

Proyecto 1- Documento de diseño Renta de Vehículos

1. Contexto del problema

En el siguiente diagrama, se muestran las diferentes funcionalidades que la aplicación debe satisfacer cuando un tipo de usuario interactúe con la interfaz. Lo hacemos para que se pueda ver independiente de la implementación en código, se puedan ver las diferentes interacciones que puede tener un individuo usuario con la aplicación (según su rol en la empresa: cliente, empleado, gerente global, o gerente local).

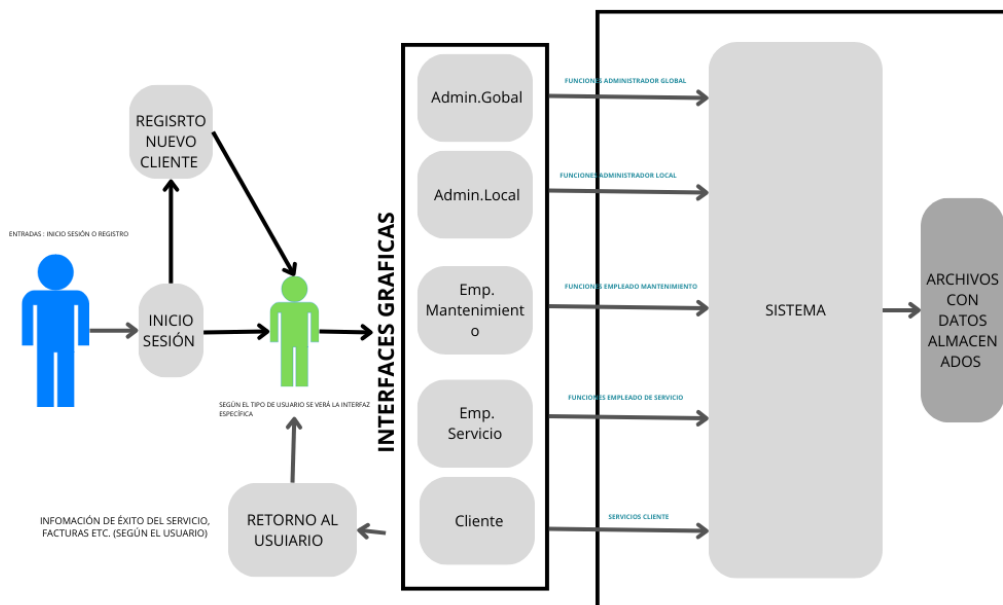


Figura1. Definición del contexto del problema

Según el tipo de usuario que debe estar registrado en el usuario, la aplicación debe dar cierto acceso a la persona de acceder a ciertos servicios. Por ejemplo, estos son algunas de las entradas que puede tener un usuario de cada tipo:

Usuario:

- Login y password

- a. Administrador global:
 - a. Registrar un vehículo nuevo
 - b. Eliminar un vehículo
 - c. Configurar seguros a ofrecer
 - d. Maneja la información de las sedes y los empleados
- b. Administrador Local:
 - a. Crear usuarios de empleados
 - b. Acceder y modificar información de empleados
- c. Empleado de Mantenimiento:
 - a. Limpiar carros
 - b. Actualizar el estado de cada carro
 - c. Indicar si necesita mantenimiento (sacar del inventario, calcular tiempo en el que no va a estar disponible)
- d. Empleado de Servicio:
 - a. Cerrar alquileres
 - b. Entrega y devolución de carros
 - c. Ofrecer seguros
 - d. Preguntar por conductores extra
 - i. Registrar los conductores extra
- e. Cliente:
 - a. Hacer reserva: solo un cliente puede hacerlo y esto implica registrar sus datos personales, datos de licencia de conducción y datos de medio de pago
 - b. Datos para una reserva:
 - Tipo de carro
 - Sede
 - Fecha y hora (recogerlo, como cuando lo vaya a devolver)
 - Donde se entrega
 - Rango de horas
 - Categoría

(estos datos están sujetos a la hora de realizar el alquiler)

