



**Universidad
de Concepción**

Ingeniería de software I:
Iteración 2 Proyecto semestral: Sistema
de viajes

Integrantes:

Lisette Morales
Danilo Muñoz
Diego Muñoz
Pablo Venegas
Daniel Vergara

Matrícula:

2014415149
2015447737
2015406615
2013462020
2016406506

Docente: Gonzalo Rojas

Concepción 29 de Noviembre 2018

Índice

1. Introducción:	2
2. Descripción y requerimientos	3
2.1. Descripción del problema	3
2.2. Requerimientos funcionales	3
2.3. Requerimientos no funcionales:	3
3. Análisis:	4
3.1. Casos de uso:	4
3.2. Casos de usos implementados:	5
3.3. Documentación:	6
3.4. Esquema conceptual sin métodos:	14
4. Diseño:	15
4.1. Diagramas de comunicación:	15
4.2. Diagrama de clases:	17
5. Conclusión	18
6. Glosario:	19

1. Introducción:

El trabajo de un ingeniero de software es concebir e implementar un programa informático que satisfaga las necesidades de los clientes o de la comunidad en general, dependiendo de la intención del profesional. A la hora de realizar esta importante y dificultosa labor, un buen modelo y una documentación clara es esencial para reducir tiempo y costos de mantención; y hacer posible la reutilización del software y/o de sus módulos funcionales.

Nuestro proyecto tiene un enfoque iterativo e incremental con dos iteraciones, basado en UML. En esta segunda versión se han corregido los errores anteriores e implementado completamente los casos de usos faltantes.

A lo largo de este informe se exponen capítulos, entre ellos están, requerimientos de software, casos de usos, documentación, diagramas de clases y finalmente diagramas de comunicación de nuestro software con implementación funcional de casos de usos, faltando corregir detalles.

2. Descripción y requerimientos

2.1. Descripción del problema

El problema a enfrentar consiste en desarrollar un sistema de software carpooling, que permita a conductores ofrecer asientos libres cuando realicen un viaje a personas interesadas en el mismo viaje o trayectos intermedios. De este modo, conductores y pasajeros ahorran dinero, junto con reducir la contaminación y congestión en un viaje potencialmente más ameno.

2.2. Requerimientos funcionales

- Permitir que pasajeros tengan su propio perfil, mostrando datos, preferencias e intereses.
- Permitir que un conductor tenga su propio perfil con información personal y del vehículo.
- Permitir que un conductor publique viajes, especificando paradas, plazas disponibles y prestaciones adicionales.
- Permitir al pasajero buscar viajes por origen, destino y fecha.
- Permitir que el pasajero agregue filtros de búsqueda para consultar ofertas para un viaje.
- Permitir al pasajero realizar una solicitud de viaje permitiendo consultar prestaciones adicionales.
- Permitir que el pasajero consulte información geográfica de viajes disponibles.
- Permitir al pasajero ver el perfil del conductor que publica un viaje.
- Permitir al conductor ver el perfil de los pasajeros que envíen solicitudes para realizar el viaje.
- Permitir al conductor aceptar/rechazar solicitudes de pasajeros para realizar el viaje.
- Permitir a pasajeros y conductor valorar y comentar el viaje.

2.3. Requerimientos no funcionales:

- Intuitivo: las consultas y operaciones deben ser sencillas, evitando sobrecargar de información y/o deberes al usuario.
- Eficiente: el software debe responder de manera rápida y precisa a las necesidades del usuario.
- Escalable: el software debe estar construido para crecer tanto en los datos que debe manejar, como en la cantidad y la calidad de las funciones.
- Seguro: protección de los datos, software y plataforma de tecnología de posibles pérdidas, actividades no permitidas o uso para propósitos no establecidos previamente.

