como salida

Excepciones y verificaciones → Estaciones y líneas.

Requisitos no funcionales:

Clases, arreglo dinámico → capacidad de trabajo: El programa debe ser capaz de gestionar eficientemente la memoria y evitar fugas de memoria. → **arquitectura apropiada:** El programa debe contar con una arquitectura acorde para la resolución del problema, mostrando así el resultado más eficiente y estructurado.

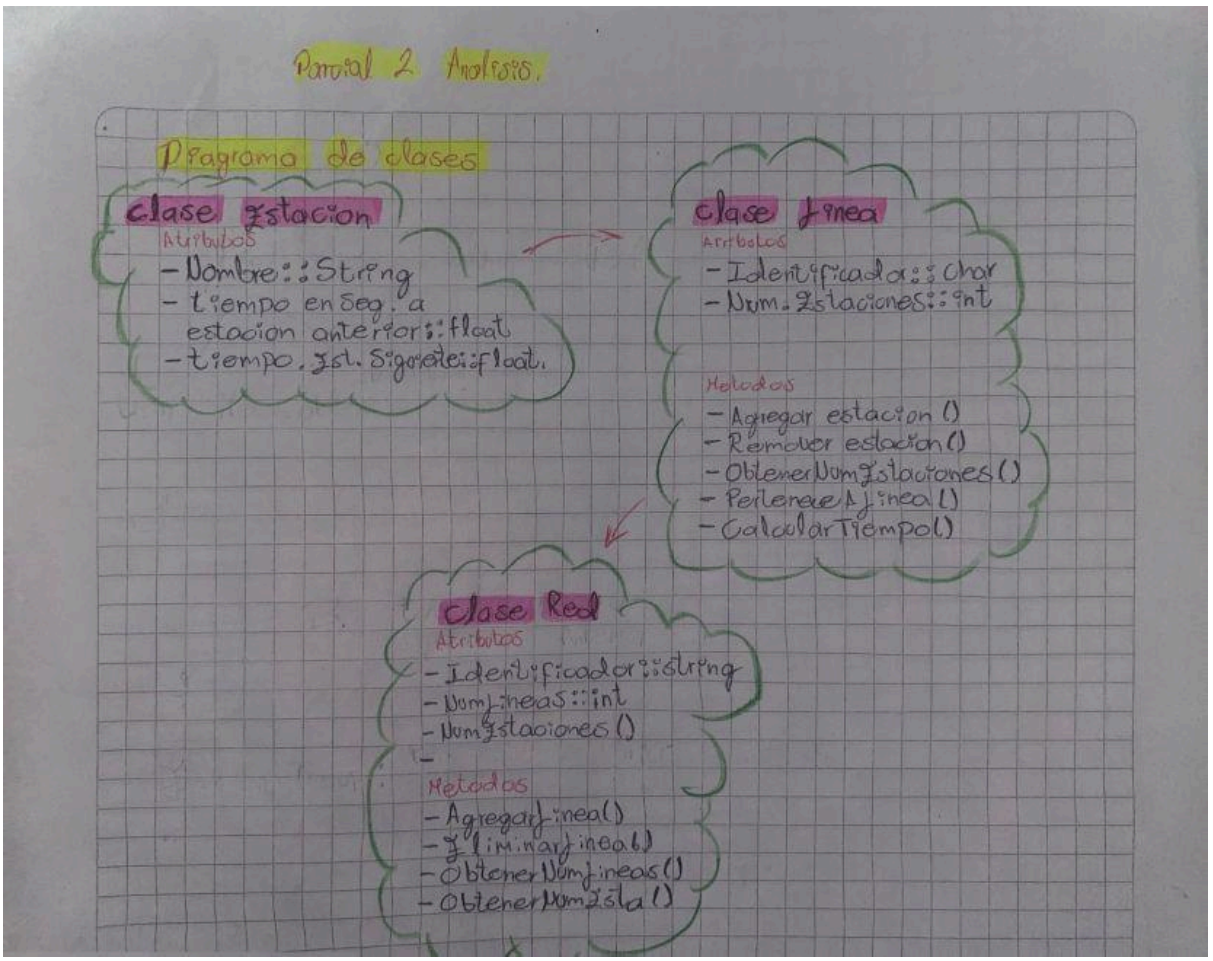
Análisis del problema:

- **Identificación de las entradas y salidas del programa.**

Entradas: Estaciones y líneas, y sus atributos, para las estaciones el nombres, los tiempos anteriores y siguientes; y las líneas su nombre.

Salidas: Red de metro.

Diagrama de clases:



Diseño de la solución:

- **Explicación de la arquitectura general del programa.**

Variables globales: nombrelinea, nombreEstacion.

Programa Principal: El programa principal consta del menú que se mostrará en el monitor repetidamente hasta que el usuario indique el momento de salida; el menú como tal funcionara con un do while y con switch, donde cada una de las opciones hará las llamadas necesarias para cumplir su objetivo, en base a esto las opciones del programa serían las siguientes:

A. Agregar una estación a una línea.

B. Eliminar una estación de una línea.

C. Saber cuantas lineas tiene la red Metro.

D. Saber cuántas estaciones tiene una línea dada.

E. Saber si una estación dada pertenece a una línea específica.

F. Agregar una línea a la red Metro.

G. Eliminar una línea de la red Metro.

H. Saber cuántas estaciones tiene la red Metro.

I. Calcular el tiempo de llegada de una estación a otra de la misma línea.

X. Salir

- **Descripción de las estructuras de datos utilizadas.**

Arreglos dinámicos→ Son estructuras de datos flexibles cuyo tamaño puede cambiar durante la ejecución del programa. A diferencia de los arreglos estáticos, cuyo tamaño se define en tiempo de compilación, los arreglos dinámicos se asignan en tiempo de ejecución utilizando punteros. Esto permite la gestión eficiente de la memoria, ya que se asigna solo la cantidad necesaria de memoria en el momento necesario. Los arreglos dinámicos son útiles cuando el tamaño del conjunto de datos no se conoce de antemano o puede cambiar durante la ejecución del programa.

Clases → Son estructuras que permiten agrupar datos y funciones relacionadas en un solo objeto. Utilizadas en programación orientada a objetos (POO), ofrecen encapsulación para controlar el acceso a los datos y abstracción para modelar entidades del mundo real. Los constructores y destructores son métodos especiales que se utilizan para inicializar y destruir objetos, respectivamente. Los modificadores de acceso, como public, private y protected, controlan la visibilidad de los miembros de la clase, asegurando una encapsulación efectiva.