 - i. Indicar fechas de entrega

2. Contexto del problema

2.1. Componentes candidatos y estereotipos

1. Considerando la necesidad de persistencia en la información con respecto a los usuarios registrados, es necesario tener un ***Information Holder*** para la **lista de usuarios** y además su información. Cabe resaltar que los clientes también cuentan como usuarios debido a que pueden crear y gestionar sus propias reservas por ello deben estar almacenados de esta manera.
2. Considerando la importancia del estado de los vehículos y la disponibilidad, también es esencial tener un archivo ***Information Holder*** de **vehículos**, que se encargue de registrar tanto la información de los vehículos, como su estado. Es decir, si un vehículo es apto para ser rentado.
3. Por otro lado, para la operabilidad del sistema, es necesario utilizar un **Controller y además un *Information Holder* para Cliente** donde desde su interfaz, el usuario pueda hacer reservas, alquilar, calcular tarifas.
4. En el sector administrativo, también es importante tener un sistema que opere según las responsabilidades de cada administrador con un **Controller para Administradores y además un *Information Holder*** (donde está la información a la cual quiere acceder, agregar o modificar el administrador). Este sistema se distingue del de operabilidad de usuario en que no interactúa con clientes de ninguna manera, sino que es donde los usuarios administrativos pueden acceder a la información de clientes, inventario, diferentes usuarios registrados e información de estos.
5. Para ser más específicos necesitamos tener un sistema que opere según las responsabilidades del administrador global con un **Controller para administrador global y además un *Information Holder*** (donde está la información a la cual quiere acceder, agregar o modificar el administrador). Este sistema se distingue del de operabilidad de usuario en que no interactúa con clientes de ninguna manera, sino que es donde este usuario administrativo puede registrar un vehículo nuevo, eliminar un vehículo, configurar los seguros, (además este tiene acceso a la información de sedes y empleados).
6. En este caso contamos con un segundo administrador que es de vital importancia para el programa y por ello debe tener un sistema que opere según las responsabilidades del administrador local con un **Controller para administrador local y además un *Information Holder***. Este sistema se distingue del de operabilidad de usuario en que no interactúa con clientes de ninguna manera, sino que es donde este usuario administrativo puede crear usuarios, acceder y modificar información de empleados de la sede para la cual está a cargo dicho administrador y empleado.
7. La empresa también cuenta con empleados que ayudan con todo el tema relacionado a la renta de carros y además son de vital importancia y es por ello por lo que debe tener un sistema que opere según las responsabilidades de cada empleado es este caso con un **Controller para empleados de mantenimiento y además un *Information Holder***. Este sistema esta a cargo de actualizar el estado de un carro, indicar si necesita mantenimiento y calcular el tiempo en el cual este no estará disponible.

8. Por otro lado la empresa también cuenta con un tipo más de empleados que ayudan con todo el tema relacionado a la renta de carros y además son de vital importancia y es por ello por lo que debe tener un sistema que opere según las responsabilidades de cada empleado es este caso con un **Controller para empleados de servicio y además un Information Holder**. Este sistema está a cargo del cierre de alquiler, entrega, devolución, ofrecer seguros, asegurar cuantos conductores extra va a incluir el cliente de cada vehículo.

9. Finalmente optamos por dejar la información de empleados de servicio y mantenimiento en un solo archivo (**Information Holder**), pero cada uno tiene acciones diferentes que cumplen diferentes requerimientos funcionales como se especificó anteriormente.

10. Considerando que la **tarifa** para el alquiler de un vehículo es muy importante decidimos guardar esta información en un (**Information Holder**). Este sistema se distingue del de operabilidad de usuario en que no interactúa con este de ninguna manera, sino que es donde se puede acceder a la tarifa dependiendo de la categoría, tarifa diaria, temporada alta o baja, y dependiendo de la sede.

11. La empresa no solo tiene una tarifa distinta dependiendo de las condiciones que acabo de mencionar anteriormente, también la tarifa varía de acuerdo con unos **seguros** que la empresa ofrece y por eso decidimos guardar esta información para más adelante hacer uso de ella en un **Information Holder**. Este sistema se distingue del de operabilidad de usuario en que no interactúa con este de ninguna manera, sino que es donde se puede acceder a los seguros adicionales donde cada uno de ellos tiene un costo adicional por día.