3. Análisis:

3.1. Casos de uso:

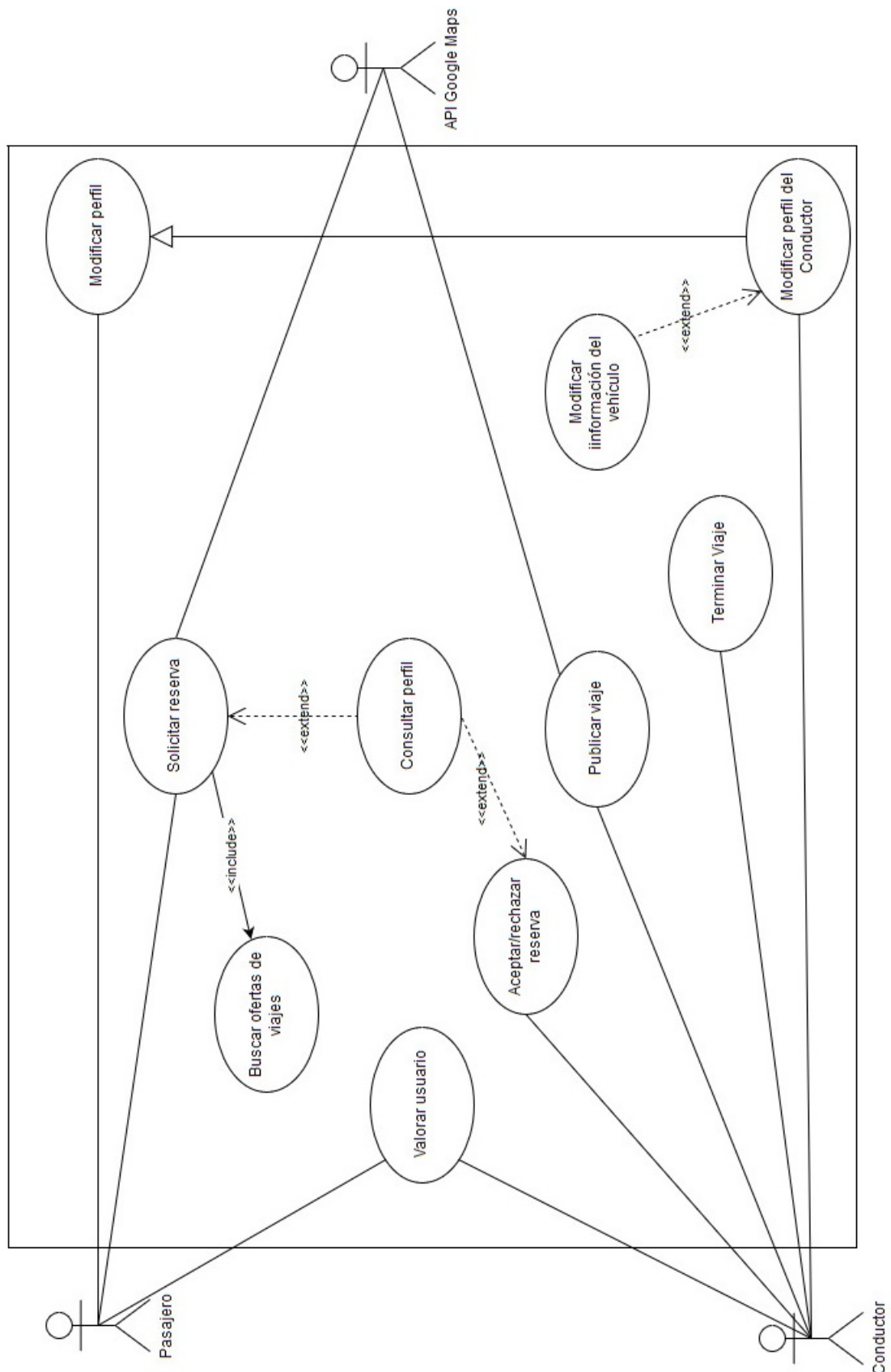


Figura 1: Casos de uso proyecto semestral sistema de viajes.

3.2. Casos de usos implementados:

Primera iteración:

- Solicitar reserva
- Buscar ofertas de viaje
- Publicar viaje
- Aceptar o Rechazar reservas.

Segunda iteración:

- Modificar perfil
- Modificar perfil del conductor
- Modificar información del vehículo
- Consultar perfil
- Terminar viaje
- Valorar usuario

3.3. Documentación:

- Caso de uso UC1: **Buscar Ofertas de Viajes**

1. Actor Principal e intereses: *Pasajero*
2. Personal Involucrado:
 - *Pasajero*: quiere buscar un viaje según sus preferencias.
 - *APIGoogleMaps*: quiere recibir información de origen y destino.
3. Precondiciones:
4. Postcondiciones: El sistema entrega los resultados de viajes disponibles que concuerdan con las preferencias del pasajero.
5. Escenario principal de éxito.
 - a) El pasajero ingresa comunas de origen y destino, fecha del viaje.
 - b) El sistema despliega los viajes compatibles con las preferencias.
6. Extensiones:
 - c' El pasajero no encuentra viajes disponibles según sus preferencias.
 - o El sistema le da la opción de cambiar los filtros de búsqueda.

