

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III
Curso 2
Segundo cuatrimestre de 2021

Alumno:	RIZZO EHRENBLOCK, Gonzalo Daniel
Número de padrón:	106475
Email:	grizzoe@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	2
4. Detalles de implementación	4
4.1. Clase AlgoRemis	4
4.2. Chofer y Autos	4
4.3. Viaje	4
4.4. Descuentos	4
5. Excepciones	4
6. Diagramas de secuencia	5

1. Introducción

El presente informe reúne la documentación de la solución del primer trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar un sistema de remisería utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

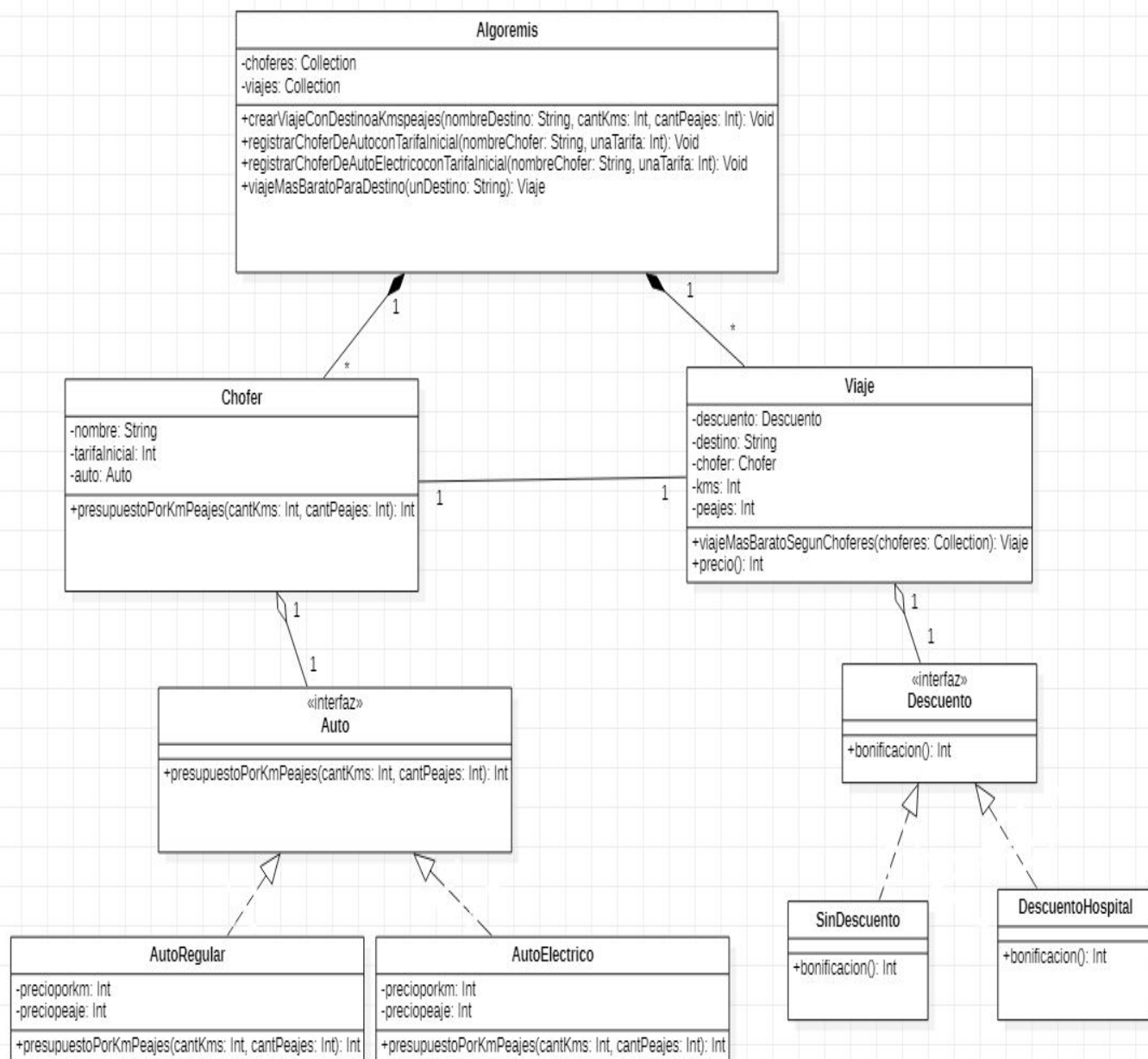
2. Supuestos

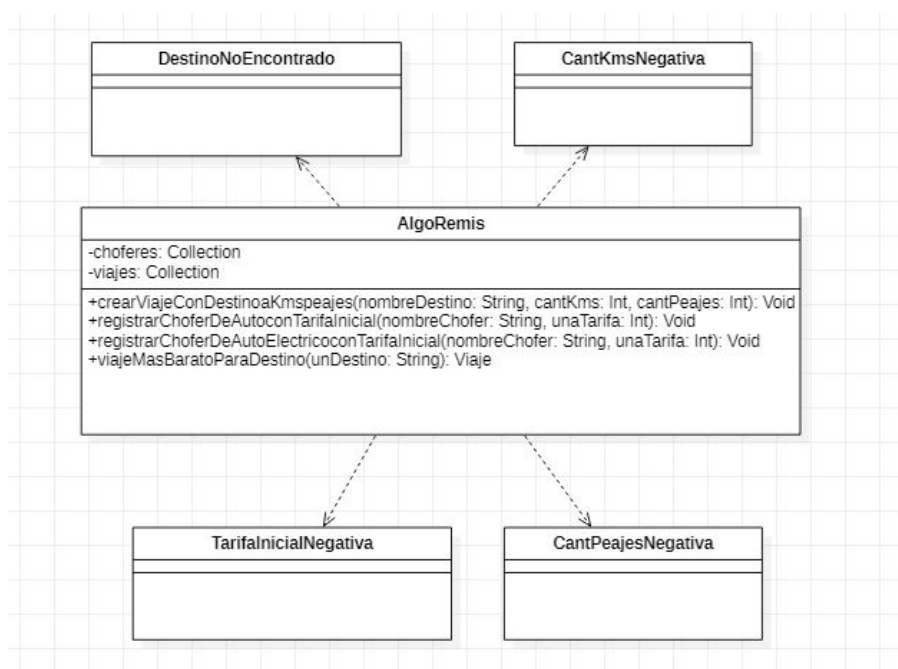
- Al crear un viaje, este se crea sin chofer, y se le asigna uno recién cuando el usuario quiera buscar el viaje mas barato para su destino, en ese momento es donde se le asigna el chofer (que sera el mas barato de todos los choferes disponibles segun la cantidad de peajes, kilometros, tarifa del chofer y características de su auto).
- Cada chofer solo puede atender un viaje a la vez.
- Las tarifas de peaje solo varían segun el tipo de vehiculo, no segun ubicacion.
- No habrán viajes al mismo destino pedidos por un mismo usuario.

3. Diagramas de clase

Estos diagramas muestran las clases utilizadas y las relaciones entre ellas. Se omiten algunas para no complicar la lectura de los diagramas. En el primero se muestran las principales clases del trabajo.

En el segundo diagrama se pueden ver las relaciones entre la clase AlgoRemis y sus excepciones.





4. Detalles de implementación

4.1. Clase AlgoRemis