12. Otra de las funcionalidades cruciales del sistema tiene que ver con la interacción entre un usuario y las interfaces gráficas. Uno de los requisitos funcionales para este sistema es que, según el tipo de usuario con el que se está trabajando, la aplicación muestre las opciones o acciones que puede ejecutar el sistema para ese usuario. Por eso el **filtro de interfaces** es uno de los componentes fundamentales. Además, también es el lugar idóneo para cargar los datos cada vez que se inicie la aplicación.

RESERVA DE CARROS



2.2. Responsabilidades

La siguiente tabla, le va a permitir al lector ver las responsabilidades que identificamos que cada uno de estos componentes debe cumplir para el correcto desempeño de la aplicación.

Numero	Responsabilidad	Componente
1	Registrar a cada usuario del sistema con un login y password	Usuarios
2	Conocer la información de los usuarios registrados	Usuarios
3	Poder guardar la información de los usuarios que no aún no están registrados	Usuarios
4	<u>Justificación:</u> decidimos que usuarios va a contener la información de los usuarios registrados en el sistema, esto incluye administradores, empleados y clientes por ello solo tiene estas responsabilidades.	Usuarios
4	Verificar y registrar la disponibilidad de vehículos en cada una de las sedes de la compañía	Vehículo
5	Registrar estado del vehículo	Vehículo
6	Clasificación por categorías	Vehículo
7	<u>Justificación:</u> Este es básicamente el “inventario”, pero decidimos mejor guardar la información de cada vehículo (la placa es el identificador de cada vehículo) y con ello sabemos en que condiciones se encuentra, si este se encuentra registrado en alguna sede, y a que categoría corresponde	Vehículo
8	Realizar reservas	Cliente
9	Alquilar vehículo	Cliente
10	Calcular tarifas	Cliente
11	Registrarse como usuarios nuevos	Cliente
12	Adquirir seguros adicionales	Cliente
13	Indicar conductores extras	Cliente
14	Acceder a la información de los usuarios	Administrador Global
15	Acceder al inventario	Administrador Global
16	Acceso a información de sedes y empleados	Administrador Global
17	Es quien se encarga de modificar la tarifa	Administrador Global
18	Es quien se encarga de modificar y agregar los seguros	Administrador Global
19	Crear usuarios nuevos	Administrador local
18	Modificar la información de empleados de la sede correspondiente	Administrador local
	Limpiar vehículo	Empleados mantenimiento

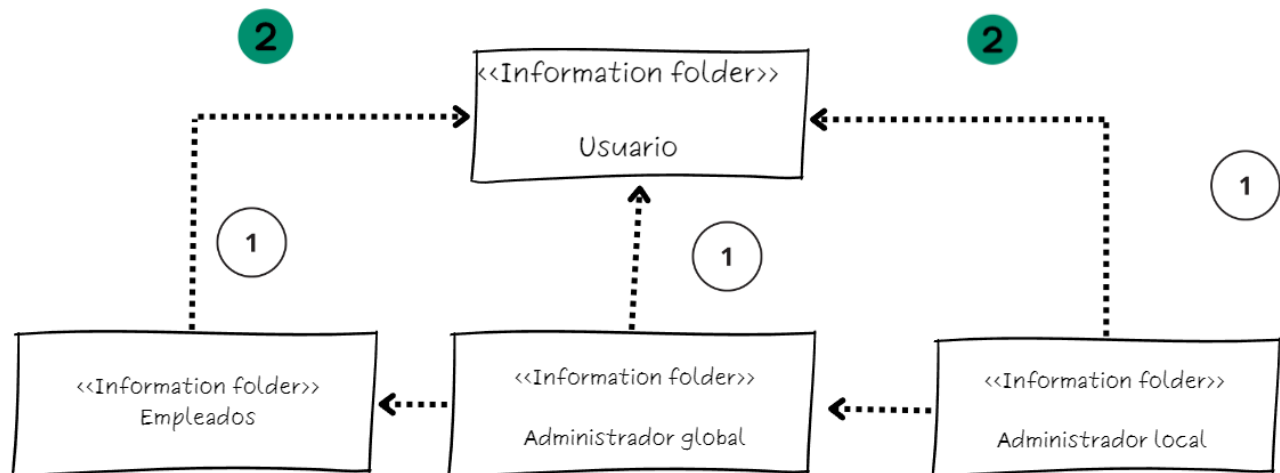
19	Actualizar el estado de un carro	Empleados mantenimiento
20	Indicar si necesita mantenimiento	Empleado mantenimiento
21	Calcular el tiempo en cual un vehículo vuelve a estar disponible	Empleado mantenimiento
22	Encargado del cierre del alquiler	Empleado servicio
23	Entrega de vehículo	Empleado servicio
24	Devolución de vehículo	Empleado servicio
25	Ofrecer seguros a los clientes	Empleado servicio
26	Asegurarse de cuantos conductores extra va a incluir el cliente de cada vehículo.	Empleado servicio