■ Caso de uso UC2: **Publicar Viaje**

1. Actor Principal: *Conductor*
2. Personal Involucrado e intereses:
 - *Conductor*: desea publicar un viaje, para que los demás usuarios puedan reservar plazas disponibles.
 - *APIGoogleMaps*: recibe información del viaje (origen, paradas intermedias, destino) para proyectarla en el mapa.
3. Precondiciones:
4. Postcondiciones: El sistema registra el viaje.
5. Escenario principal de éxito:
 - a) Conductor ingresa: origen, paradas, prestaciones adicionales y destino del viaje, junto a la hora aproximada en que pasará por cada una de las ubicaciones, la fecha en que se llevará a cabo el trayecto y el número de plazas disponibles para el viaje.
 - b) El sistema envía parámetros de la ruta a la API Google Maps.
 - c) API Google Maps despliega el mapa de la ruta.
 - d) El viaje se registra en el sistema.
6. Extensiones:
 - +El API Google Maps no envía datos del mapa.
 - d' ○ El sistema no despliega el mapa y emite un mensaje de error.

■ Caso de uso UC3: **Solicitar Reserva**

1. Actor Principal: *Pasajero*
2. Personal Involucrado e intereses:
 - *Pasajero*: quiere enviar una solicitud de reserva para realizar un viaje.
 - *APIGoogleMaps*: quiere desplegar la ruta correctamente en el mapa.
3. Precondiciones: El pasajero ha seleccionado un viaje según sus preferencias.
4. Postcondiciones: El pasajero realiza con éxito su solicitud de reserva de viaje.
5. Escenario principal de éxito:
 - a) Ejecutar UC1
 - b) Pasajero selecciona el viaje para el que desea hacer una solicitud de reserva.
 - c) El Sistema despliega la información de fecha, cantidad de plazas disponibles en cada tramo del viaje, prestaciones adicionales y las paradas del viaje con sus horas aproximadas y API Google Maps muestra el mapa del viaje.
 - d) El Pasajero ingresa cantidad de plazas a reservar y confirma la solicitud de reserva.
 - e) El Sistema registra la solicitud de reserva.
6. Extensiones:
 - a' La API Google Maps no envía datos del mapa.
 - El sistema no despliega el mapa y emite un mensaje de error.
 - b' El viaje no dispone de plazas.
 - El sistema informa al pasajero que no quedan suficientes plazas disponibles para el tramo que desea reservar.

■ Caso de uso UC4: **Aceptar o rechazar reservas**

1. Actor Principal: *Conductor*
2. Personal Involucrado e intereses:
 - *Conductor*: quiere aceptar o rechazar una reserva.
 - *Pasajero*: recibe información si fue aceptada o rechazada la reserva.
3. Precondiciones: Tener publicado al menos un viaje.
4. Postcondiciones: Se registra como aceptada o rechazada la solicitud, y se notifica a los pasajeros correspondientes.
5. Escenario principal de éxito:
 - a) El Sistema despliega la lista de reservas del día, con pasajeros que pidieron reserva para un viaje.
 - b) El Conductor acepta o rechaza la solicitud de un pasajero.
 - c) El Sistema envía las notificaciones a los pasajeros para informar de la resolución tomada por el conductor.
6. Extensiones:
 - b' El Conductor puede acceder a los perfiles públicos de los pasajeros que enviaron solicitudes de reserva.

