

María Alejandra Pinzón

Gabriela Escobar

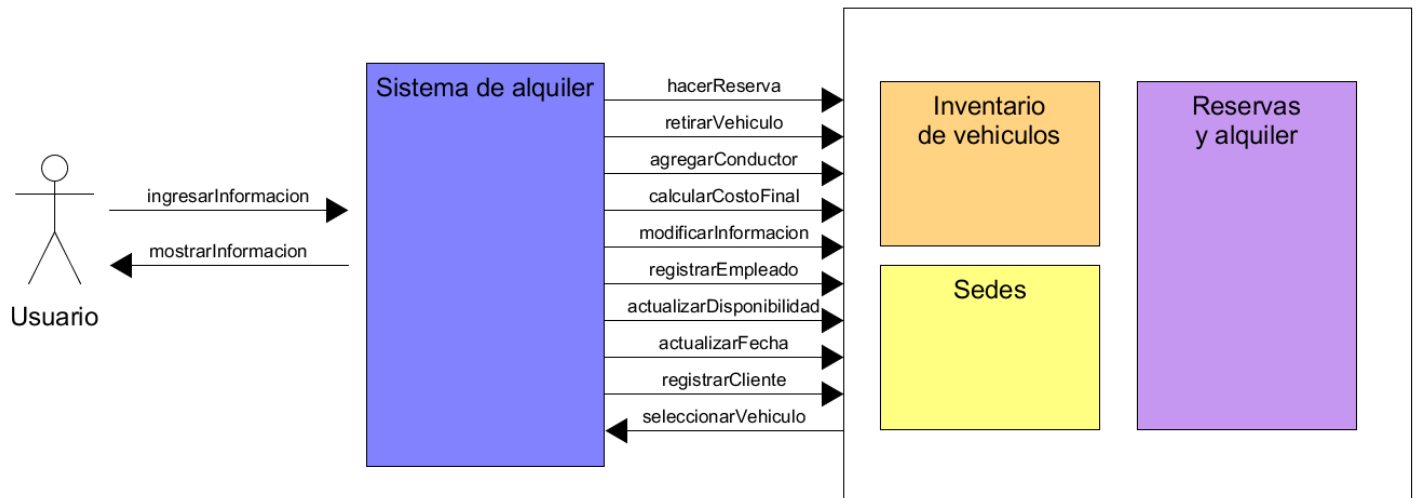
Mateo López

Diseño – Alquiler de vehículos

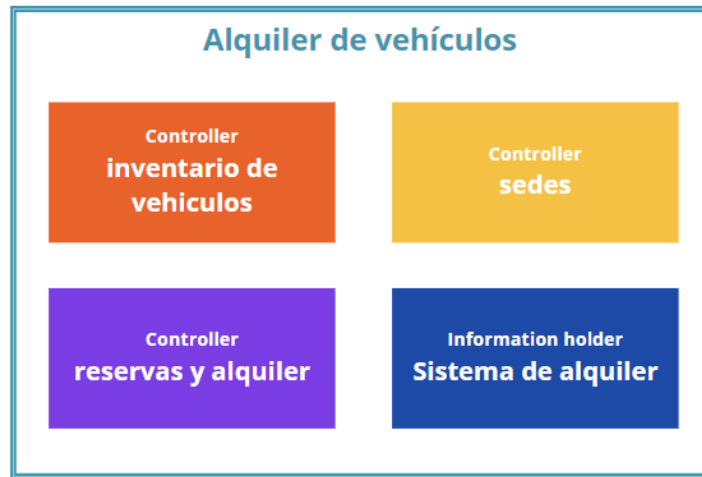
NIVEL 1

1. Contexto del problema

Para este proceso de diseño, es esencial definir las funcionalidades de alto nivel que el alquiler de vehículos debe satisfacer, especialmente en su interacción de los usuarios con la interfaz que en este caso la nombramos “Sistema de alquiler”. El sistema de alquiler se encargará de interactuar con los otros componentes: Inventario de vehículos, Sedes y Reservas y alquiler. Estos componentes tendrán la función de manejar la lógica de todo el sistema.



2. Componentes candidatos y estereotipos- primera iteración



#	Responsabilidad	Componente
1	Gestionar la información de un vehículo	Inventario de vehículos
2	Gestionar el estado de un vehículo (disponible, alquilado por alguien, en limpieza, etc.)	
3	Gestionar la información de las categorías de los carros	
4	Gestionar una reserva	Reservas y alquiler
5	Gestionar entrega y devolución de un vehículo	
6	Gestionar la información básica de las sedes (nombre, ubicación y horarios de atención)	Sistema de sedes
7	Gestionar la información de usuarios (clientes, empleados y administrador)	
8	Gestionar la información de los vehículos de la sede.	
9	Manejar un usuario para cada persona	Sistema de alquiler
10	Mostrar al usuario información sobre el proceso de reserva	

Diagrama Diseño UML de alto nivel (nivel inicial de descomposición)

Clases de proyecto divididos por componentes **esto sirve como uml de alto nivel?**

