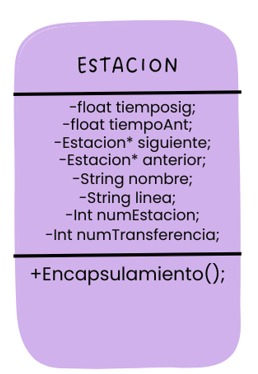
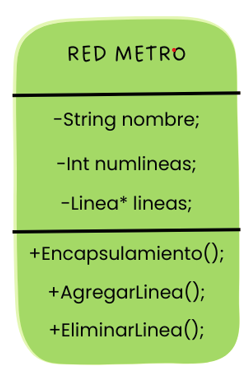
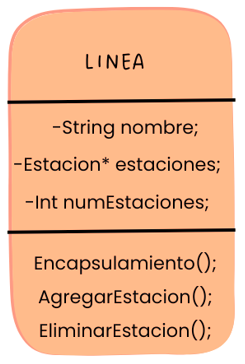
**Proceso de análisis y documentación - Desafío 2.**

1. **Análisis del problema y consideraciones para la alternativa de solución propuesta.**
2. **Diagrama de clases**

****

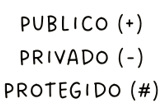
**2…n**

**1**

****

**n**b

**1**

****

**Explicación detallada de las clases, sus atributos y sus métodos.**

**Clase “RED METRO”:** Hemos decidido crear una clase llamada “red metro” con el fin de que esta contenga un conjunto de atributos y métodos que describan cómo se verá y se comportará un objeto de este tipo.

**Para la clase “red metro” hemos creado los siguientes atributos:**

* **String nombre:** Este atributo es de tipo string porque almacenará un nombre; es decir, un conjunto de caracteres; en este caso particular, el nombre que el usuario le pondrá a la “red metro”.
* **Int numlineas**: Este atributo almacenará un número entero positivo y diferente de cero; el cuál, representará la cantidad de líneas que compondrán la “red metro”.
* **Linea\* líneas:**

**Para la clase “red metro” hemos creado los siguientes métodos:**

* **Encapsulamiento:**
* **AgregarLinea:**
* **EliminarLinea:**

**¿Cuántos objetos tendrá la clase “red metro”?** Según nuestro análisis la clase “red metro” tendrá un solo objeto, pues nuestro programa sólo se encargará de modelar una sola red.

**Clase “LINEA”:** Hemos decidido crear una clase llamada “ Linea” con el fin de que esta contenga un conjunto de atributos y métodos que describan cómo se verá y se comportará un objeto de este tipo.

**Para la clase “línea” hemos creado los siguientes atributos:**

* **String nombre:** Este atributo es de tipo string porque almacenará un nombre; es decir, un conjunto de caracteres; en este caso particular, el nombre que el usuario le pondrá a cada una de las líneas ( objetos) que contendrá la “red metro”.
* **Int numEstaciones:** Este atributo almacenará un número entero positivo, diferente de cero, y mayor o igual a 2; el cuál, representará la cantidad de estaciones que compondrán cada línea(objeto) de la “red metro”.
* **Estacion\* estaciones:**

**Para la clase “línea” hemos creado los siguientes métodos:**

* **Encapsulamiento:**
* **AgregarEstacion:**
* **EliminarEstacion:**

**¿Cuántos objetos tendrá la clase “linea”?** La clase línea tendrá la cantidad de objetos que contenga el atributo definido como **Int numlineas** en la clase “red metro”.

**Clase “ESTACION”:** Hemos decidido crear una clase llamada “Estacion” con el fin de que esta contenga un conjunto de atributos y métodos que describan cómo se verá y se comportará un objeto de este tipo.