Caso de uso UC5: **Terminar viaje**

1. Actor Principal: *Conductor*
2. Personal Involucrado e intereses:
 - *Conductor*: Avisar al sistema que un viaje ha sido realizado.
3. Precondiciones: El conductor debe tener viajes creados.
4. Postcondiciones: El conductor confirma que finalizó un viaje.
5. Escenario principal de éxito:
 - a) El Sistema despliega un lista de los viajes no terminados.
 - b) El Conductor selecciona el que desea terminar.
 - c) El Sistema registra el viaje como terminado y activa las valoraciones para el conductor y pasajeros que realizaron el viaje.
6. Extensiones:

Caso de uso UC6: **Modificar perfil**

1. Actor Principal: *Pasajero*
2. Personal Involucrado e intereses:
 - *Pasajero*: quiere modificar su perfil.
3. Precondiciones: El pasajero debe tener un perfil creado.
4. Postcondiciones: El sistema registra la modificación del perfil del pasajero.
5. Escenario principal de éxito:
 - a) El pasajero ingresa a su perfil.
 - b) El sistema muestra el perfil.
 - c) El pasajero selecciona el campo que desea modificar de su perfil.
 - d) El sistema registra la modificación y actualiza el perfil del pasajero.
6. Extensiones:
 - c' El sistema no registra la modificación del perfil del pasajero y emite un mensaje de error

Caso de uso UC7: **Modificar perfil del conductor**

1. Actor Principal e intereses: *Conductor*
2. Personal Involucrado:
 - *Conductor*: quiere modificar su perfil.
3. Precondiciones: El conductor debe tener un perfil creado.
4. Postcondiciones: el sistema registra la modificación del perfil del conductor y/o del vehículo.
5. Escenario principal de éxito:
 - a) El conductor modifica su perfil llama a UC6
 - b) El conductor selecciona el campo que desea modificar del vehículo.
 - c) El sistema registra la modificación y actualiza el perfil del conductor.
6. Extensiones:
 - c' El Sistema no registra la modificación del perfil del conductor y emite un mensaje de error.

Caso de uso UC8: **Valorar Usuario**

1. Actor Principal e intereses: *Pasajero, Conductor*
2. Personal Involucrado:
 - *Pasajero*: quiere valorar a un usuario con un rango de 1 a 5.
 - *Conductor*:
3. Precondiciones:
4. Postcondiciones: el usuario es valorado en un rango de 1 a 5.
 - a) Si el Usuario es Conductor, el Sistema despliega una lista con todos los pasajeros que hayan realizado un viaje con él y que no hayan sido valorados. Si es Pasajero, el Sistema muestra una lista con los Conductores con los que realizó un viaje y no ha valorado.
 - b) El Usuario selecciona la calificación e ingresa un comentario.
 - c) El Sistema registra la valoración.
5. Extensiones:

3.4. Esquema conceptual sin métodos:

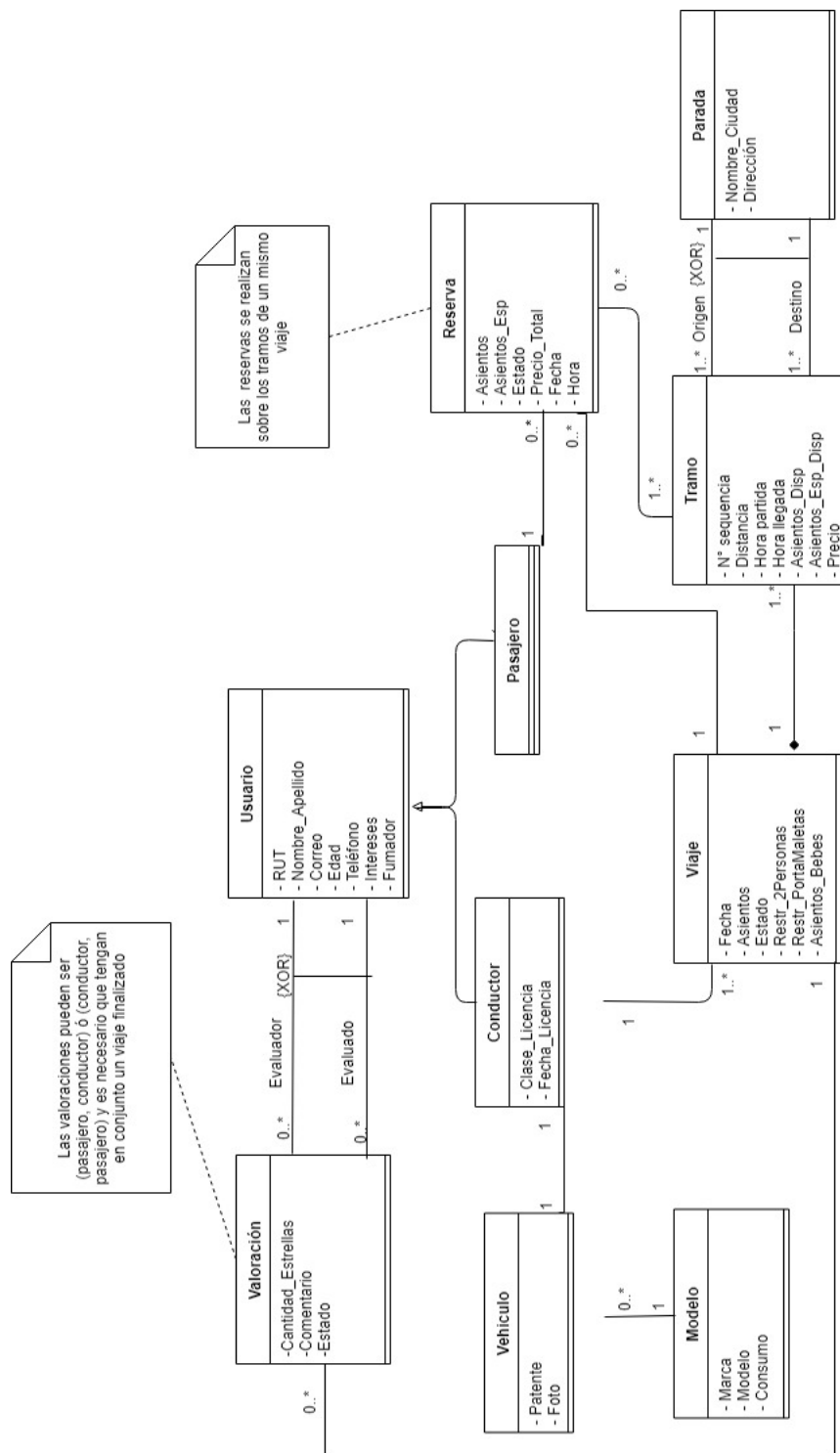


Figura 2: Diagrama de clases sin métodos del proyecto semestral sistema de viajes.