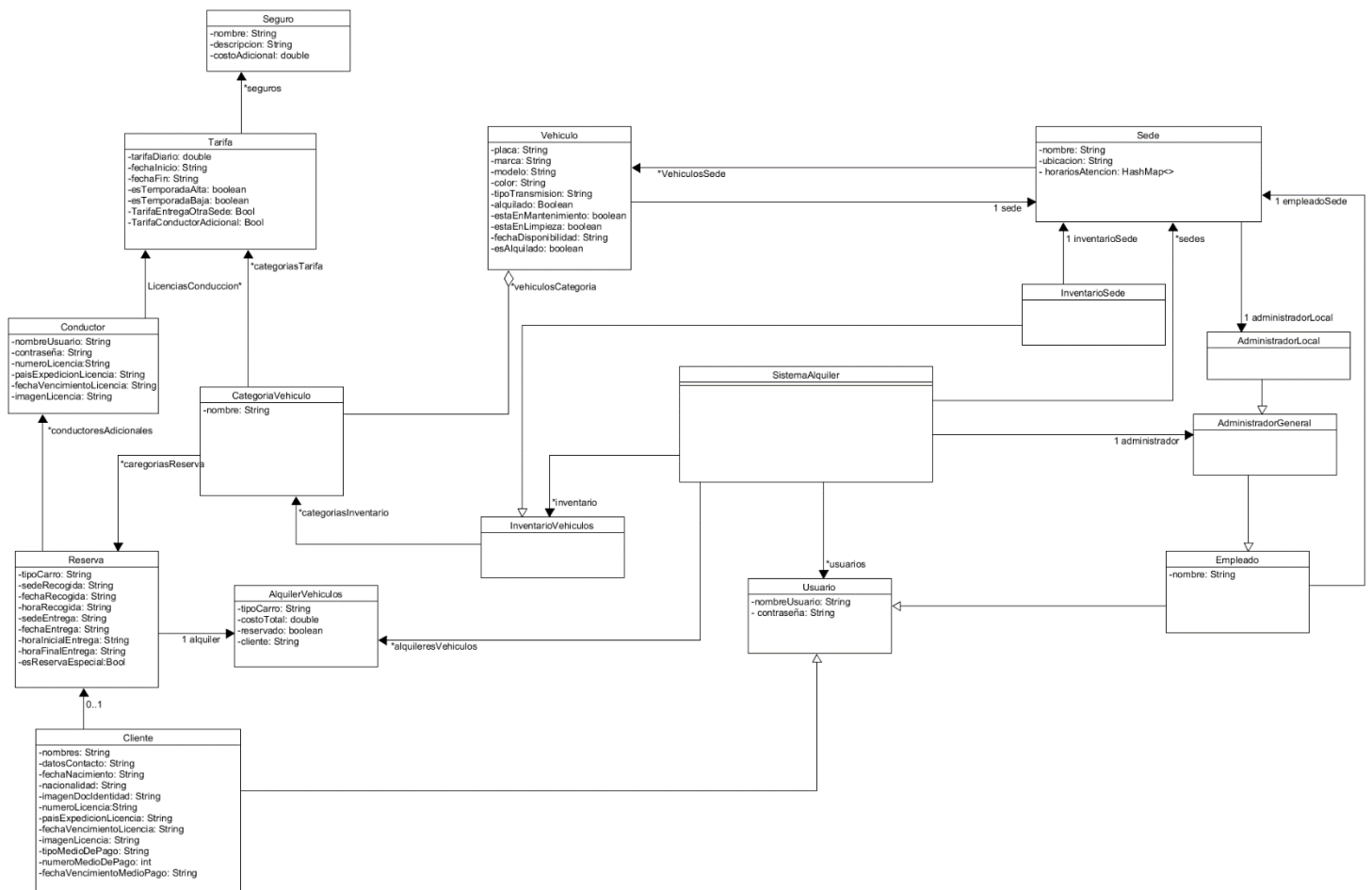


Figura 1: UML alto nivel – nivel 1

Justificación de cambios con respecto al diagrama de clases del análisis:

En esta primera iteración de diseño, definir los componentes principales del proyecto y sus responsabilidades fue fundamental para hacer los cambios con respecto al diagrama de análisis. Al definir estos, nos dimos cuenta de existen algunos comportamientos o responsabilidades en común de ciertas clases, de relaciones que no habíamos especificado anteriormente, de errores en la lógica del diagrama anterior, y, en general, empezamos a estructurar concretamente cómo íbamos a construir nuestro proyecto. A continuación, daremos algunos ejemplos de estos cambios.

Con respecto a que existían algunas clases con comportamientos en común, un ejemplo de esto pueden ser las clases de InventarioVehículos e InventarioSede. De hecho, InventarioSede no existía en el diagrama de diseño, pero considerando que no habíamos tomado en cuenta el hecho de que era importante saber el inventario, no solo de todo el negocio, pero de cada sede específica, se decidió crear esta clase. Ambos son inventarios, la diferencia es que InventarioVehículos es el inventario de todo el negocio, entonces cuando se decide comprar un carro nuevo, se añade en este, o si se decide retirar un carro del negocio, se hace en este inventario; por otro lugar, el InventarioSede es fundamental para saber cuáles son los carros disponibles en cada momento en una sede en específico, lo cual sirve para seleccionar los carros que se rentarán. Así, al ambos ser inventarios,

los vehículos están clasificados en las mismas categorías, y los carros tienen las mismas características. Es por lo anterior que decidí hacer una relación de herencia entre estas dos clases, pues ambas tienen las mismas características, pero difieren en la información y la cantidad de esta que muestran.

Otro cambio que hicimos fue con respecto a las relaciones, pues en el análisis no habíamos explicitado algunas o nos equivocamos en el planteamiento de otras. Una relación que no habíamos explicitado anteriormente y que es fundamental para el desarrollo del proyecto es la relación entre conductor y reserva. Cada reserva debe tener la información del conductor que va a ser quien utilizará el servicio brindado por la empresa. Un error que notamos de las relaciones que habíamos planteado anteriormente es la de CategoríaVehículo y Vehículo, pues en el diagrama de análisis la relación de composición había sido planteada del otro sentido.

Un ejemplo que puede explicar lo que queremos decir cuando mencionamos errores en la lógica, es la lógica de las clases: Usuario, Empleado, AdministradorGeneral y AdministradorLocal. En el diagrama de análisis, esta sección de “actores” era distinta, pues había clases adicionales y las relaciones eran distintas. Mediante la discusión entre nosotros y revisando ejemplos reales, nos dimos cuenta de que la lógica planteada en la solución propuesta en el análisis era redundante y existían clases que no se iban a usar en ningún momento porque no hacían parte de la solución que queríamos plantear ahora, en este primer nivel de diseño. Al entender esto, decidimos replantear esta parte del diagrama.

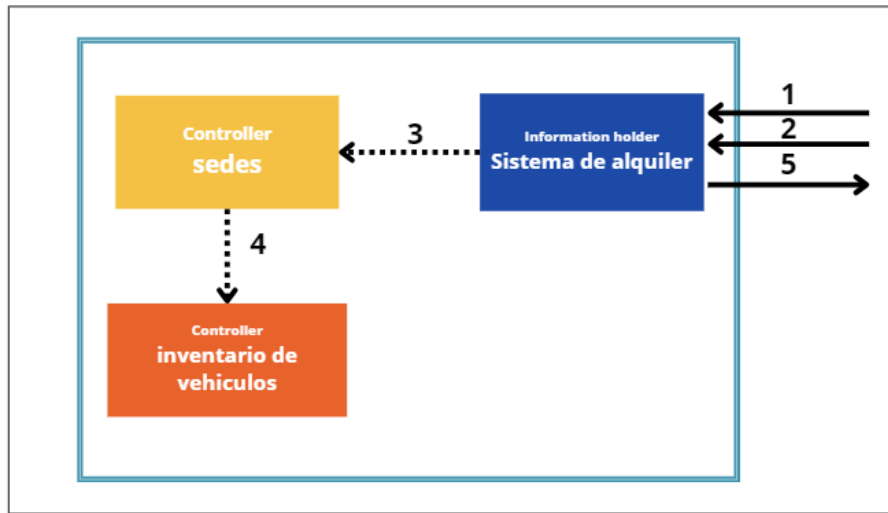
Finalmente, lo que se mencionamos anteriormente de la estructuración un poco más concreta del proyecto, se debe principalmente a la división de componentes. En el diagrama de análisis, la idea del proyecto puede que sea la misma a la creada en este primer nivel de diseño, pero en esta versión más actualizada, hemos tenido que pensar en cómo tenemos que lograr cumplir las responsabilidades, los requerimientos funcionales. Esta organización permite que se pueda asignar a cada una de las clases uno de los componentes principales que definimos.