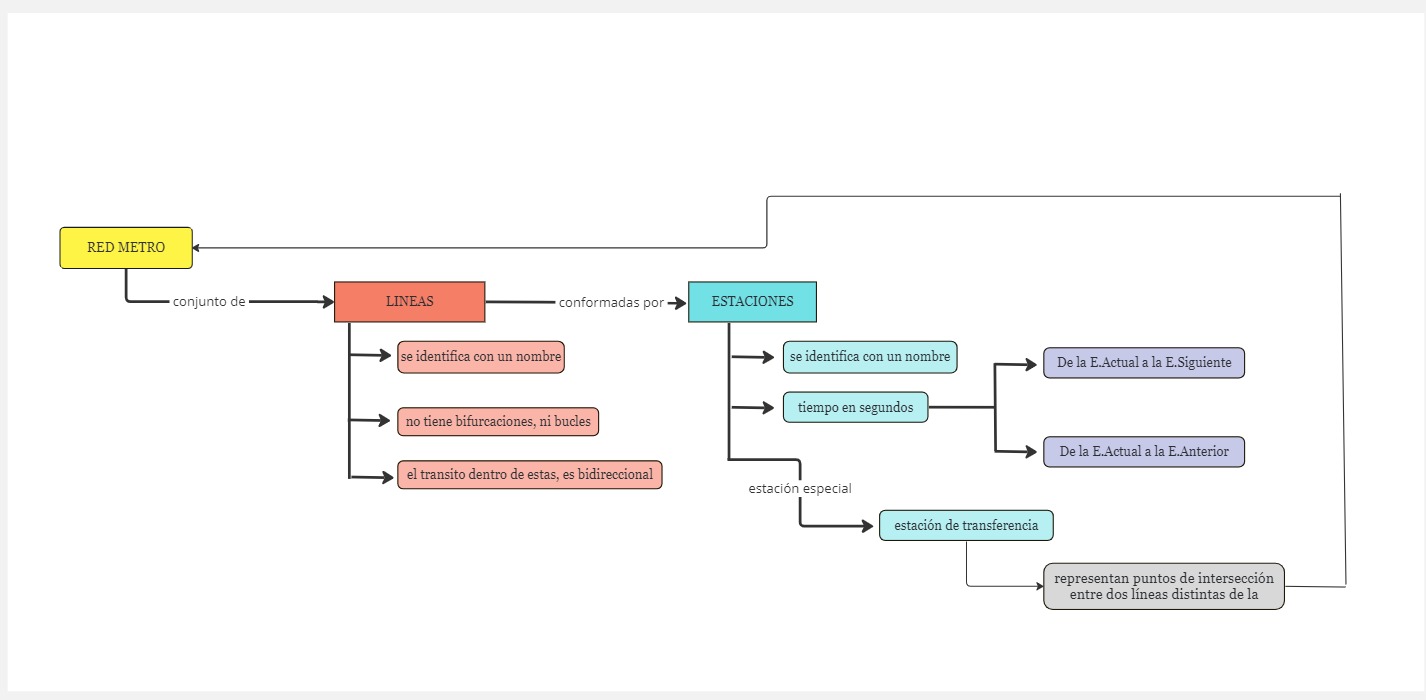
**Para la clase “estacion” hemos creado los siguientes atributos:**

* **Float tiemposiguiente:**
* **Float tiempoAnterior:**
* **Estacion\* siguiente:**
* **Estacion\* anterior:**
* **String nombre:**
* **String línea:**
* **Int numEstacion:**
* **Int numTransferencia:**

**Para la clase “estacion” hemos creado los siguientes métodos:**

**-Encapsulamiento:**

**¿Cuántos objetos tendrá la clase “estacion”?**

1. **Diagrama simplificado de las condiciones del problema.**

**Condiciones a tener en cuenta por parte del problema:**

1. Agregar una estación a una línea, en los extremos o en posiciones intermedias.
2. Eliminar una estación de una línea. No se pueden eliminar estaciones de transferencia.
3. Saber cuántas líneas tiene una red Metro.
4. Saber cuántas estaciones tiene una línea dada.
5. Saber si una estación dada pertenece a una línea específica.
6. Agregar una línea a la red Metro.
7. Eliminar una línea de la red Metro (sólo puede eliminarse si no posee estaciones de transferencia).
8. Saber cuántas estaciones tiene una red Metro (precaución con las estaciones de transferencia).

Teniendo en cuenta el esquema y las condiciones anteriores, hemos diseñado un plan de desarrollo con el fin de resolver el problema planteado en este desafío.

Se pondrán las ideas a manera de ítems; en donde cada uno de ellos, corresponderá a la descripción de las estrategias que se utilizarán para resolver cada una de las tareas.

**Estrategia de solución para el inciso “A”:**