Tabla 1: Asignación de responsabilidades

2.3. Colaboraciones

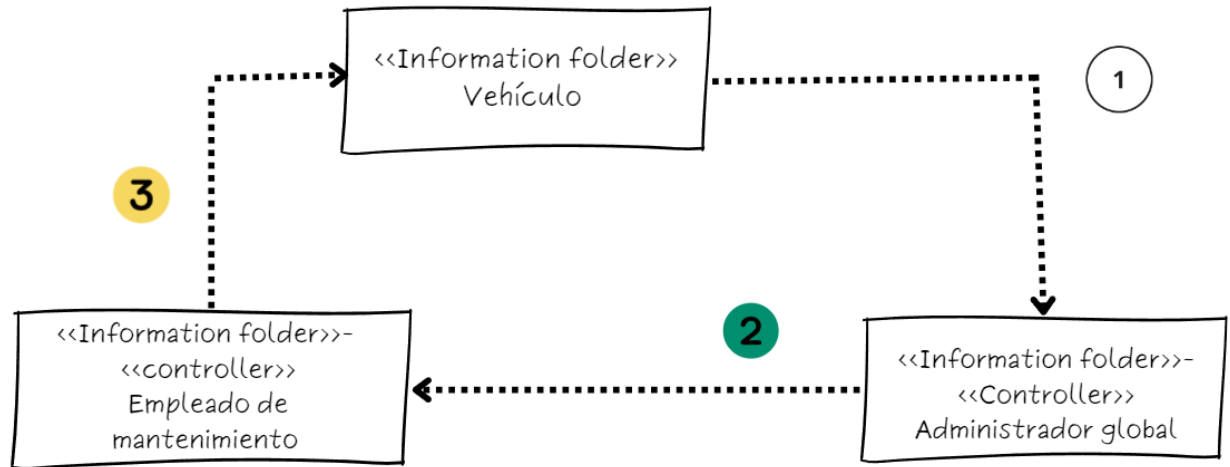
Los componentes que acabamos que mencionar en la tabla 1, no solo aportan sus propias habilidades si no que colaboran entre ellos para lograr asumir responsabilidades más grandes.

- **Registrar un usuario:**
 1. La *interface holder* de Usuarios, revisa la *interface holder* tanto a empleados, administradores y clientes.
 2. Cada uno de estos holders le indica al de Usuarios si el usuario existe o no existe.
 3. Es importante saber este tipo de información porque es la manera de acceder al programa, por medio de un login y un password.
 4. Además, también es importante saber esta información para clientes porque no se le renta un vehículo a un cliente que no esté registrado.