NIVEL 2

1. Relaciones entre componentes de primera iteración

1- Gestionar el estado de un vehículo:

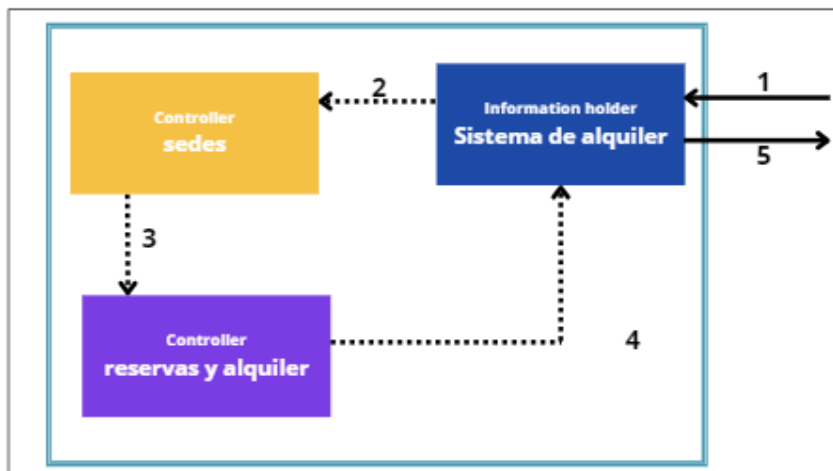
Descripción: Un administrador actualiza el estado de un vehículo (disponible, alquilado por alguien, en limpieza, etc.)



- 1) El administrador local de una sede ingresa sus datos para poder modificar la información
- 2) El administrador local de la sede indica el vehículo que quiere modificar y la información del estado del vehículo actualizada
- 3) El inventario de la sede a la que pertenece el vehículo se actualiza.
- 4) Con el inventario de la sede, inventario general de vehículos se actualiza.
- 5) Mensaje de actualización exitosa

2- Hacer una reserva.

Descripción: un cliente entra al sistema de la empresa para hacer la reserva de alquiler de un carro.



- 1) El usuario interactúa con el sistema, donde inicia sesión y da la información básica del carro que desea alquilar (categoría, en qué sede quiere recoger el carro, etc.)

- 2) El sistema de alquiler conecta la solicitud del usuario con la sede en la que quiere recoger el carro. En la sede verifican la disponibilidad de autos de la categoría especificada por el usuario.
- 3) Envía la información del usuario y del auto que se va a alquilar a Reservas y alquiler, donde se guarda la información de la reserva.
- 4) Una vez hecha la reserva, la información de esta se regresa al sistema de alquiler. Esta información contiene el estado de la reserva: si se puede alquilar, y si no la razón de esto.
- 5) El sistema de alquiler le entrega la información al cliente.

NIVEL 3.

1. Revisión de componentes- segunda iteración

a. Descomposición de inventario de vehículos

#	Responsabilidad	Componente
1.	Retirar un vehículo del catálogo del inventario.	Inventario de vehículos
2.	Registrar un nuevo vehículo en el catálogo.	
3.	Gestionar las categorías de los carros	
4.	Modificar la disponibilidad de un vehículo por alquiler	
5	Modificar el estado de un vehículo por limpieza o mantenimiento	

b. Descomposición de reservas y alquiler

#	Responsabilidad	Componente
6.	Crear una reserva	Reservas y alquiler
8.	Configurar la información de los seguros	
9.	Gestionar entrega y devolución de un vehículo	
10.	Registro de conductores adicionales	
11.	Calcular el costo total en reserva de seguros	
12.	Calcular el costo final de una tarifa	

c. Descomposición de sistema de sedes

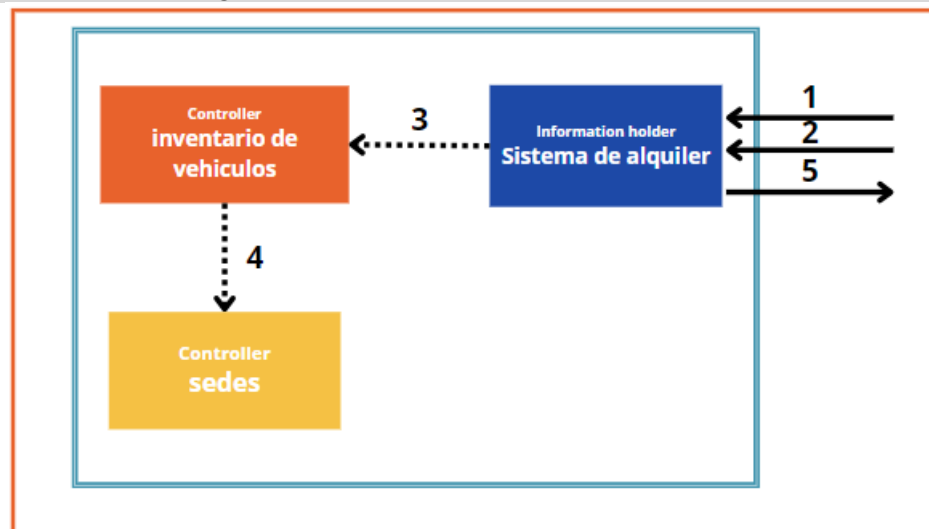
#	Responsabilidad	Componente
13.	Registrar información de empleados de la sede	Sistema de sedes
14.	Manejar información de empleados de la sede	
15.	Actualizar estado de los carros de sede (cuando el carro es devuelto después de ser limpiado antes de que vuelva a estar disponible)	
16.	Sacar un carro del inventario disponible si este necesita mantenimiento.	