Posee una colección de choferes y otra de viajes. Delega todas las responsabilidades a las clases Viaje y Chofer, no tiene más lógica que buscar dentro de las colecciones y delegar. Con esta clase podemos ver los conceptos de delegación, abstracción y tell don't ask ya que AlgoRemis solo da instrucciones para que se ejecuten los métodos de las demás..

4.2. Chofer y Autos

Cada chofer puede tener un Auto. Como este auto puede ser Regular o Electrico, decidí representar esto con la interfaz Auto, según de que tipo sea, cambiara el precio por kilometro y el precio de los peajes. Las interfaces son útiles para lograr una solución polimorfa y si se llegan a agregar más tipos de autos solo tengo que agregarlos y no cambiar todo mi código.

4.3. Viaje

Al crear un viaje, este se inicializa sin chofer, esto es así para que luego al ejecutar el método `viajeMasBaratoSegunChoferes: choferes` se busca entre los choferes y se asigna el más conveniente.

4.4. Descuentos

Cada viaje puede tener solo un tipo de descuento. Decidí representar esto la interfaz Descuento, según de que tipo sea cambiara el precio final del viaje.

5. Excepciones

CantPeajesNegativa, TarifaInicialNegativa, CantKmsNegativa Fueron creadas para evitar que el usuario introduzca números negativos donde no corresponde.

DestinoNoEncontrado fueron creadas para informar que el destino ingresado no se encuentra en el sistema, por lo que no se realiza la operación de pretendía hacer el usuario.

6. Diagramas de secuencia

A continuación se detallan los diagramas de secuencia de las principales y más relevantes acciones. Se omiten algunas para no complicar la lectura de los diagramas.

