

## Informe Desarrollo Del Parcial II

Nombre: -Alejandro Zapata Quintero

-Juana María Márquez Guzmán

proceso de análisis y diseño de la solución.

**Proceso de análisis:** Este informe detalla la creación y desarrollo de un simulador de red de Metro programado en C++. El objetivo principal de este proyecto es generar una herramienta interactiva que facilite a los usuarios el estudio y entendimiento de la operación de un sistema de transporte, como el Metro, en una ciudad imaginaria.

**Evaluación del Problema:** El reto propuesto implica la creación y desarrollo de un sistema que pueda simular la complicada red de Metro de una ciudad. Esto conlleva la generación de una estructura de datos que represente de manera adecuada las estaciones, las líneas y la red en su totalidad. Adicionalmente, el sistema debe permitir la adición de estaciones a las líneas, llevar a cabo simulaciones de tiempos de llegada entre estaciones y ofrecer otras funcionalidades cruciales para la administración eficaz del Metro.

**Solución diseñada hasta el momento:** Para solucionar el problema y realizar apropiadamente el diseño de la red de metro, se propuso la utilización de clases, las cuales tienen una respectiva pertenencia entre ellas. Por lo cual planteamos dos clases principales, "Estacion" y "Linea" las cuales comparten ciertos elementos necesarios para la adecuada construcción de la red. No se planteó una clase llamada "RedMetro" ya que las funcionalidades de esta al ser tan pocas pueden ser manejadas a partir de contadores propuestos desde el main u otras clases, un ejemplo de estas funcionalidades es "Saber cuántas líneas tiene una red Metro".

**Selección de Estructuras de Datos:** Se realizó una cuidadosa selección de las estructuras de datos más adecuadas para almacenar la información de las estaciones, líneas y la red del Metro. Se optó por utilizar arreglos y punteros normales debido a su eficiencia en el manejo de datos y su bajo consumo de recursos. Esta elección garantiza un rendimiento óptimo del simulador, incluso cuando se trabaja con grandes cantidades de información.

**Criterio de eficiencia:** se tendrá en cuenta la capacidad del software para simular el funcionamiento del sistema de manera rápida, precisa y con un consumo de recursos

mínimo, mediante la optimización de las operaciones de simulación, la gestión eficiente de datos y el uso adecuado de algoritmos y estructuras de datos.

**Función “Cálculo del Tiempo de Llegada”:** Esta función especializada se encarga de calcular el tiempo de llegada de un tren específico entre dos estaciones pertenecientes a la misma línea. Este subprograma emplea algoritmos eficientes para realizar los cálculos de manera precisa y rápida. Se tiene previsto que, utilizando los atributos “tiempoAnterior” y “tiempoSiguiente”, los cuales estarán definidos en la clase Estación, y tomando en cuenta los nombres de las dos estaciones entre las que se desea conocer el tiempo de tránsito, se podrá calcular el tiempo de llegada. Esto proporcionará al usuario información valiosa sobre los tiempos de viaje dentro del sistema de Metro.

- A. Agregar una estación a una línea, en los extremos o en posiciones intermedias.
- B. Eliminar una estación de una línea. No se pueden eliminar estaciones de transferencia.
- C. Saber cuántas líneas tiene una 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 (sólo puede eliminarse si no posee estaciones de transferencia).
- H. Saber cuántas estaciones tiene una red Metro (precaución con las estaciones de transferencia).

Por último se necesita un subprograma que permita hacer una sencilla simulación del funcionamiento de la red. Dicha simulación (“Cálculo del tiempo de llegada”), toma la hora actual como tiempo de salida de un tren desde una estación de la red, y se debe predecir el tiempo que tardaría ese tren específico en llegar a otra estación dentro de la misma línea. Este subprograma solo realiza esta tarea entre estaciones de la misma línea.

clase Línea

#### ATRIBUTOS:

- estaciones
- nombre de la línea
- Estacion de transferencia

CLASE LINEA estacion

#### Atributos

- contador estaciones ( 2 o mas )
- nombres de estaciones
- tiempos ( anterior , despues )

Estas son las clases tentativas que tenemos con sus respectivos atributos después de hacer un análisis en la lógica, sin embargo este planteamiento está sujeto a cambios.