d. Descomposición de sistema de alquiler

#	Responsabilidad	Componente
19.	Registrarse como cliente en el sistema de alquiler.	Sistema de alquiler
20.	Iniciar sesión en el sistema de alquiler.	
21.	Mostrar categorías disponibles para reserva y alquiler en la ubicación del usuario	
23.	Mostrar el vehículo asignado para el cliente	

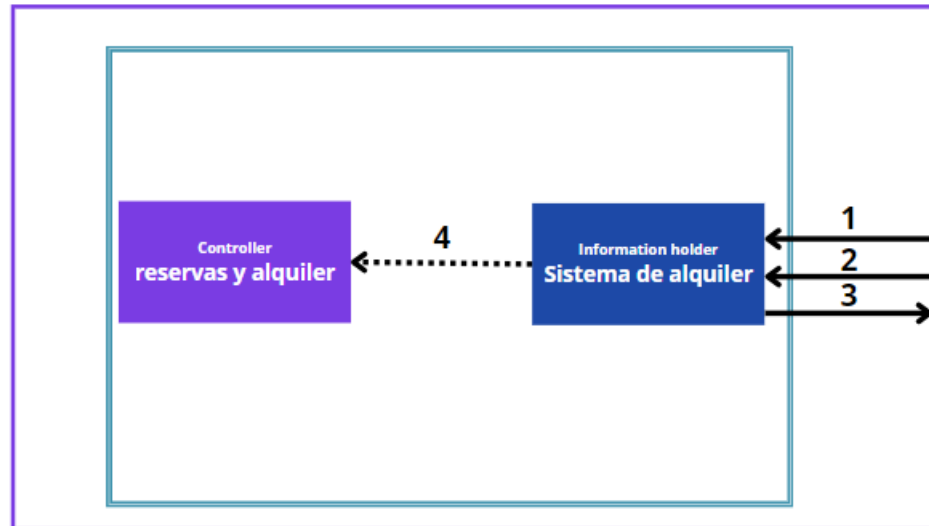
2. Relación entre componentes

1- Retirar un vehículo del catálogo del inventario.



- 1) El administrador accede al sistema y selecciona el vehículo que se debe retirar del catálogo.
- 2) El administrador selecciona el estado del vehículo como "retirado" o similar.
- 3) El sistema de alquiler conecta la solicitud del administrador con el inventario indicando el vehículo que quiere modificar y el inventario de vehículos actualiza la disponibilidad del vehículo a "no disponible" para futuras reservas.
- 4) Con la información actualizada en el inventario general del sistema de alquiler, el inventario de la sede a la que pertenece ese vehículo se actualiza también.
- 5) Mensaje de confirmación de retiro exitoso

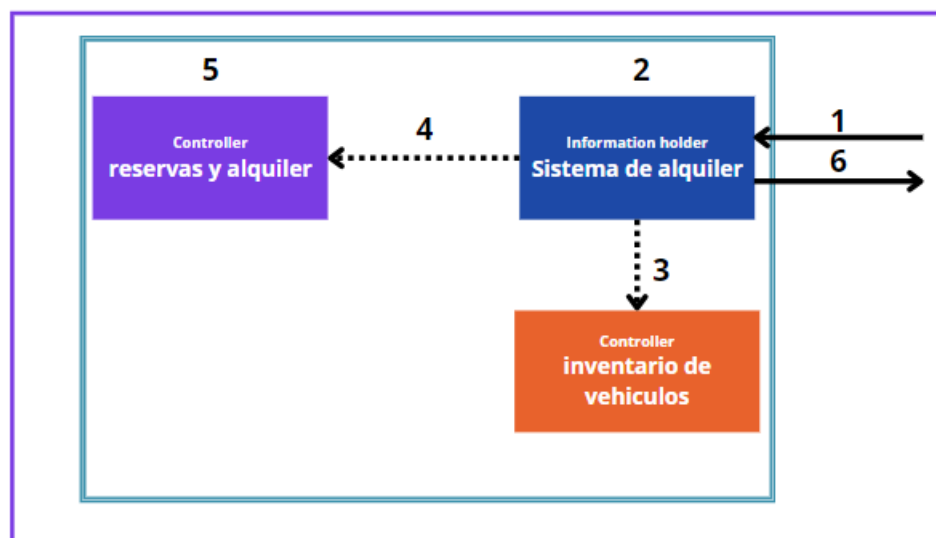
2- Registro de conductores adicionales



Teniendo en cuenta que el cliente ha indicado que desea incluir uno o más conductores adicionales:

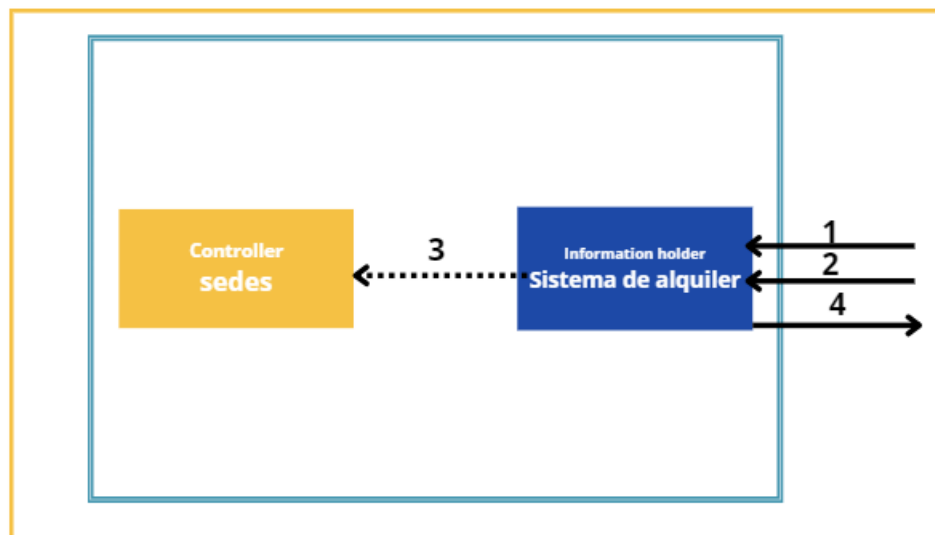
- 1) Un empleado de la empresa accede al sistema y selecciona la reserva del cliente que va a añadir conductores adicionales
- 2) El empleado brinda los conductores adicionales uno por uno al sistema con su respectiva información
- 3) El sistema valida la información de los conductores adicionales asociados a la reserva y muestra un mensaje de confirmación
- 4) El sistema de alquiler conecta la solicitud del empleado con reservas y alquiler agregando cada conductor a la reserva del cliente.