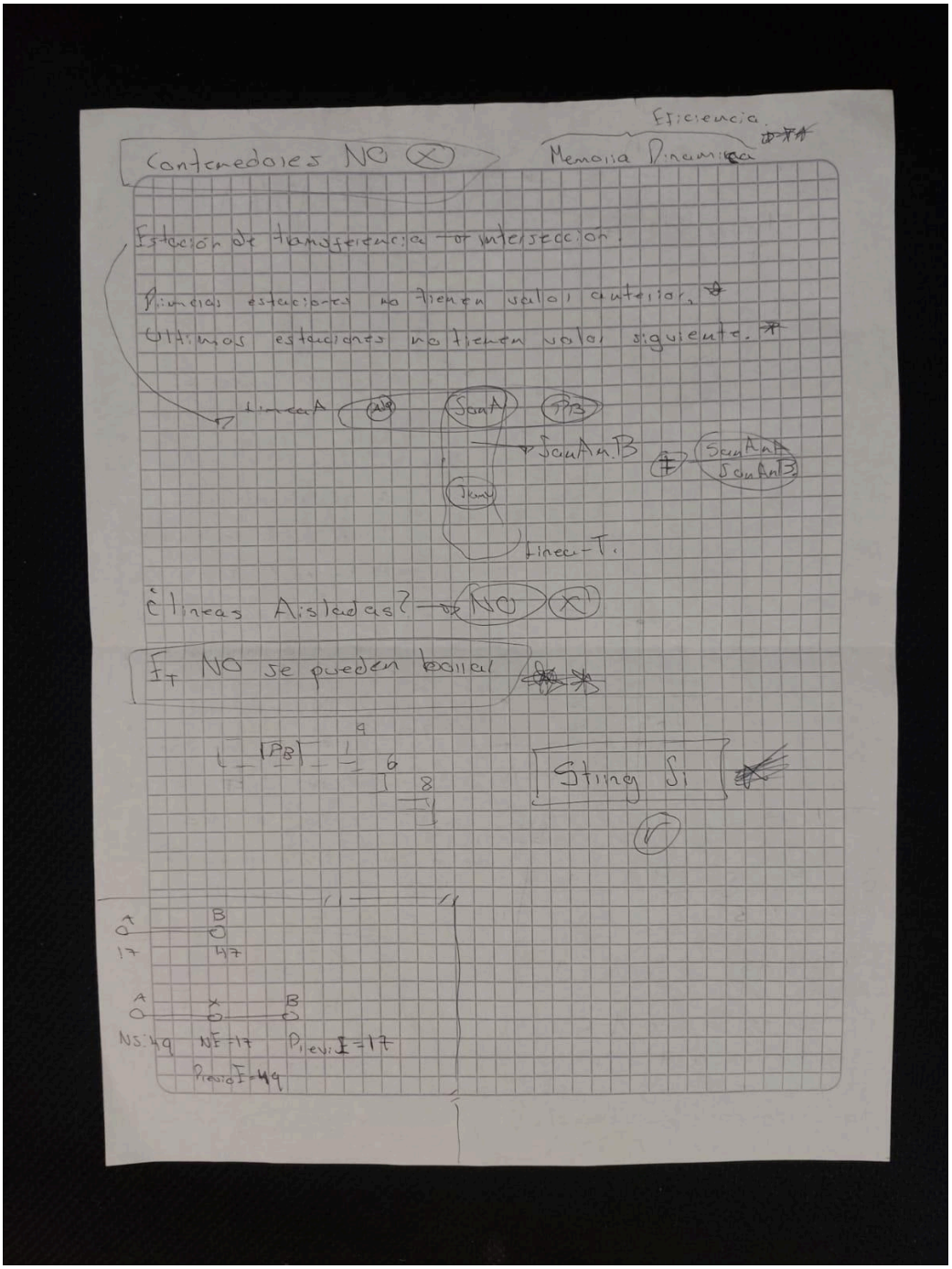
Algoritmos empleados para resolver el problema.

