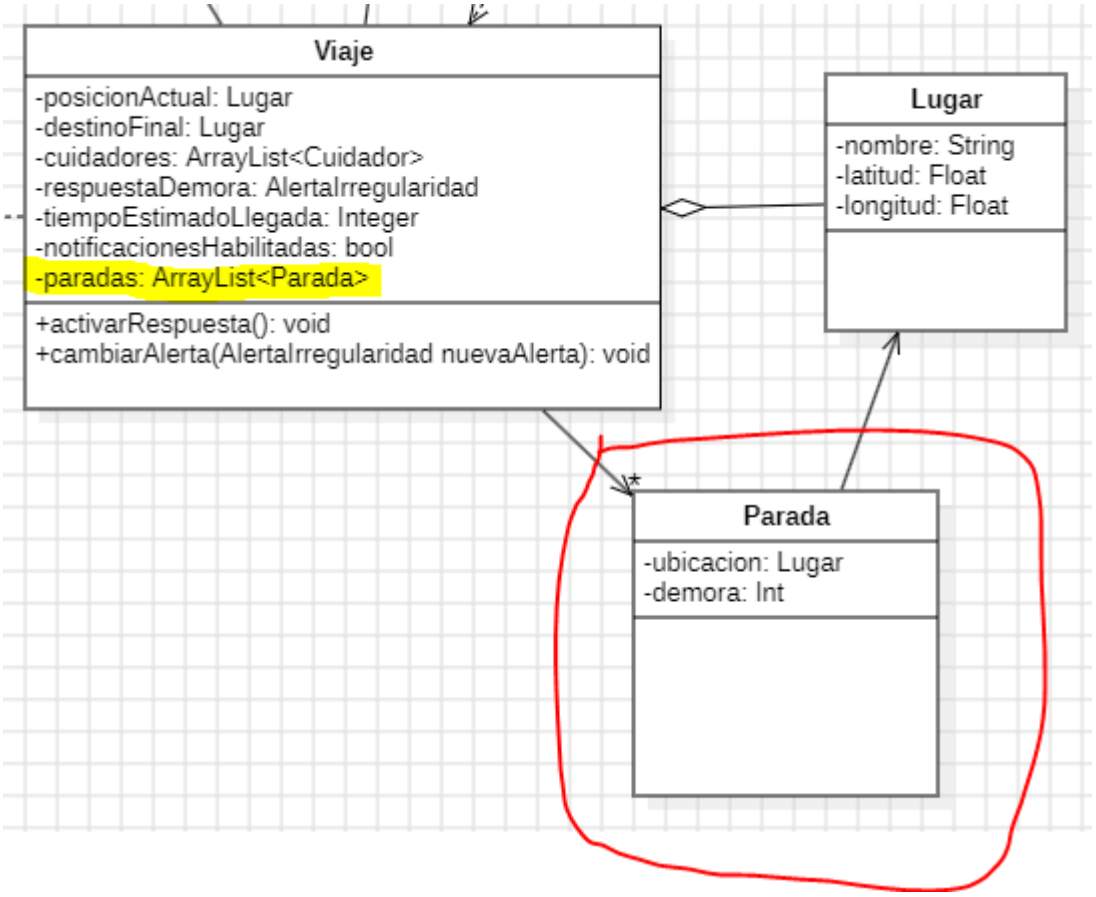
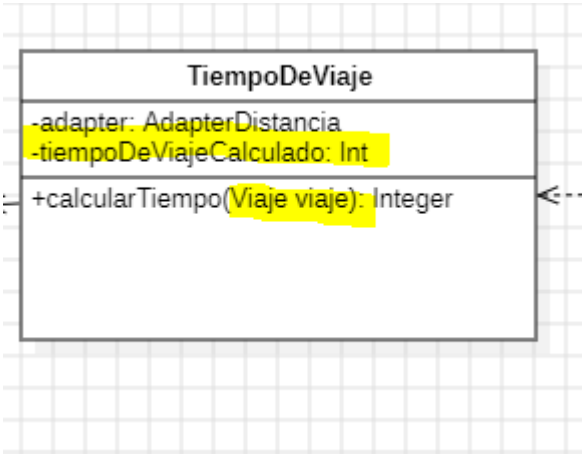
Punto 2:

Realizamos un código parcial de la solución, asumimos que se creará una clase intermedia llamada “Parada” la cual tendrá el atributo de la demora, la cual el “transeúnte” determinaría previamente, además del propio lugar. Mientras que “Viaje” poseería como atributo un “ArrayList” de paradas.



Dicho esto nuestra solución sería determinar que si existe por lo menos 1 parada entonces el “transeúnte” habría de tener que calcular las demoras y las distancias correspondientes. Esto lo hace corroborando en un “if” si es así y procediendo a ejecutar un código similar a este. Cabe aclarar que en esta situación estaría recibiendo un “Viaje” y no solamente dos “lugares”, junto a un nuevo atributo para no manipular de manera constante el dato que se almacena en transeúnte.



Código parcial:

```java
public class TiempoDeViaje {

    no usages
    AdapterDistancia adapter;
    3 usages
    int tiempoDeViajeCalculado=0;


    no usages
    public static int calcularTiempo(Viaje viaje)
    {

        if(viaje.paradas.size()!=0){
        for(int i=0;viaje.paradas.size()>i;i++){
            tiempoDeViajeCalculado=API.distancia(viaje.posicionActual,viaje.paradas.get(i).getUbicacion());
            tiempoDeViajeCalculado+=viaje.paradas.get(i).getDemora();
            viaje.setPosicionActual(viaje.paradas.get(i).getUbicacion());
        }
        }
        return tiempoDeViajeCalculado+API.distancia(viaje.posicionActual,viaje.destinoFinal);
    }

}
```