3- Calcular el costo final de una tarifa



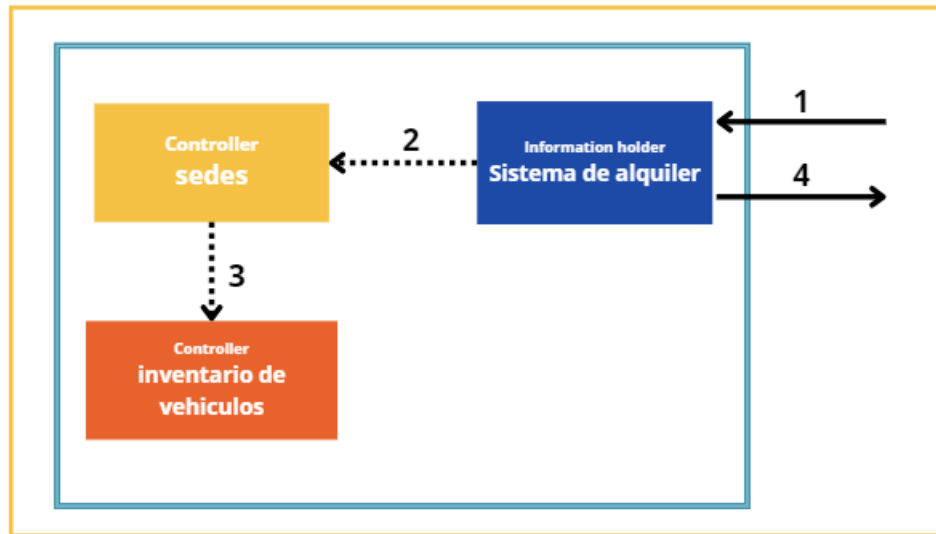
- 1) Un usuario (cliente o empleado) solicita al sistema de alquiler calcular el costo final para un alquiler específico.
- 2) El sistema de alquiler obtiene los detalles de la reserva, como el tipo de vehículo, la cantidad de días, la cantidad de conductores adicionales, y si es una reserva especial o no.
- 3) El sistema de alquiler consulta al "Inventario de vehículos" para obtener la tarifa base o precio diario del vehículo seleccionado basado en su categoría.
- 4) El sistema de alquiler conecta la información del vehículo de la reserva con el componente de reservas y alquiler
- 5) Dentro del componente Reservas y Alquiler, se aplican los modificadores correspondientes a la tarifa base. Esto incluye ajustes por duración del alquiler, conductores adicionales, seguros, entre otros. Y se realiza el cálculo final sumando la tarifa base y todos los modificadores aplicados.
- 6) El sistema de alquiler le muestra al usuario el costo para un alquiler específico.

4- Registrar la información de los empleados de la sede



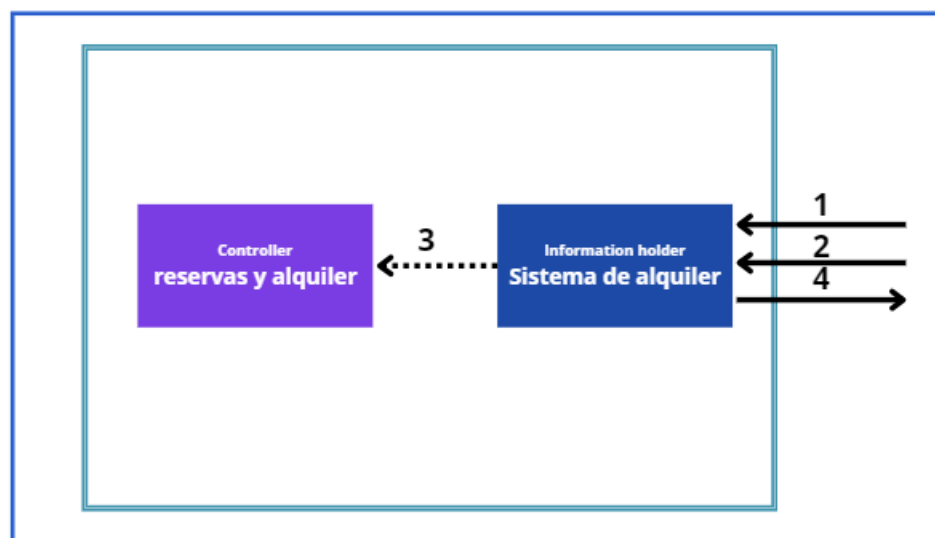
- 1) El administrador local de la sede ingresa al sistema de alquiler y selecciona la opción de registrar un nuevo empleado
- 2) El administrador local ingresa la información del empleado
- 3) El sistema de alquiler conecta la solicitud del administrador local con el componente de sedes y este guarda la información del empleado
- 4) Se muestra un mensaje de confirmación

5- Actualizar el estado disponible si este necesita mantenimiento



- 1) El empleado de la sede ingresa al sistema con sus credenciales y selecciona el vehículo para el cual quiere actualizar su disponibilidad
- 2) El sistema de alquiler conecta la solicitud del empleado con el inventario de la sede respectiva y actualiza los datos relacionados con disponibilidad y mantenimiento del vehículo
- 3) El inventario de la sede brinda los datos actualizados al inventario general de vehículos
- 4) Se muestra un mensaje de confirmación

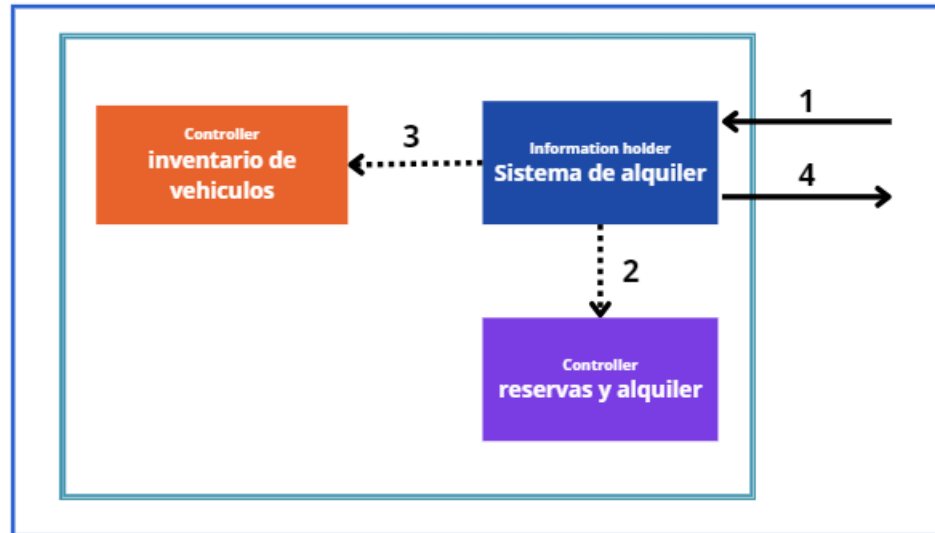
5) Registrarse como cliente en el sistema de alquiler



- 1) El cliente selecciona la opción registrarse en el sistema de alquiler
- 2) El cliente ingresa los datos solicitados para registrarse

- 3) El sistema de alquiler guarda los datos del en el componente de reservas y alquiler.
- 4) Se muestra un mensaje de confirmación de registro.