4. Diseño:

4.1. Diagramas de comunicación:

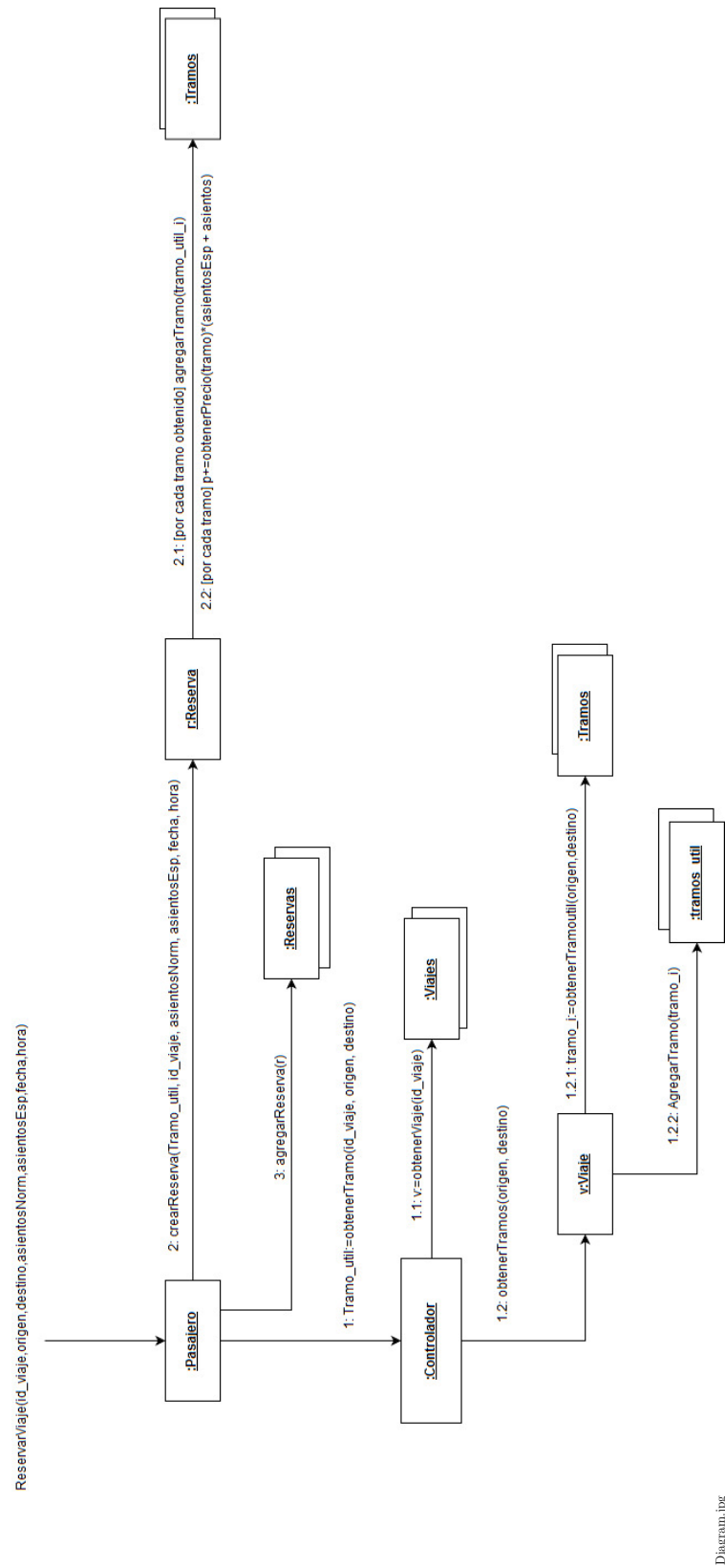


Figura 3: Diagrama de comunicación Reservar Viaje.

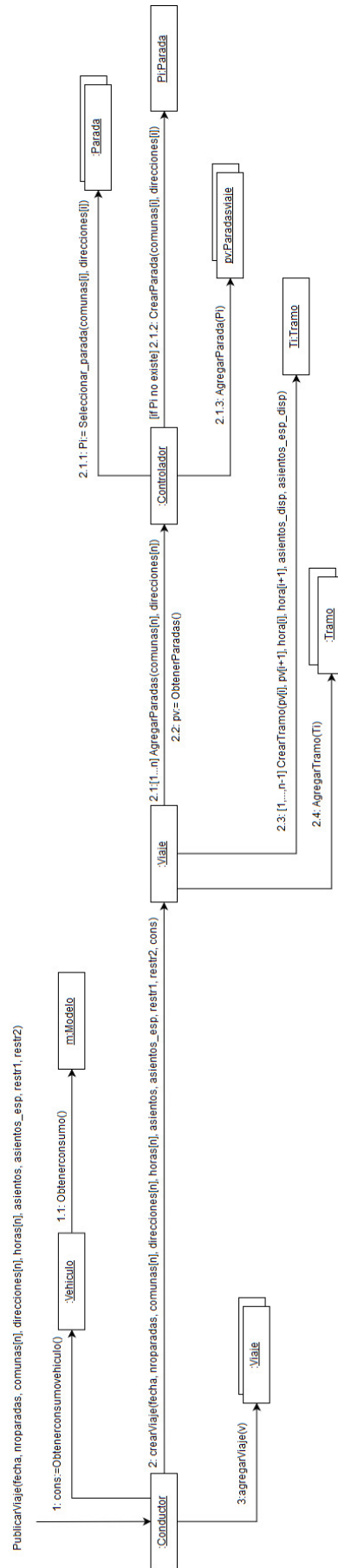


Figura 4: Diagrama de comunicación Publicar Viaje.