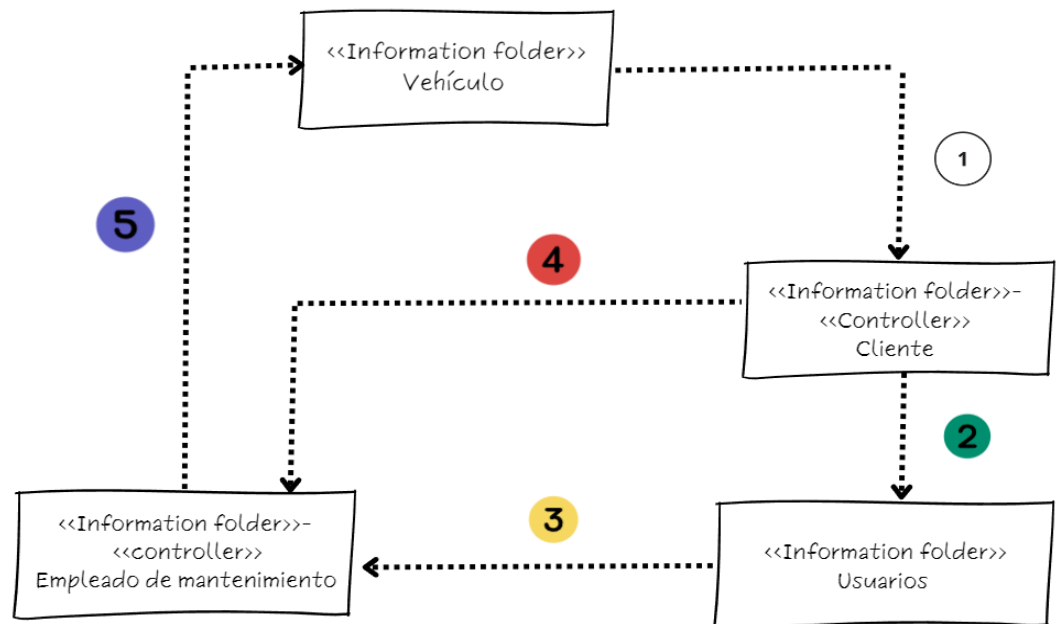
- **Verificar la disponibilidad de un vehículo:**

1. Para saber si un vehículo esta disponible primero se tiene que acceder a la información de cada sede y quien tiene acceso a esa información es el administrador global. Además, a ello decidimos acceder a vehículo como objeto único a través de su placa que tiene como llave y como valor toda su información como color, marca, etc.
2. El administrador global accede a la información de la sede y luego a las condiciones en las que se encuentra el vehículo.
3. Además de ello también tiene que acceder a empleados para conocer la información relevante de cada vehículo, en especial a empleado de mantenimiento dependiendo de si presenta alguna falla o si ya no esta en condiciones de seguir en el inventario.