6) Mostrar el vehículo asignado para el cliente



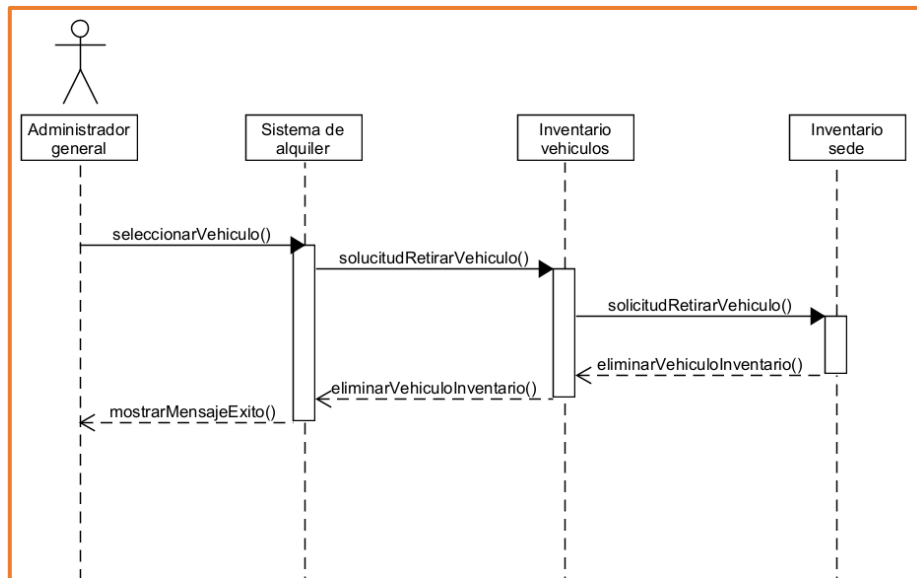
- 1) El cliente ingresa los datos de su reserva para hacer efectivo el alquiler
- 2) El sistema de alquiler recolecta todas las especificaciones de la reserva conectándose con el componente de reservas y alquiler
- 3) El sistema de alquiler utiliza de esos datos la categoría del vehículo que selecciono el cliente y lo conecta con el inventario de vehículos general para identificar si hay disponibilidad en esta categoría y en caso de que si se seleccionara un vehículo al azar. En el caso contrario se seleccionará un vehículo al azar de una mejor categoría.
- 4) Se muestra un mensaje con el vehículo y las características de este.

NIVEL 4

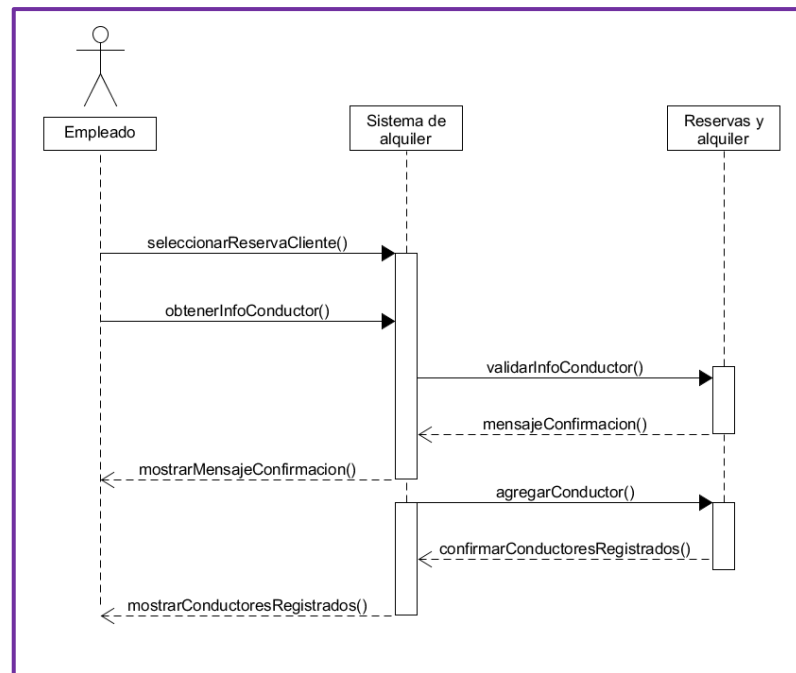
Descomposición final

1. Diagramas de secuencia de algunas de las responsabilidades más importantes

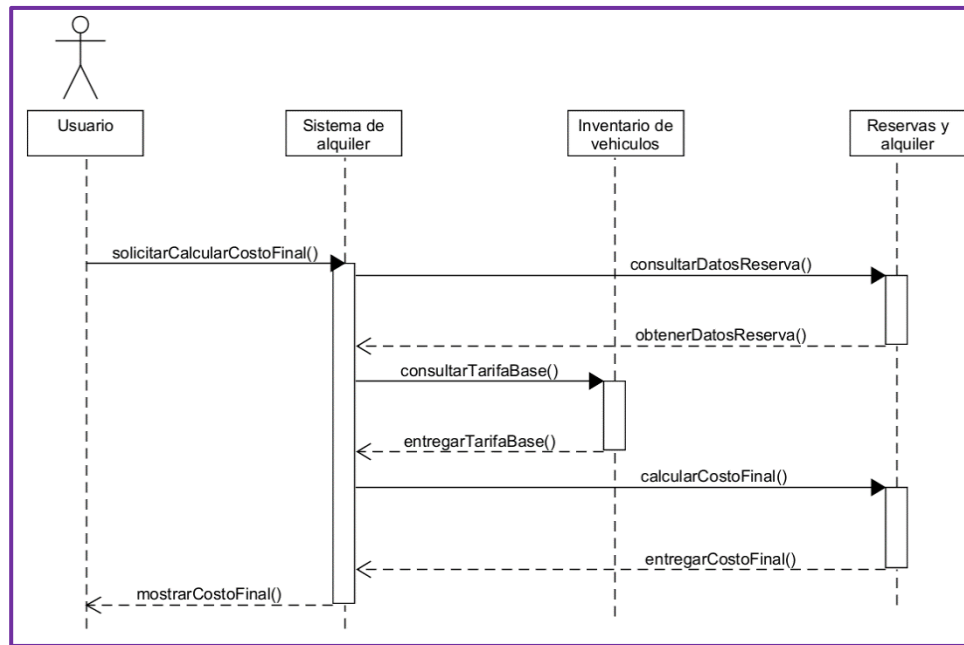
1- Retirar un vehículo del catálogo del inventario.



