

# Informe Desafio 2

Por: Kevin Daniel Uribe Pérez

Profesores: Augusto Salazar, Aníbal Guerra.

Informática II

2024-1



## **Análisis y consideraciones del problema:**

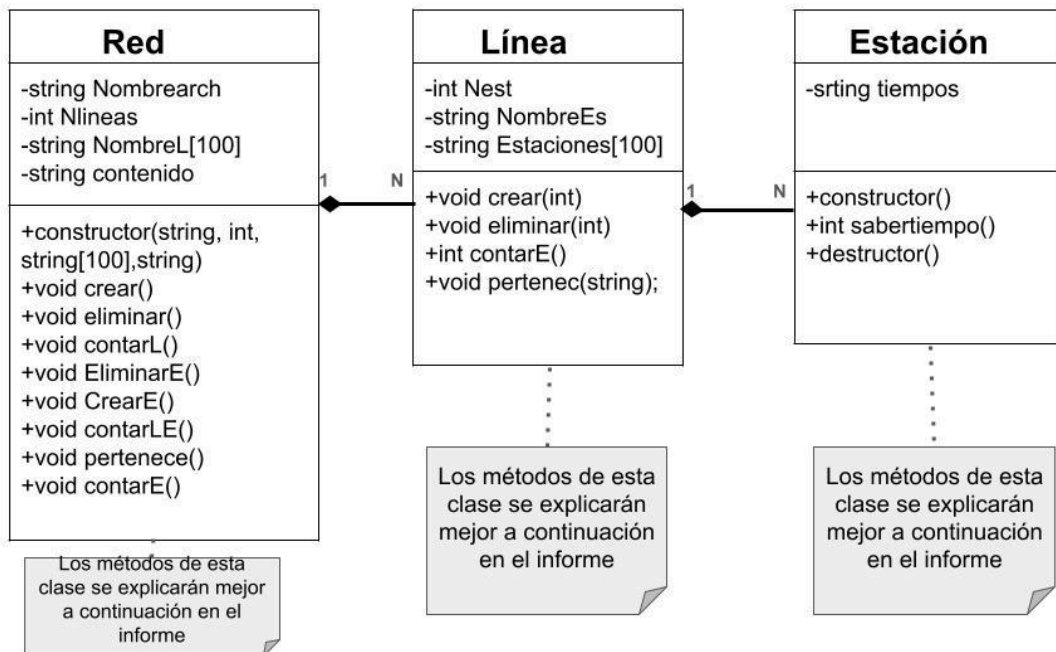
- Tengo que hacer una clase red esta clase tendrá como atributos un nombre, un número de líneas (N) y un arreglo dinámico de tamaño N que contendrá los strings de los nombres de las líneas, sus métodos serán crear una nueva línea, eliminar una línea, saber cuántas líneas tiene una red.
- Tengo que hacer una clase línea que tendrá como atributos un nombre, un número de estaciones (N) y un arreglo dinámico de tamaño N que contendrá los strings de los nombres de las estaciones, sus métodos serán crear una nueva estación, eliminar una estación, saber cuántas estaciones tiene una línea, saber cuantas estaciones de transferencia tiene una línea, saber cuantas estaciones de no transferencia tiene una línea, saber si una estación pertenece a esa línea.
- Tengo que hacer una clase estación que tendrá como atributos un nombre, un nombre de destino, un número de estaciones (N), un arreglo dinámico de tamaño N que contendrá los nombres de las estaciones y un arreglo dinámico de tamaño N que contendrá los tiempos que tarda esa estación a las otras estaciones en caso de ella misma será 0s, sus métodos saber el tiempo de viajar a la estación especificada.
- Para eliminar una línea hay que comprobar que no haya más de una línea ya que todas están conectadas si hay solo una línea se borra.