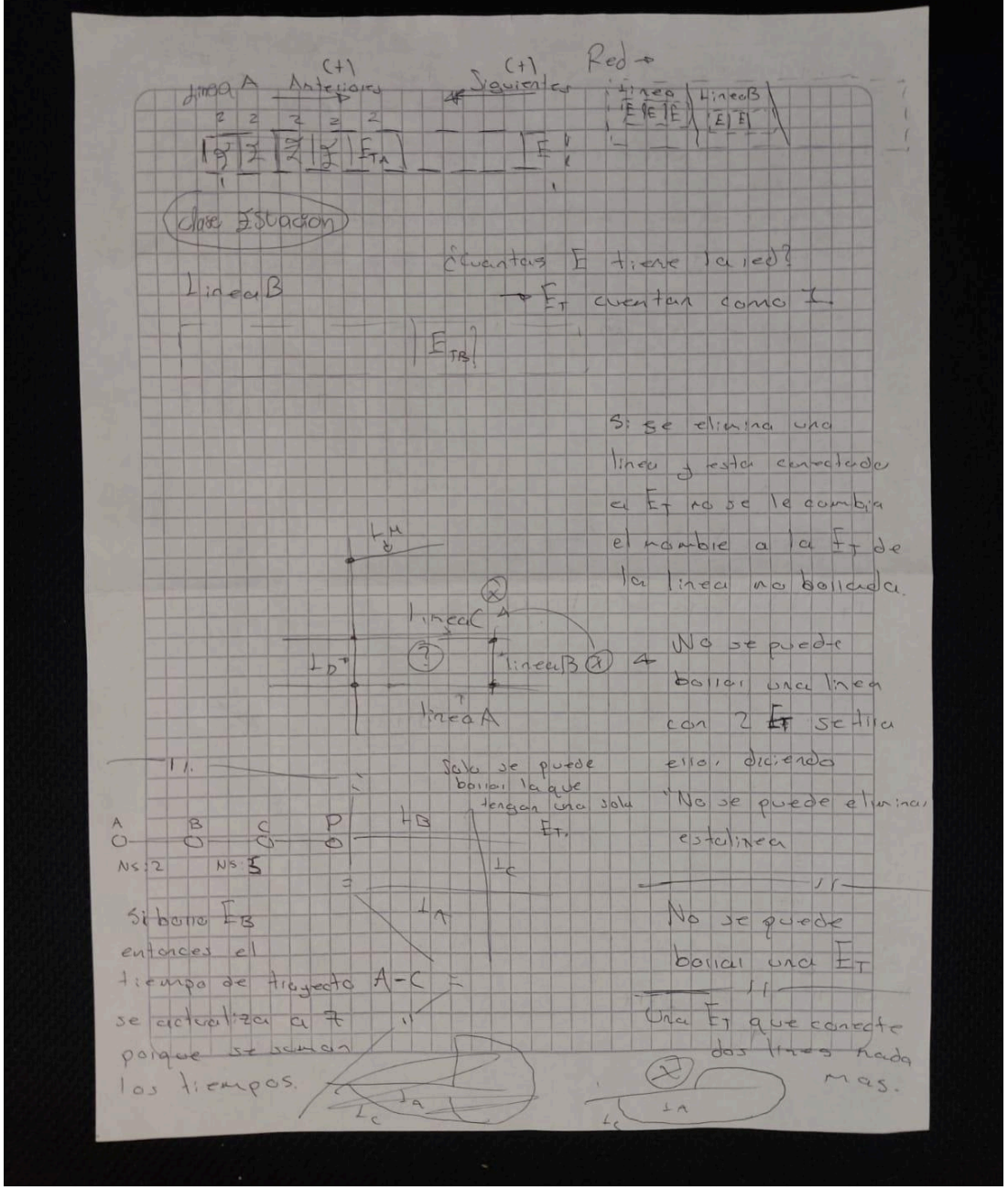
Clases, operaciones con punteros, operaciones aritméticas.

Anexos

En el siguiente link se muestran las notas/análisis/planteamiento del proyecto que se van realizando a lo largo del desarrollo del parcial.

https://udeaeduco-my.sharepoint.com/:o:/r/personal/axel_ortega_udea_edu_co/Documents/Blocs%20de%20notas/Desafio2?d=w4e4b50ff1cf34f8391d9dee39a96b379&csf=1&web=1&e=Bsj3xI




Línea A
(+1) Anteriores
(+1) Siguientes
Red
Línea B
clase ESTACION
¿Cuantas E tiene la red?
→ ET cuentan como 1.
Línea B
ETB
Si se elimina una linea y esta conectada a ET no se le cambia el nombre a la ET de la linea no bolleada.
LM
Línea C
Línea B
Línea A
No se puede bolar una linea con 2 ET se lia ello, diciendo "No se puede eliminar esta linea"
A
B
C
D
NS: 2
NS: 5
Solo se puede bolar la que tengan una sola ET.
Si bolla ETB entonces el tiempo de trayecto A-C = se actualiza a ∞ porque se anulan los tiempos.
No se puede bolar una ET
Una ET que conecte dos lineas nada mas.