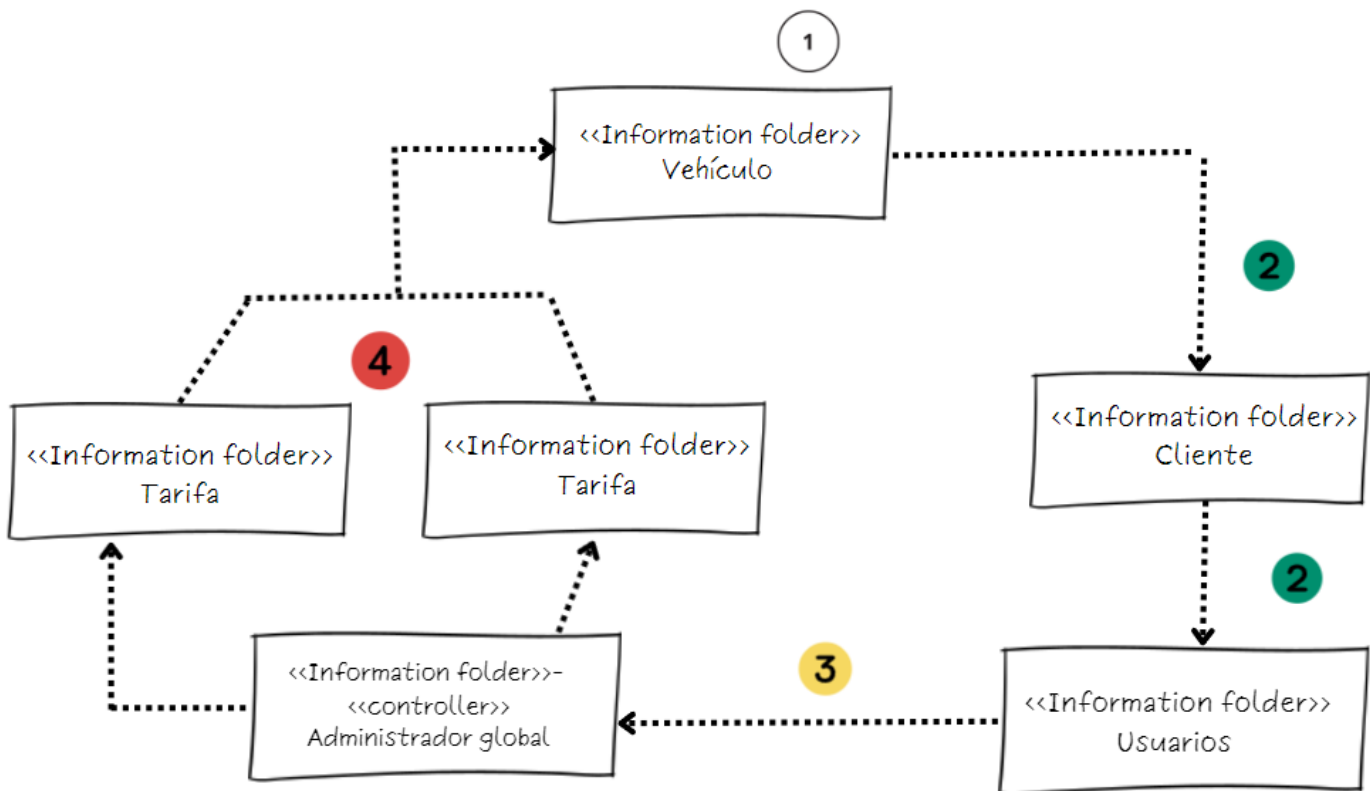
- **Hacer reserva de un vehículo:**

1. Para hacer una reserva es necesario verificar primero si la categoría de vehículo que el cliente quiere esté disponible.
2. Luego hay que verificar si el cliente puede reservar debido a que solo si está registrado podrá hacer la debida reserva.
3. Los usuarios de reserva son tanto los empleados de la empresa como los clientes
4. Los clientes deben crear y gestionar sus propias reservas
5. Los empleados deben encargarse de la entrega, devolución y registro de conductores extras.



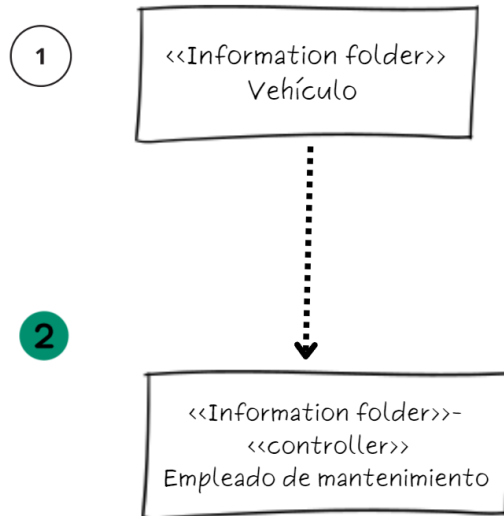
- **Alquiler de vehículo:**

1. Verificar disponibilidad del vehículo.
2. Verificar cliente esta registrado o no registrado.
3. Pasa al administrador global que es quien esta a cargo de ofrecer los seguros cuando se realiza un alquiler y además registrar si hay un conductor extra.
4. Además, una parte fundamental del alquiler es la tarifa que se define como una tarifa diaria establecida, o también por temporada alta, baja, categoría, dependiendo la sede, etc.



- **Eliminar un vehículo:**

1. Primero verificar si el vehículo está dentro del catálogo.
2. Segundo validar con los empleados de mantenimiento que el vehículo esta apto para ser eliminado del inventario



Uno de los aspectos importantes para tener en cuenta son los requisitos no funcionales del sistema que son los siguientes (que se tendrá en cuenta en la implementación):

- El sistema debe estar disponible 24/7 para los clientes y los empleados de la empresa.
- El sistema debe proteger los datos de los clientes y los empleados de la empresa.
- El sistema debe poder soportar un aumento en el número de clientes y transacciones.

