Análisis del problema y consideraciones para la alternativa de solución propuesta:

El problema consiste en diseñar e implementar un simulador de Red de Metro que permita representar la estructura de una red metro cualquiera, así como ejecutar operaciones básicas como agregar/eliminar estaciones y líneas, consultar información de la red, etc.

Para abordar este problema, se propone utilizar el paradigma de Programación Orientada a Objetos (POO) para modelar las entidades principales: Estación, Línea y Red de Metro. Estas clases estarán relacionadas jerárquicamente, donde una Estación puede pertenecer a varias Líneas (si es una estación de transferencia), una Línea sólo puede pertenecer a una Red, y una Red contiene múltiples Líneas interconectadas.

Se diseñarán e implementarán las clases necesarias, junto con sus atributos, métodos y relaciones apropiadas, para representar adecuadamente la estructura de la Red de Metro y permitir la ejecución de las operaciones requeridas.

+---------------------+

| Estacion |

+---------------------+

- nombre: string

- tiempoSiguiente: int

- tiempoAnterior: int

- lineas: vector<Linea\*>

+---------------------+

+ Estacion(nombre, tiempoSig, tiempoAnt)

+ agregarLinea(Linea\*)

+ removerLinea(Linea\*)

+ esTransferencia(): bool

+ obtenerNombreCompleto(): string

+ ... (otros métodos)

+---------------------+

+---------------------+

| Linea |

+---------------------+

- nombre: string

- tipo: string (tren/tranvia)

- estaciones: vector<Estacion\*>

- red: Red\*

+---------------------+

+ Linea(nombre, tipo)

+ agregarEstacion(Estacion\*, posicion)

+ removerEstacion(Estacion\*)

+ contarEstaciones(): int

+ perteneceEstacion(Estacion\*): bool

+ ... (otros métodos)

+---------------------+

+---------------------+

| Red |

+---------------------+

- lineas: vector<Linea\*>

+---------------------+

+ Red()

+ agregarLinea(Linea\*)

+ removerLinea(Linea\*)

+ contarLineas(): int

+ contarEstaciones(): int

+ calcularTiempoLlegada(Estacion\*, Estacion\*): int

+ ... (otros métodos)

+---------------------+

Relaciones:

- Una Estacion puede pertenecer a varias Lineas (estaciones de transferencia)

- Una Linea pertenece a una sola Red

- Una Red contiene varias Lineas

En este diagrama, se muestran las tres clases principales (Estación, Línea y Red) con sus respectivos atributos y métodos. Además, se representan las relaciones entre ellas, donde una Estación puede pertenecer a varias Líneas, una Línea pertenece a una sola Red, y una Red contiene varias Líneas.

Las operaciones específicas como agregar/eliminar estaciones y líneas, consultar información de la red, y calcular el tiempo de llegada entre estaciones, se implementarán dentro de los métodos correspondientes en cada clase.