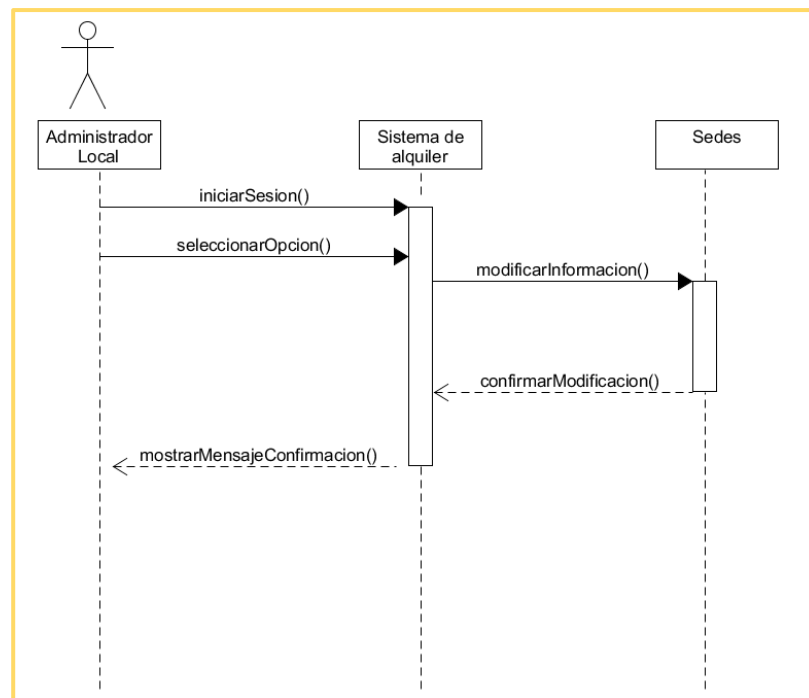
2- Registro de conductores adicionales



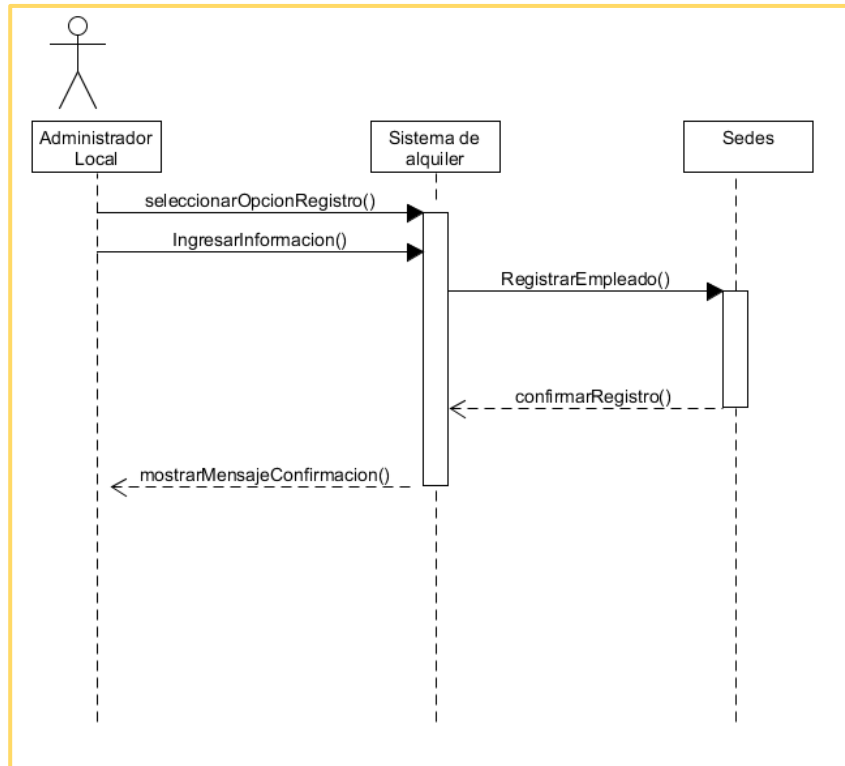
3- Calcular el costo final de una tarifa



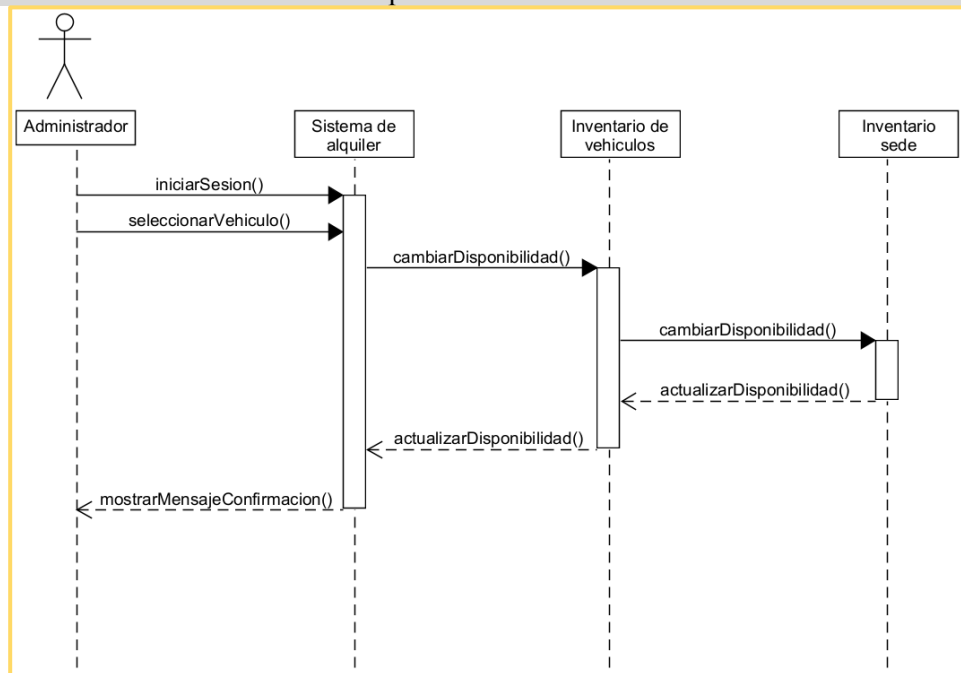
4- Gestionar la información básica de las sedes (nombre, ubicación y horarios de atención)



