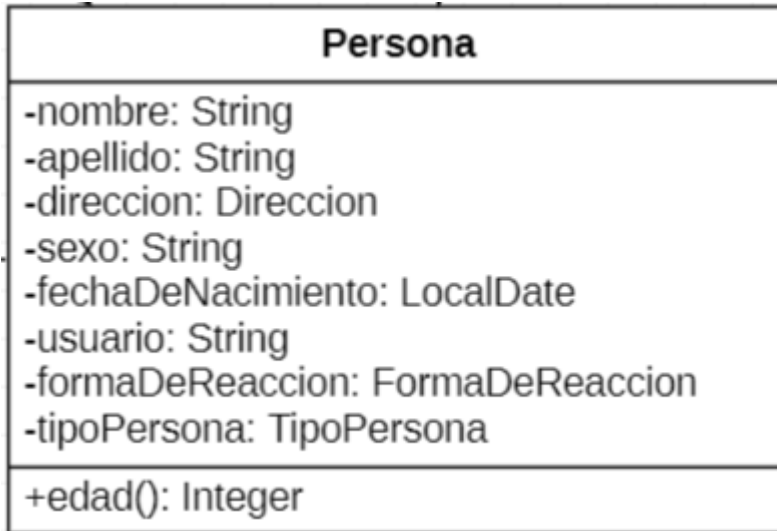
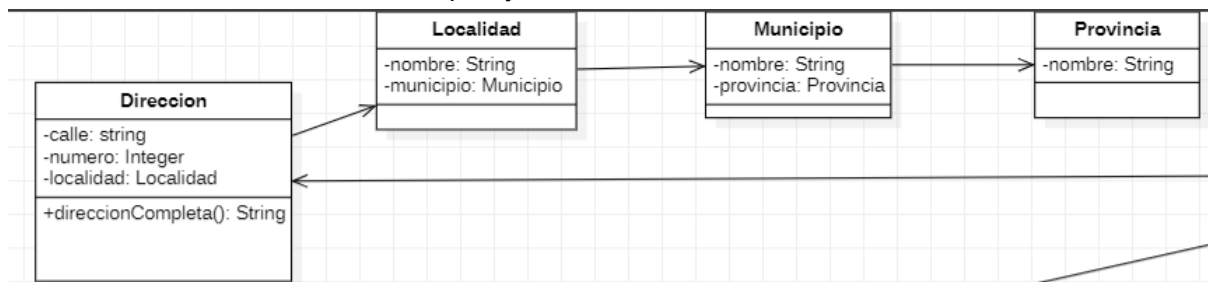


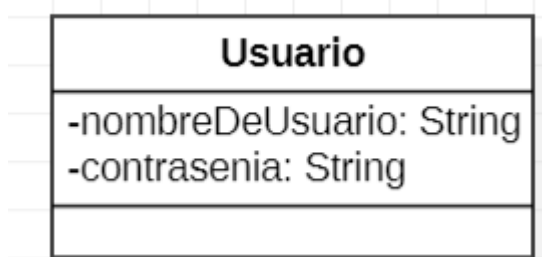
-Los transeúntes y los cuidadores son personas. Pensé a ambos como una única clase. Las personas tienen como atributo nombre, apellido, dirección, sexo, fechaDeNacimiento(se calcula la edad con el método edad()), un usuario, una forma de reaccion y un tipoDePersona(ENUM) para saber si es cuidador o transeuente. El transeúnte configura de qué forma reaccionará el sistema frente a un incidente. :



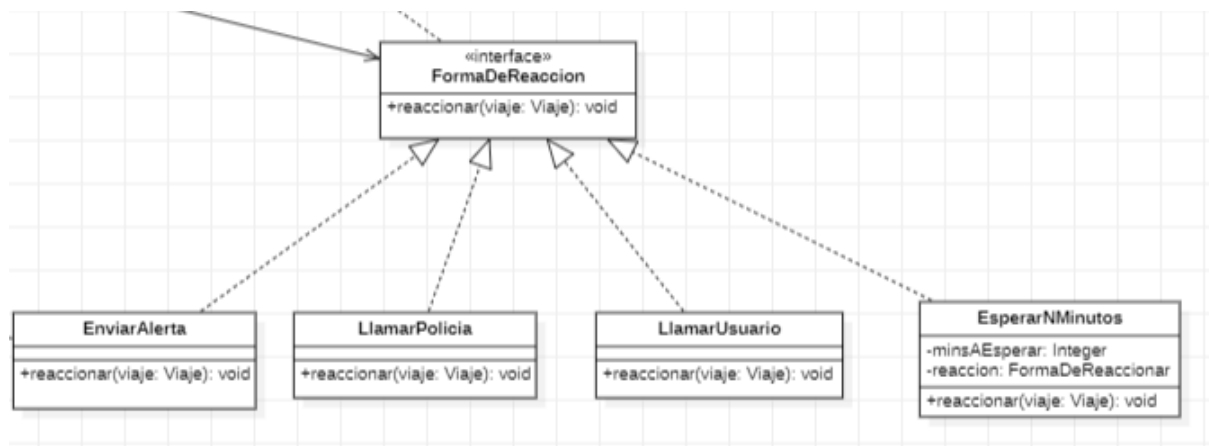
La dirección: nos interesa su punto de origen y sus destino. La clase Dirección dentro varios atributos(calle, numero, localidad) y conexión con la clases Localidad, que a su vez tiene conexión con la clase Municipio, y esta a su vez tiene conexión con la clase Provincia:



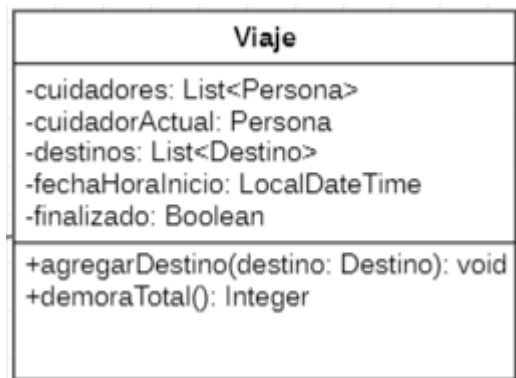
Clase Usuarios: Decí crear una clase para las credenciales de inicio de sesión para facilitar la cohesión y tener menos acoplamiento:



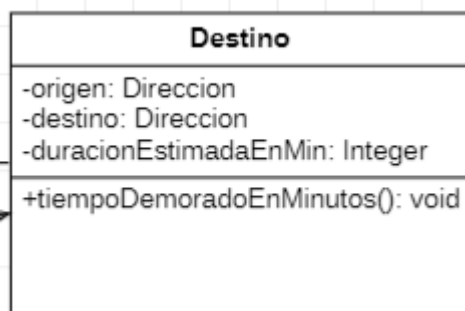
Para la formaDeReaccion que se encuentra un atributo en persona realice una interfaz de FormaDeReaccion que tiene el método reaccionar que recibe como parametro el viaje. Cree también las clases enviarAlerta, llamarPolicia, llamarUsuario, esperarNMinutos que implementa la interfaz FormaDeReaccion. Con esta solución en caso de querer agregar una nueva forma de reacción se podrá fácilmente creando una nueva clase que implemente la interfaz:



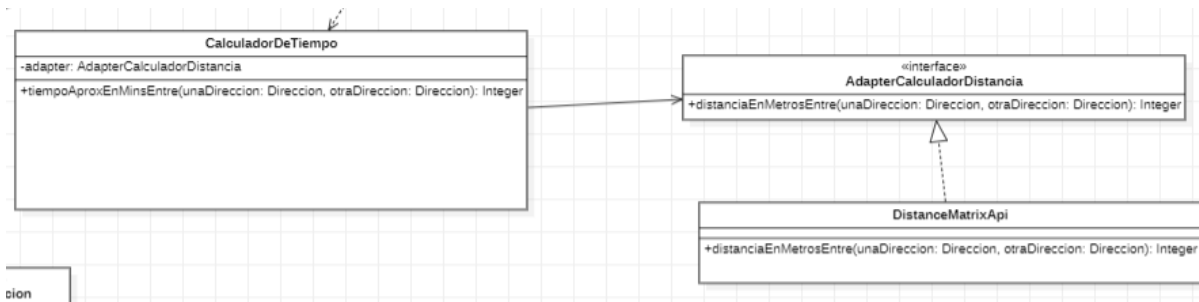
Se crea una clase **Viaje**, que tendrá varios atributos como los cuidadores (lista de Personas), cuidador actual, una lista de destinos, una fecha y hora de inicio de viaje y un booleano que representa si el viaje está finalizado o no. Además cree los métodos `agregarDestino` que en caso de querer agregar un destino al viaje se realice fácilmente. También tenemos el método `demora total` que calcula la demora total del viaje:



Cree una clase **Destino** para poder representar el punto 2 del modelo de dominio que va a tener el origen (para guardar donde se encuentra), el destino y la duración estimada para guardar mediante el método `tiempo demorado en minutos`:



Se crea una clase **CalculadoraDeTiempo**, que tendrá un método `calcularTiempoAproximadoEntre` que recibe como parámetro dos Direcciones y devuelve un entero que representa al tiempo en minutos que se tarda en recorrer la distancia entre esas dos direcciones. Cree una interfaz **AdapterCalculadorDistancia** que tenga un método `distanciaEnMetrosEntre`, el cual recibe dos Direcciones y devuelve un entero representando a la distancia en metros. Este ADAPTER, entonces, calculará la distancia. Además se crea una clase **DistanceMatrixAPI** que implemente la interfaz **AdapterCalculadorDistancia** (el ADAPTER):



Parea las notificaciones cree una clase Notificador, que tendrá un método `notificar()` que reciba como parámetro una **Notificación**, clase que tendrá como atributos a un emisor, a un receptor y a un mensaje. Se modela un STRATEGY con las estrategias de notificación que por el momento serán **NotificacionWhatsApp** y **notificacionPush**:

