

ANÁLISIS DEL PROBLEMA

JUAN DIEGO TORO FRANCO

LUIS ANGEL OSORIO ALANDETE

INFORMÁTICA II

CONTEXTO PROBLEMA: Se nos pide crear un programa que simule una red de metro. Esta red está constituida de un conjunto de líneas las cuales van a tener una cantidad de estaciones, para que la red sea diseñada de la mejor manera posible, es preciso especificar que se va a tener unas limitaciones para la creación de esas líneas y sus respectivas estaciones. Estas son:

- **Para las líneas:**
 - a. Una línea sólo puede estar una vez en una red.
 - b. Si una red tiene más de una línea, estas líneas no pueden estar desconectadas.
- **Para las estaciones:**
 - a. Una estación puede pertenecer a varias líneas (estaciones de transferencia)
 - b. Una estación sólo puede estar una vez en una línea.

DISEÑO DE LA SOLUCIÓN: Para iniciar con el diseño del problema definimos cuales van a ser las clases. En la solución de este problema hemos decidido que van a ser tres: la clase red, la clase línea y la clase estación.

- **Clase red:** Esta clase va a tener un único atributo, un arreglo dinámico de punteros a objetos de la clase línea ¿Por qué punteros? En este problema usar punteros a objetos nos permite una mejor gestión en la eficiencia de la memoria ya que evita copias de los objetos en los arreglos. Los métodos básicos van a ser:
 - **Agregar línea:** Agrega una nueva línea a la Red, si no existe una línea con el mismo nombre.
 - **Entradas:** Nombre de la línea
 - **Salidas:** Confirma si la línea fue agregada o informa de un error si la línea ya existe
 - **Eliminar línea:** Elimina una línea de la red, si existe y no tiene estaciones de transición.
 - **Entradas:** Nombre de la línea
 - **Salidas:** Confirma si la línea fue eliminada o informa de un error.
 - **Buscar línea:** Busca y devuelve una referencia o puntero a una línea específica
 - **Entradas:** Nombre de la línea
 - **Salidas:** Puntero a la línea, o **nullptr** si no se encuentra
 - **Contar líneas:** Devuelve el número de líneas actualmente en la red
 - **Entradas:** Ninguna
 - **Salidas:** Números de líneas

- **Listar líneas:** Imprime los nombres de todas las líneas en la red
 - **Entradas:** Ninguna
 - **Salidas:** Ninguna
- **Contar estaciones de la red:** Calcula el número total de estaciones en toda la red
 - **Entradas:** Ninguna
 - **Salidas:** Número total de estaciones
- **Clase línea:** Esta clase va a tener dos atributos, el nombre de la línea y un arreglo dinámico que contiene puntero de objetos a clase estación, el uso de punteros a objetos tiene la misma justificación ya dada en la clase red. Los métodos más básicos van a ser:
 - **Agregar estación:** Agrega una estación a la línea, si no existe
 - **Entradas:** Nombre de la estación, nombre de la línea, tipo, tiempo a partir de sus estaciones contiguas
 - **Salidas:** Confirma si la estación fue agregada o informa de un error si no se logró
 -
 - **Eliminar estación:** Elimina una estación de la línea, si existe
 - **Entradas:** Nombre de la estación
 - **Salidas:** Confirma si la estación fue eliminada o informa de un error si la estación no existe
 - **Buscar estación:** Busca y devuelve una referencia o puntero a una estación específica en la línea
 - **Entradas:** Nombre de la estación
 - **Salidas:** Puntero a la estación o **nullptr** si no se encuentra
 - **Contador de estaciones:** Devuelve el número de estaciones en la línea
 - **Entradas:** Ninguna
 - **Salidas:** Número de estaciones
 - **Listar estaciones:** Imprime los nombres de todas las estaciones en la línea
 - **Entradas:** Ninguna
 - **Salidas:** Ninguna
- **Clase estación:** Esta clase va a tener cinco atributos, el nombre de la estación, el nombre de la línea a la que pertenece, si es de transición o normal, tiempo a la estación anterior y tiempo a la estación siguiente. Los métodos más básicos van a ser:
 - **Obtener nombre:** Devuelve el nombre de la estación
 - **Entradas:** Ninguna
 - **Salidas:** Nombre de la estación
 - **Obtener tipo:** Devuelve el tipo de estación
 - **Entradas:** Ninguna
 - **Salidas:** El tipo de estación
 - **Obtener tiempos:** Devuelve el tiempo respecto a sus estaciones contiguas

- **Entradas:** Ninguna
- **Salidas:** Tiempo anterior o siguiente
- **Modificar tiempos:** modifica los tiempos respecto a sus estaciones
 - **Entradas:** Nuevo tiempo
 - **Salidas:** Ninguna

Para el diseño de la solución aparte de establecer cuáles van a ser las clases y sus respectivos métodos y atributos, modelamos los menús y sus interacciones con el usuario.

MODELACIÓN DE MENÚS:

“Hola, bienvenido a la modelación de una red de metro: “

Menú red:

1. Crear red de metro
2. Salir del sistema

> [El usuario selecciona opción]

(Si selecciona crear red de metro)

“Primero definamos el nombre de la red: “

> [Ingresa nombre]

Menú línea:

1. Agregar línea
2. Eliminar línea
3. Cantidad de líneas en la red
4. Cantidad de estaciones en la red
5. Menú estaciones
6. Volver a menú red

(Si selecciona menú de estaciones)

Menú estaciones:

1. Agregar estación a una línea
2. Eliminar estación de una línea
3. Saber cantidad de estaciones de una línea
4. Volver al menú línea

INTERACCIÓN CON EL USUARIO:

- **Menú de línea:**

- Las opciones de agregar y eliminar le piden al usuario los nombres de la línea con la diferencia de que para agregar se hace la verificación de que la línea no exista y se le pide las líneas con las que se intercepta y en eliminar se verifica que la línea exista pero que no sea una línea en donde haya estaciones de transición.
- En cuanto a las opciones de cantidad de líneas y estaciones son iteraciones por los arreglos, en la que para el primer hace la iteración solo para el arreglo que contiene las líneas y en el segundo aparte de hacer la iteración mencionada anteriormente, recorre las estaciones para esas líneas y las va contando.

- **Menú de estación:**

- Las opciones de agregar y eliminar le piden al usuario los nombres de la estación con la diferencia de que para agregar se verifica que la estación no exista, se le pide si es de transición o normal, si quiere agregarla entre dos estaciones o de forma secuencial y los respectivos tiempos dependiendo que estaciones tiene contigua a ellas y en cuanto a eliminar solo se verifica que si exista la estación y que no sea de transición.
- La opción de saber estaciones de línea se le pide el nombre de la línea y se hace una iteración por las estaciones y las va contando.