Diseño final: a continuación, tenemos el diagrama de clases que incluye todas las clases, relaciones, atributos y métodos. (Se adjunta PDF para verlo con mayor precisión)

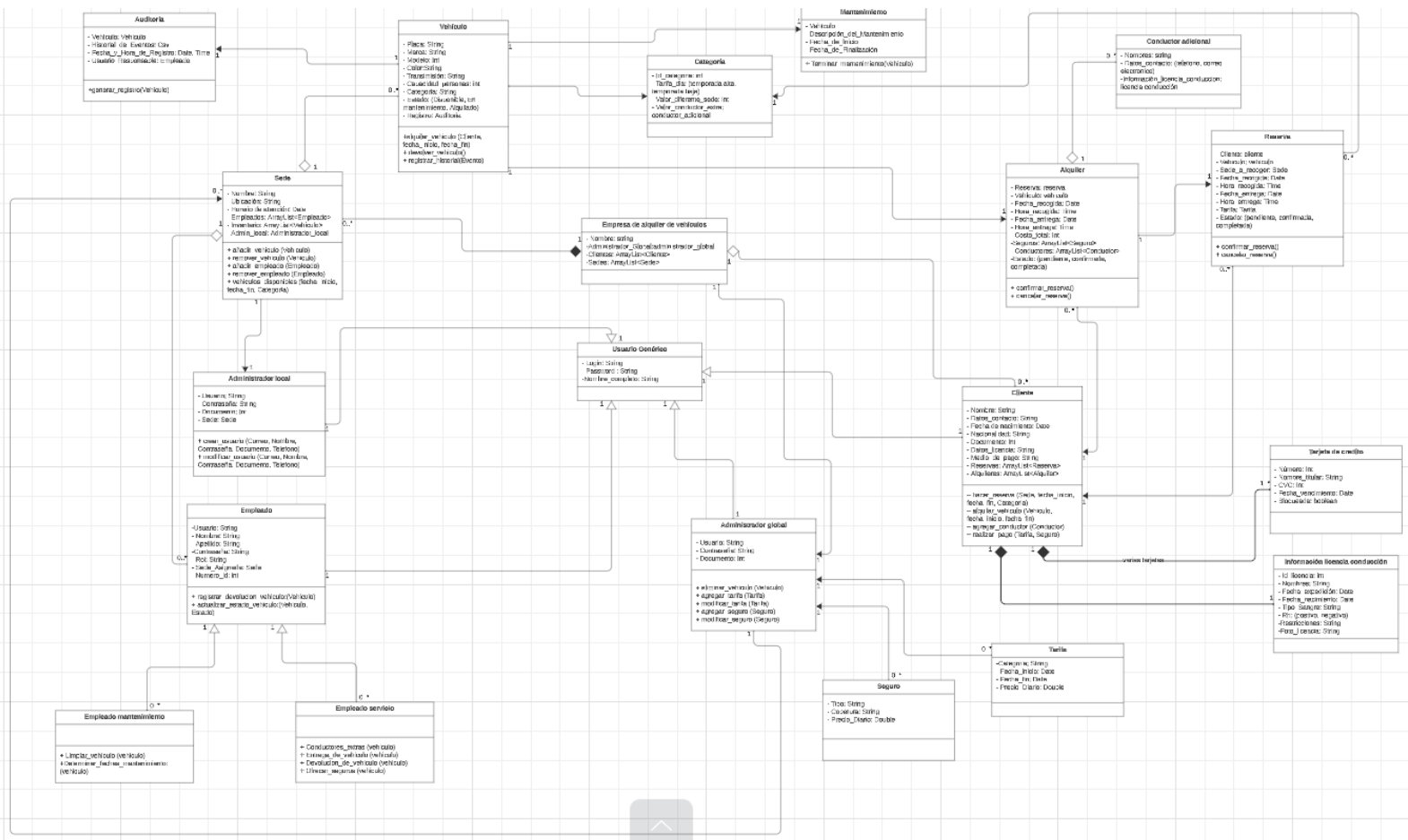


Diagrama de alto nivel: en este diagrama de alto nivel se pueden evidenciar los aspectos más importantes y generales, proporcionando una visión general del programa, sin entrar en detalles. (Se adjunta PDF para verlo con mayor precisión)

