**DESAFIO II**

**Informática II**

**Ethan Salomon Parra Reyes**

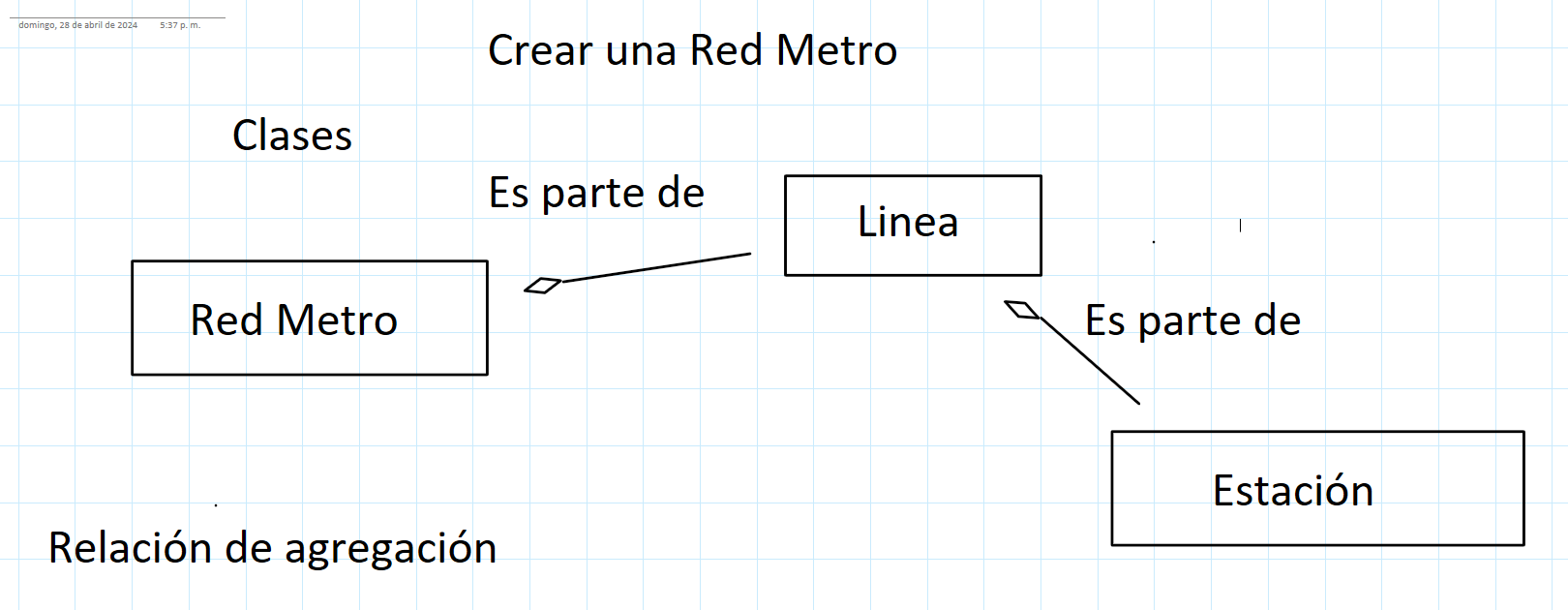
**Juan Felipe García Bonilla**

**Universidad de Antioquia**

**Medellín**

**2024**

**Análisis.**



* Se debe implementar un sistema para gestionar el proceso de guardado de la información en las líneas.
* Se debe desarrollar un método que permita el ingreso y retorno de las estaciones con sus respectivos tiempos siguientes o previos con respecto a las estaciones contiguas.
* Se buscara la forma de crear arreglos de tamaño variable, pero sin olvidarse de la eficiencia y el uso de memoria que conlleva este proceso.
* Es importante implementar mecanismos para manejar situaciones inesperadas.
* Hacer métodos para validar que no existan dos líneas ni dos estaciones con el mismo nombre en la red Metro, y de igual forma dos estaciones con el mismo nombre en la misma línea.
* Se deberá crear métodos para procesa la información que es pedida en el DesafioII.
* Se crearan métodos de eliminación de líneas y estaciones, siguiendo las reglas que permiten eliminar una de estas.

C**onsideraciones a tener en cuenta para el desarrollo del problema.**

* Una estación puede pertenecer a varias líneas (si es una estación de transferencia). Los nombres de las estaciones de transferencia se conforman concatenando el nombre de la estación con el nombre de la línea donde se encuentra.
* Una estación sólo puede estar una vez en una línea.
* Una línea sólo puede estar una vez en una red.
* No se pueden crear estaciones independientes a una línea.
* Si una red tiene más de una línea, estas líneas no pueden estar desconectadas.
* Una estación de transferencia puede pertenecer a multiples líneas.
* No se permite que las líneas estén conectadas de tal forma que se creen bucles.
* No podrán haber dos estaciones ni líneas con el mismo nombre.
* Las líneas del metro serán arreglos que contendrán las estaciones y el tiempo a las estaciones contiguas (LineA [re, 2, leA, 5, me, 8, te, 4, ce]).
* Se le preguntara al usuario cual es la estación de transferencia.
* Se duplicara el tamaño de los arreglos de acuerdo a las necesidades, esto con el fin de minimizar el uso de memoria, modificando el tamaño del arreglo cada vez que se ingresa o elimina una nueva estación.
* Cada vez que se cree una nueva estación se le indicara al usuario que si desea crearla en las esquinas de la línea, deberá ingresar un cero ya sea en el tiempo siguiente o previo para identificar la posición de esta nueva estación.

**Red Metro.**

Atributos.

* nameRed.

Métodos.

**Línea.**

Atributos.

* nameLine.
* arrayStation.

Métodos.

* constructorLine.
* getarrayStation.
* setarratStation.
* createLine.
* deleteLine.
* addStation\_array.
* deleteStation\_array
* calcularTime.
* contStation\_Line
* stationInLine

**Estación.**

Atributos.

* nameStation.
* timePrev.
* timeNext.

Métodos.

* constructorStation.
* getStation.
* setStation.
* createStation.
* deleteStation.