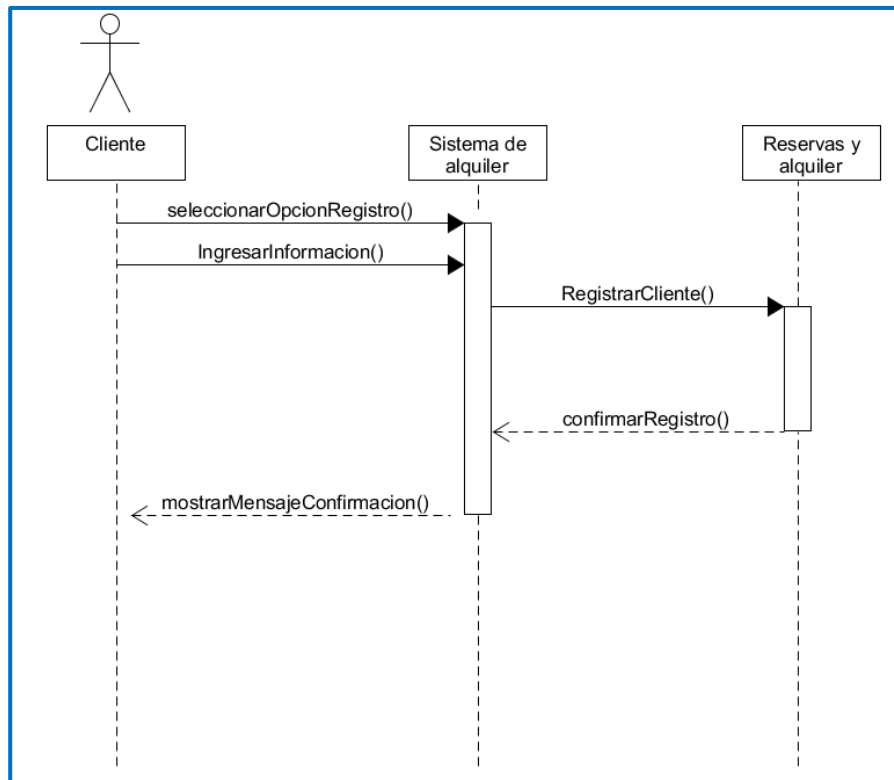
5- Registrar la información de los empleados de la sede



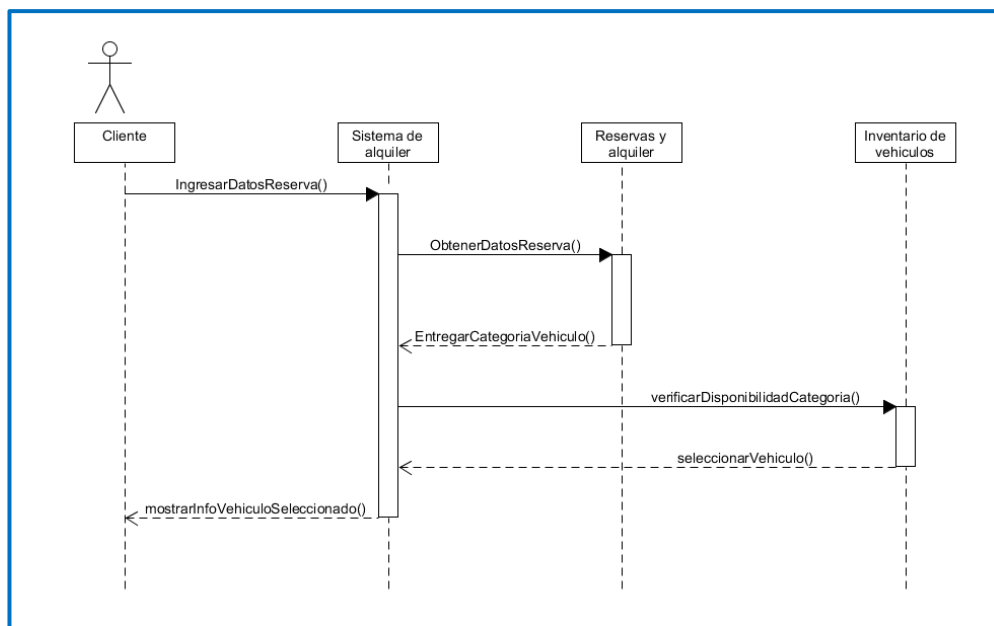
6- Sacar un carro del inventario disponible si este necesita mantenimiento



7- Registrarse como cliente en el sistema de alquiler



8- Mostrar el vehículo asignado para el cliente



2. Diagrama de clases de alto nivel, que incluya todas las clases y sus relaciones, pero no todos los métodos ni atributos.

Imagen adjuntada en pdf en la entrega.

Figura 2: UML alto nivel – nivel 4

3. Diagrama de clases de diseño que incluya todas las clases, incluyendo sus las relaciones, atributos y métodos.

Imagen adjuntada en un pdf en la entrega.

Figura 3: UML detallado – nivel 4

JUSTIFICACIONES DE LOS CAMBIOS:

- Se eliminó la clase Sede, ya que era irrelevante realizar procesos sobre alguna sede singular, y se prefirió realizar todos los cambios en el inventario general y luego enviarlos a las sedes individuales sin la necesidad de esta clase.
- Se creó una interfaz llamada “Tarifable” la cual permitía organizar los métodos de la clase de Reserva y la de AlquilerVehiculo, ya que estas son suficientemente similares.
- La clase CategoriaVehiculo no se utiliza, tampoco la clase Tiempo. Estas no fueron necesarias para la terminación del proyecto, ya que se utilizó el dato primitivo int para guardar las categorías, y String para guardar el tiempo.