4.2. Diagrama de clases:

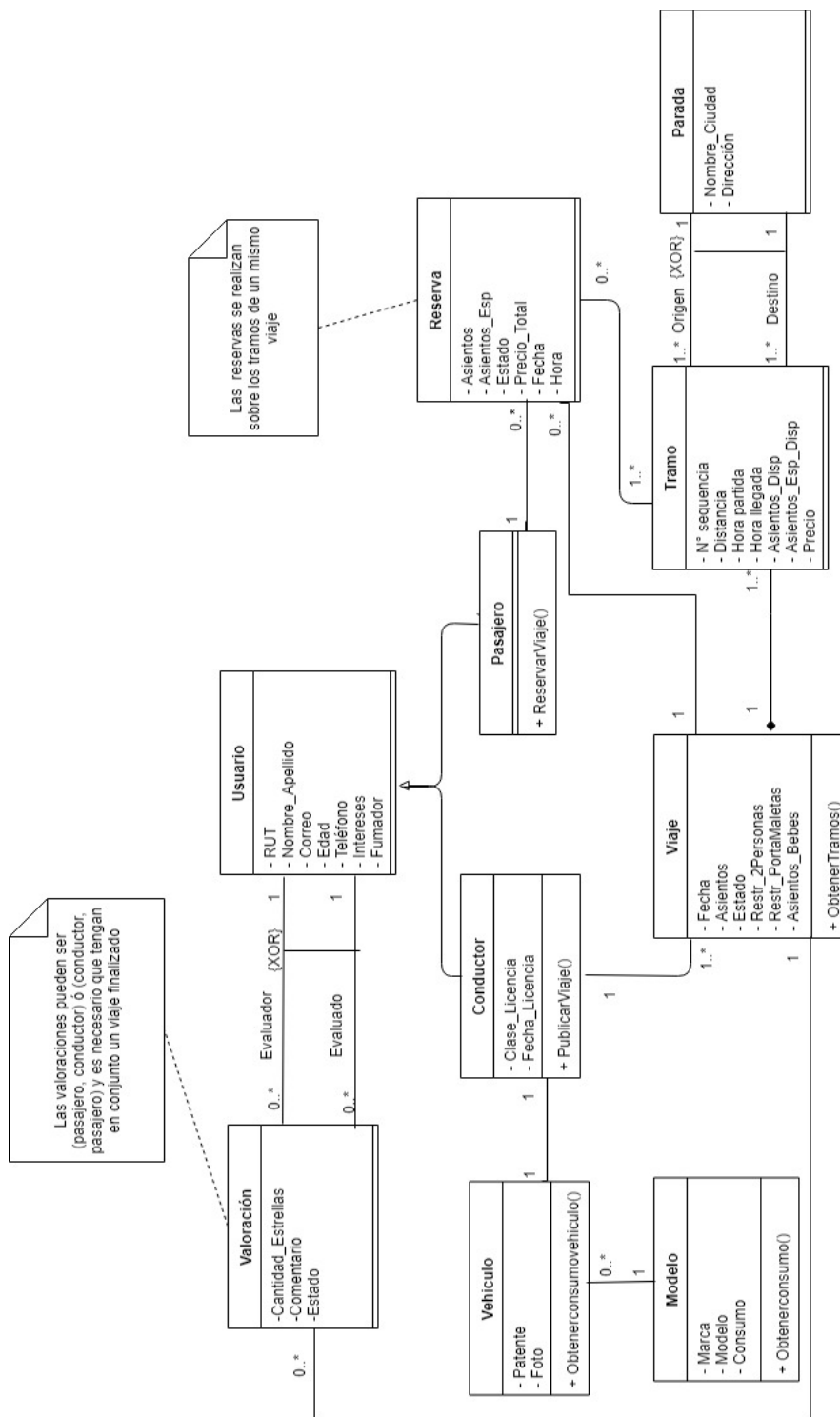


Figura 5: Diagrama de clases con métodos de los diagramas de comunicación.

5. Conclusión

En nuestra segunda iteración corregimos y cambiamos casos de uso como valorar pasajero por valorar usuario, además de eliminar el sistema verificador, para simplicidad de implementación.

Con respecto a la implementación de los casos de uso, cada uno esta implementado completamente cumpliendo con el objetivo del proyecto semestral.

Nuestro principal logro fue que a través de una lista de requerimientos pudimos construir, diseñar e implementar un sistema de software, logrando el objetivo del proyecto semestral y finalmente entendimos la importancia de cada una de las etapa para llegar a implementar de una manera eficiente y eficaz.

Nuestra dificultad fue que no se escogieron las herramientas adecuadas para la implementación del software, por otra parte, elegimos un camino difícil en el modelo para implementar la API de Google Maps.

6. Glosario:

Prestaciones adicionales: significa que el conductor ofrece para un pasajero preferencias como, silla para bebés, un máximo de plazas para un viaje y equipaje en el portamaletas.

Tramo: trayecto entre paradas consecutivas.