- Para eliminar una estación hay que mirar que en su nombre no haya un guión
- Los datos de la red se van a guardar en un archivo de texto llamado red.txt con el siguiente formato para facilitar la recolección de ellos.

```
//nombre de red.  
RED Medellín.  
//Nombres de líneas.  
A,B,T.  
//Nombres de estaciones  
Estrella,San Antonio-A,Niquia.  
San Antonio-B, San Javier.  
San Antonio-T.  
//Tiempos de viaje  
2,2.  
3,1.  
0.
```

- Para eliminar una estación hay que mirar que en su nombre no haya un guión.
- Para agregar una línea o una estación hay que mirar que la línea no exista en la estación o que la estación no existe en la línea.
- Al contar las estaciones solo se contarán las estaciones que no son de transferencia, las estaciones de transferencia se meterán en un arreglo dinámico en el cual se mirara que ya no existan en dicho arreglo y al final las contara.
- la clase línea tendrá polimorfismo con la clase Red.
- Crear menú para poder seleccionar dichas funcionalidades.

## Diagrama de clases de la solución planteada.



## Lógica de las tareas planteadas.

- En el programa principal se llamará a una función la cual será un menú, considero que este programa tiene una solución trivial ya que simplemente es hacer un bucle que no salga hasta que el usuario de la opción de salir, al salir del menú se llamará a otra función que guarda todos los cambios realizados mientras el programa esté activo y así no se borren las estaciones, esto se hará guardarlo en un archivo de texto con el nombre de la red.
- El método crear, pedirá a un usuario que ingrese el nombre de la estación o de la línea que

quiere agregar, además de esto en caso de que sea una estación se deberá preguntar si la quiere agregar en un extremo o entre dos estaciones, en el primer caso se debe preguntar qué extremo y en el segundo caso se debe preguntar entre que dos estaciones se desea crearla, se le dará las opciones posibles para evitar errores de ingreso de datos, en caso de ser una línea preguntara con cuál línea quiere hacer intercesión en caso de que exista solo una se elige esa y en caso de que no haya ninguna no se intersecta, luego de seleccionar la línea con la cual hace intercesión se debe seleccionar la estación de transferencia

- El método eliminar elimina una línea solo si solo existe esa línea, eliminará una estación siempre y cuando la estación no tenga un guión en su nombre.
- El método contar solo recibe un arreglo de strings y los cuenta al igual que todos los métodos que dicen contar en su nombre.
- El método estación pertenece revisa si la estación pertenece al arreglo dinámico de strings.
- El método saber tiempo, pregunta entre qué estaciones quiere averiguar el tiempo de viaje en caso de que solo haya una estación el resultado será la hora actual, en caso contrario será la suma de la hora actual más el tiempo que se demore.
- Además de esto tengo que crear una función que abra el documento y organice los datos en un

arreglo dinámico de dos dimensiones para guardar líneas y estaciones.

## **Algoritmos implementados.**

### **Problemas de desarrollo que afronte.**

- Quise hacer la mayoría de métodos usando arreglos dinámicos pero cuando intentaba que fueran argumentos, me pasaba un error de syntaxis.
- Por ahora tengo un error al crear una estación con el mismo nombre que una de transferencia. para solucionar este problema voy a impedirle al usuario que ingrese un nombre que contenga un carácter especial.
- No tenía muy claro cómo crear un arreglo que contuviera objetos de una clase específica, este error lo solucione creando un constructor default que crea el objeto cuando no se le mandan datos.
- Tengo un error en que las estaciones aún no se guardan, para esto creo que voy a actualizar el archivo con los datos de las estaciones al salir del método crear y eliminar.

### **Problemas de desarrollo que afronte.**

- Me di cuenta que la mayoría de métodos era mejor llamarlos desde la clase Red ya que es la que contiene a los Objetos de la clase Línea así como la clase Línea contiene los objetos de la clase estación.

- Cambie la opción de arreglos dinámicos por arreglos de dimensión específica.
- Me di cuenta que además de crear un arreglo con los nombres de las líneas, también es necesario crear un arreglo de la clase líneas.
- Quiero crear un archivo para crear funciones que suelo utilizar bastante en el mismo código para reducir líneas.